

HP Asset Manager

ソフトウェアバージョン : 9.30

管理

ドキュメントリリース日 : 2011年 3月 31日
ソフトウェアリリース日 : 2011年 3月



ご注意

著作権について

(c) Copyright 1994-2011 Hewlett-Packard Development Company, L.P.

権利の権限

機密性のあるコンピュータソフトウェアです。

これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。

商用コンピュータソフトウェア、コンピュータソフトウェアに関する文書類、および商用アイテムの技術データは、FAR12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

保証

HP製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。

ここでの記載で追加保証を意図するものは一切ありません。

ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

サードパーティまたはオープンソースのライセンス契約の詳細を確認する場合、またはオープンソースコードを表示する場合は、次のいずれかのパスを使用してください。

- ソフトウェアのインストールCD-ROMの「ThirdParty」ディレクトリ
- ソフトウェアのインストール後にサードパーティまたはオープンソースツールのバイナリファイルが格納されるディレクトリ
- 『Open Source and Third-Party Software License Agreements』ガイドに記載されているコンポーネントのURL

商標について

- Adobe®, Adobe logo®, Acrobat®およびAcrobat Logo®は、Adobe Systems Incorporatedの商標です。
- Corel®およびCorel logo®は、Corel Corporation or Corel Corporation Limitedの商標または登録商標です。
- Javaは、Oracle Corporationおよびその関連会社の登録商標です。
- Microsoft®, Windows®, Windows NT®, Windows® XP, Windows Mobile®およびWindows Vista®は、米国におけるMicrosoft Corporationの登録商標です。
- Oracle®はOracle Corporationおよびその関連会社の登録商標です。
- UNIX®は、The Open Groupの登録商標です。

謝辞

本製品には、Apache Software Foundation (<http://www.apache.org/> [<http://www.apache.org/>]) により開発されたソフトウェアが含まれています。Copyright © The Apache Software Foundation.All rights reserved.

本製品には、OpenLDAP Foundationにより開発されたソフトウェアが含まれています。Copyright ©, The OpenLDAP Foundation, Redwood City, California, USA.All Rights Reserved.OpenLDAP®は、OpenLDAP Foundationの登録商標です。

本製品には、OpenSSL Toolkit (<http://www.openssl.org/>) での使用を目的にOpenSSL Projectにより開発されたソフトウェアが含まれています。Copyright © The OpenSSL Project.All rights reserved.

本製品には、OpenSymphony Group (<http://www.opensymphony.com/>) により開発されたソフトウェアが含まれています。Copyright © The OpenSymphony Group.All rights reserved.

本製品には、RSA Data Securityからライセンス供与されたコードが含まれています。

本製品には、JDOM Project (<http://www.jdom.org/>) により開発されたソフトウェアが含まれています。Copyright © Jason Hunter & Brett McLaughlin.All rights reserved.

目次

はじめに	11
管理モジュールの対象ユーザ	11
管理モジュールの使用目的	11
1. ライセンスキーをインストールする	13
HP AutoPass ライセンスキーを取得する	13
HP AutoPass ライセンスキーをインストールする	14
HP AutoPass ライセンスキーファイルを変更する	17
2. Asset Manager Application Designer - GUI	19
はじめに	19
グラフィカルインタフェース	20
3. データベースの標準記述ファイル	25
Asset Manager データベースの定義	25
データベース記述ファイルの概要	26
Database.txt と Tables.txt の構造	27
テーブルについて	28
フィールドについて	29
リンクについて	35
インデックスについて	42

4. Asset Managerデータベースの作成、変更、削除	45
Asset Managerデータベースの作成の前提条件	45
DBMSを使って空のシェルを作成する	46
データベースパスワードの暗号化	55
Asset Managerへの接続を作成する	56
Asset Manager Application Designerを使ってデータベース構造を作成する	57
データベース構造の更新	62
Asset Manager Automated Process Managerを使用してデータベースに署名する	62
Asset Managerクライアント言語を変更する	63
DBMSを変更する	72
Asset Managerデータベースを削除する	74
5. データベース記述ファイルの作成	77
はじめに	78
データベース記述パラメータ	79
記述テンプレートのシンタックス	81
特定のデータベース記述パラメータについて	89
6. データベースの診断と修復	109
GUI以外の診断方法	110
7. ODBCドライバを使ったデータベースへのアクセス	111
Asset Managerデータベースへのアクセス	111
例：ODBCドライバでSAP Crystal Reportsのレポートを作成する	113
8. Citrix XenAppを使用したデータベースへのアクセス	115
概要	115
Citrix XenAppをインストールする	116
既知の制約点	116
9. アーカイブ	117
アーカイブを有効にする	117
アーカイブの原則	119

アーカイブの無効化	124
10. 非正規化	125
値を正規化した方がいいのはどんな時ですか?	125
Asset Managerの非正規化の原則	126
値の非正規化	126
例	127
11. インポート	129
データのインポートの概要	130
推奨事項	131
フィールドにデータをインポートする際の制約事項	132
テキストファイルまたはODBCデータベースをインポートする	134
インポートスクリプトを保存および実行する	158
コマンドプロンプトからインポートスクリプトを実行する	160
12. データベースへのアクセスコントロール	163
アクセス権限の管理の重要性と概要	163
データのセキュリティと機密性	164
アクセス権限の定義	164
アクセス条件を定義する	166
Asset Managerユーザにアクセス条件を関連付ける	175
データベースへの接続数を管理する	177
Asset Managerデータベースへのユーザ認証を管理する	186
13. Asset Manager Automated Process Manager	231
Asset Manager Automated Process Managerの概要	231
Asset Manager Automated Process Managerを実行する	232
Asset Manager Automated Process Managerのメイン画面	235
Asset Manager Automated Process Managerの全般オプション	236
Asset Manager Automated Process Managerでモニタするモジュールを設定する	238
Asset Manager Automated Process Managerの検証を即時に起動する	260
Asset Manager Automated Process ManagerをWebブラウザで管理する	260
14. メッセージシステム	265
メッセージシステムの仕組み	265
Asset Managerでメッセージシステムを指定する	266

メッセージを送受信する	275
一般的な接続エラー	277

15. スケジュールのグラフィック表示画面のカスタマイズ 281

概要	281
スケジュールのグラフィック表示のページ機能をカスタマイズする	281
テーブルにスケジュールのグラフィック表示のページを追加する	282
ウィザードにスケジュールのグラフィック表示のページを追加する	287
参考情報：スケジュールのグラフィック表示のページのパラメータのシナタックス	289

16. Asset ManagerをDDEサーバとして使用する 297

DDEサーバの定義	297
DDEコールのメカニズム	297
DDEコマンドの概要	299
グローバルコマンド	299
テーブルに関連するコマンド	303
テーブルおよびフィルタまたはリンクと関連するコマンド	310
ヘルプデスクの応答に関連するコマンド	315
DDEコール例の概要	319
例1：Asset Managerの内部DDEコール	319
例3：Visual BasicからのDDEコール	321

17. WANネットワークにおけるAsset Managerの最適化 325

〔編集／オプション〕メニューのオプション	325
リスト	326
表示を制限する	329
接続キャッシュ	329
アクセス制限	330
1つのクライアントの設定を他のクライアントに適用する	330

索引 331

図の一覧表

2.1. データベースを開く画面	22
3.1. 「履歴」テーブルとターゲットテーブルとのリンク	37
3.2. 中間テーブルの役割	38
3.3. 「資産」テーブルと「固定資産」テーブルとのリンク	38
11.1. インポートするデータのタイプを選択する	129
14.1. 内部メッセージシステムの仕組み	266
15.1. スケジュールのグラフィック表示 - 作業指示での例	286

表の一覧表

3.1. フィールド - タイプ	31
3.2. データの入力形式とタイプ	31
3.3. [ユーザタイプ] フィールド値	33
3.4. 既存のリンクのタイプ	36
3.5. リンクにより保存される情報の属性	36
4.1. Oracleインスタンスを作成するために重要なサーバパラメータ	48
4.2. サーバとデータベースの重要なパラメータ	50
4.3. Asset Manager Windows クライアント - ラベル、画面、ページなどの表示言語	66
4.4. Asset Manager Windows クライアント - データの表示言語	66
4.5. Asset Manager Windows クライアント - ワンポイントの表示言語	67
4.6. Asset Manager Windows クライアント - プログラムメッセージの表示言語	68
4.7. Asset Manager Windows クライアント - レコードタイプに基づく表示言語	68
5.1. Databaseインスタンスのプロパティ	90
5.2. Tableインスタンスのプロパティ	90
5.3. Tableインスタンスのオブジェクト	91
5.4. Fieldインスタンスのプロパティ	92
5.5. Fieldインスタンスのオブジェクト	94
5.6. Typeプロパティの値	94
5.7. UserTypeプロパティの値	95
5.8. Linkインスタンスのプロパティ	96
5.9. Linkインスタンスのオブジェクト	97

5.10. Typeプロパティの値	97
5.11. UserTypeプロパティの値	98
5.12. Indexインスタンスのプロパティ	98
5.13. Indexインスタンスのオブジェクト	99
5.14. Scriptインスタンスのプロパティ	99
5.15. Scriptインスタンスのオブジェクト	100
5.16. FeatDescインスタンスのプロパティ	100
5.17. FeatDescインスタンスのオブジェクト	101
5.18. FeatParamDescインスタンスのプロパティ	101
5.19. FeatParamDescインスタンスのオブジェクト	102
5.20. FeatClassDescインスタンスのプロパティ	103
5.21. FeatParamDescインスタンスのオブジェクト	103
5.22. CalcFieldDescインスタンスのプロパティ	104
5.23. CalcFieldDescインスタンスのオブジェクト	104
5.24. Viewインスタンスのプロパティ	104
5.25. Viewインスタンスのオブジェクト	105
5.26. Pageインスタンスのプロパティ	105
5.27. PageItemインスタンスのプロパティ	106
5.28. PageItemインスタンスのオブジェクト	106
5.29. SysEnumValueインスタンスのプロパティ	107
5.30. Stringインスタンスのプロパティ	107
5.31. ScriptFieldインスタンスのプロパティ	107
5.32. グローバル変数	108
7.1. ODBCドライバの利点	112
12.1. データベースのアクセスタイプ	177
12.2. 「一時」アクセスタイプ - 読み書きモードでアクセス可能なテーブル	178
12.3. 「一時」アクセスタイプ - 特定の条件下で、読み書きモードでアクセ スが可能なテーブル	181
13.1. Asset Manager Automated Process Managerプログラムのメニュー とアイコン	235
13.2. 実行日	238

はじめに

管理モジュールの対象ユーザ

管理モジュールは、Asset Managerを使用するすべてのユーザ企業を対象としています。

このモジュールは主に、以下のユーザにより導入されます。

- ネットワーク管理者
- データベース管理者
- Asset Manager管理者

管理モジュールの使用目的

管理モジュールを使うと、以下のタスクを実行できます。

- Asset Managerデータベースの管理
- ODBCデータベースへの接続の管理
- Asset Managerの管理
- Asset Manager Automated Process Managerの管理
- ユーザプロファイルの管理

上記のタスクを実行すると、次のことができます。

- データベースを作成、更新、メンテナンス、カスタマイズする。
- データをインポートする。

- ODBCデータベースへの接続を作成、更新、削除する。
- ネットワーク上での利用のためにAsset Managerを最適化する。
- 自動モニタリングプロセスを設置する。
- ユーザプロファイルを作成、更新、削除する。

1 ライセンスキーをインストールする

HP AutoPassライセンスキーを取得する

Asset Managerデータベースに接続するには、HP AutoPassライセンスキーが必要となります。HPが提供するライセンスキーには、契約で取得したAsset Managerのモジュール、機能、および関連サービスのリストを使用する権限が含まれています。

以下の項目も決まります。

- 承認されたユーザの数（固定ユーザと不特定ユーザ）
- 一時ユーザの数

ライセンスキーは、初期のインストール中に取得およびインストールされ、定期的に更新できます。

新しいライセンスキーを取得するたびに、HP AutoPass License Management Toolに記録されている既存のセットに追加する必要があります。

ヒント:

HP AutoPass License Management Toolでは、インストールされたライセンスキーのセットを、HP AutoPass License Management Toolインストールフォルダの「data」フォルダ（デフォルトでは、「C:\Program Files\HP\Asset Manager 9.30\xx\autopass\data」）にある「LicFile.txt」に格納します。

HPからライセンスキーを取得するには、いくつかの方法があります。これらの方法については、「[HP AutoPassライセンスキーをインストールする \[P. 14\]](#)」セクションの手順で説明されています。

HP AutoPass License Management Toolと共に提供されるオンラインヘルプには、これらの方法の詳細情報（[ヘルプ]メニューを参照）、および関連するメンテナンス手続きも記載されています。または、HPテクニカルサポートへお問い合わせください。

HP AutoPassライセンスキーをインストールする

警告:

このインストールを試行する前に、HP AutoPass License Management Toolを起動するマシンにJava（少なくともJRE 1.5）がインストールされていることを確認し、JREまたはJDKのインストールフォルダを指すように**JAVA_HOME**システム変数を追加または変更してください。

例：

```
C:\Program Files\Java\jre6
```

ライセンスキーのフルセットを次のマシンにインストールする必要があります。

- 1 **Asset Manager Application Designer**を使用してデータベースをアクティブにするマシン。
 - ▶ **HP AutoPassライセンスキーファイルを変更する [P. 17]**.
- 2 **Asset Manager Automated Process Manager**のサーバとして機能するマシン（ここから定期的にデータベースに署名される）。

注意:

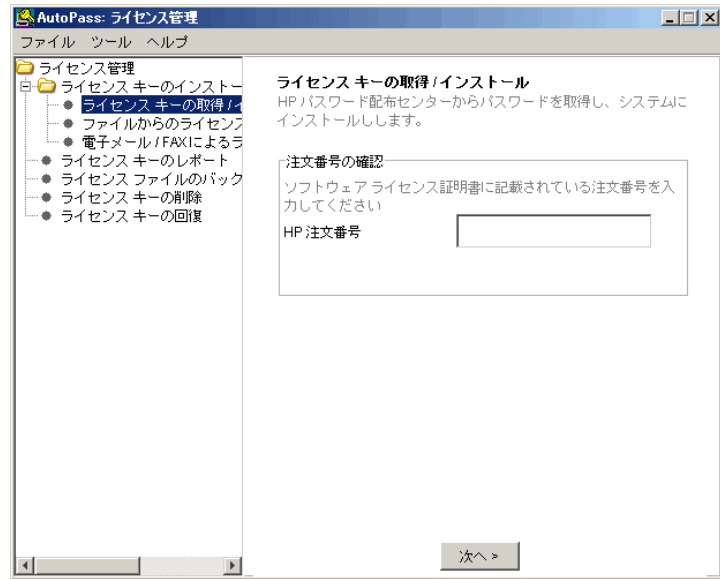
これは同じマシンにできます。

HPからライセンスキーを取得するには、いくつかの方法があります。

例えば、ファイル経由でライセンスキーを受け取る場合、次の手順に従います。

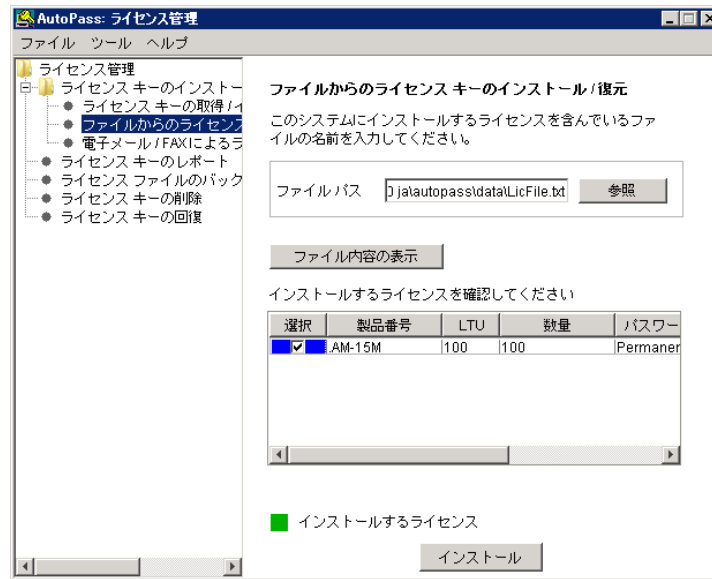
- 1 **Asset Manager**プログラムグループ（例えば、Windowsメニューの[スタート/すべてのプログラム/HP/**Asset Manager 9.30 ja**]メニュー）から

HP AutoPass License Management Toolを起動します。次の画面が表示されます。



- 2 [ファイルからのライセンスキーのインストール/復元] を選択します。
- 3 [参照] をクリックし、受け取ったファイルを選択します。

- 4 【ファイル内容の表示】をクリックします。次の例のように、ライセンス権限のリストが表示されます。



- 5 インストールするすべての権限（通常、すべて利用可能）の【選択】列のチェックボックスをオンにし、【インストール】をクリックします。権限が緑色に変わり、インストールされたことを確認できます。
- 6 【ライセンスキーのレポート】を選択すると、HP AutoPass License Management Toolに記録されているライセンスキーを確認できます。
- 7 これらの権限を別のマシンにバックアップまたは複製する場合は、【ライセンスファイルのバックアップ】を選択します。これにより、【ファイルからのライセンスキーのインストール/復元】を使用してインストールまたは復元できる、「.dat」ファイルが作成されます。
- 8 【ファイル/閉じる】を選択します。

インストールを完了するには、さらに2つの手続きを実行する必要があります。

- 1 新規または変更済みのライセンスキーファイルをインストールしたら、Asset Manager Application Designerを使用してデータベースをアクティブにします。HP AutoPassライセンスキーファイルを変更する [P. 17]を参照してください。
- 2 Run Asset Manager Automated Process Managerを実行し、データベースが定期的に署名されるように、データベースサーバに信号送信（UpdateToken）モジュールを設定します（【ツール/モジュールの設定】メニュー）。この

サーバのインスタンスは、なるべく他のコンピュータに移動しないでください。移動すると、ライセンスファイルを変更する必要があります。

この署名プロセスでは、HP AutoPassによって付与されたライセンス権限（ローカルのライセンスキーファイル）とAsset Manager Automated Process Managerが接続されているデータベースでアクティブにされたライセンス権限が比較されます。前者の権限が後者の権限より少ない場合、データベースはロックされます。

 **注意:**

DBMSの応答時間が短いコンピュータでAsset Manager Automated Process Managerを実行することをお勧めします。

HP AutoPassライセンスキーファイルを変更する

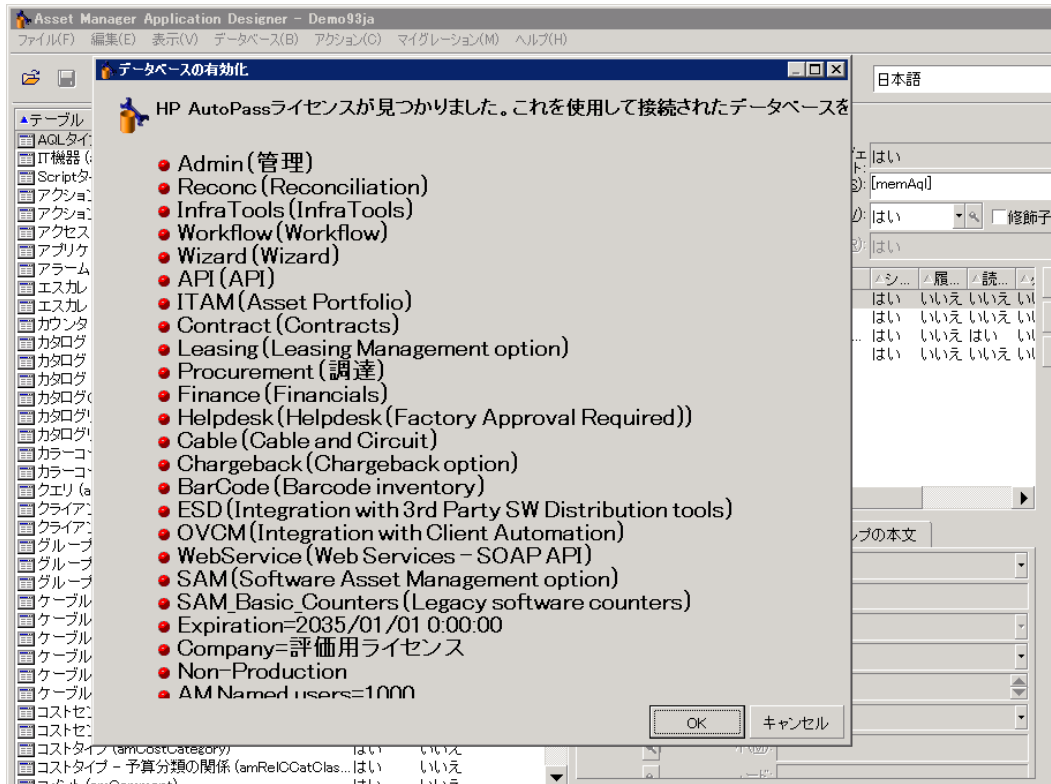
次の場合、HP AutoPassライセンスキーを再インストールする必要があります。

- **ITAM**ライセンスキーの有効期限が切れた場合（**AM**資産ポートフォリオモジュール）
- 追加のライセンスキーを受け取った場合

上記のいずれかに該当する場合、次の手順に従います。

- 1 新規または追加のHP AutoPassライセンスキーを取得します。
 - ▶ HP AutoPassライセンスキーを取得する [P. 13]
- 2 新規または変更済みのライセンスキーをインストールします。
 - ▶ HP AutoPassライセンスキーをインストールする [P. 14]
- 3 Asset Manager Application Designerを使用してデータベースを再度アクティブにします。
 - a **【アクション/データベースの有効化】** オプションを選択します。以下の例に示すように、**【データベースの有効化】** 画面が表示され、契約によって付与された権限のリストが表示されます（この時点で確認する必要があります）

ります。全リストを表示する必要がある場合、ウィンドウのサイズを変更します)。



- b [OK] をクリックし、表示されたリストを使用して接続されたデータベースをアクティブにします。

警告:

アクティブ化されていないデータベースに接続を試みると、次のエラーメッセージが表示されます。

- Windowsクライアント :

データベースがアクティブ化されていません。有効なHP AutoPassライセンスを使用してデータベースをアクティブにする必要があります。

- Webクライアント :

com.peregrine.ac.AmException: Error (12 001) : このデータベースにはライセンスファイルがありません。開くことができません

2 Asset Manager Application Designer - GUI

はじめに

Asset Manager Application Designerは、Asset Managerデータベースの管理用ツールです。このツールを使用すると、以下の多様な操作を実行できます。

- データベースの作成
- データベースに格納されたオブジェクトのカスタマイズ（テーブル、フィールド、リンク、インデックス、画面など）
- 破損したデータベースの修復
- データベース構造の更新
- データベースからのデータの抽出

このツールはデータベース管理者が使うことを想定しており、使用する前に次の点を考慮する必要があります。

- このツールを使ってAsset Managerデータベースの構造を変更するには、専門の経験と知識が必要です。
- Asset Managerに「Admin」でログインすると、データベースを変更できますが（オブジェクトの設定、リンクの設定など）、Asset ManagerとAsset Manager Application Designerを使用して、同一のデータベースを同時に変更してはなりません。
- Asset Manager Application Designerでデータベースに接続するには、「Admin」（管理者）または同等の権限でログインする必要があります。「Admin」ログインを複数のユーザに割り当てないでください。データベースの更新時に競合が発生したり、データベースの整合性を調整するためにデータベースが使えなくなったりする場合があります。

グラフィカルインタフェース

Asset Manager Application Designerは、Asset Managerと同じプログラムグループから実行できます。プログラムのアイコンをクリックして起動します。

注意:

Asset Manager Application Designerを起動しても、ファイルを読み込まない限りメイン画面は灰色で表示されます。最後に使用したドキュメントを自動的に読み込むように設定するには、[編集/オプション]メニューの[ドキュメント]項目で[起動時に前回最後に使用したドキュメントを自動的に読み込む]オプションを[はい]にします。

インタフェースの概要

Asset Manager Application Designerのユーザインタフェースは、3つの枠から構成されています。

- ツールバーが付属したメニューバー
- Asset Managerデータベースのテーブルリストを含む枠
- メイン枠、またはカスタマイズ枠。テーブルのオブジェクトに関する情報をまとめています。

注意:

現在存在する言語の重要性はすべて同じです。作成言語が特別なステータスを持つわけではありません。

Asset Manager Application Designerのグラフィカルインタフェースは、特にデータの参照と作成に関してはAsset Managerと同様のインタフェースになっています。基本的な画面上での操作については、『はじめに』ガイドを参照してください。

グラフィカルインタフェースの基本操作

言語の選択

ツールバーの右下には、マルチリンガルのデータベースオブジェクト（フィールド、画面、ページラベル、システムリストデータ値など）の表示言語を選択するためのドロップダウンリストがあります。

▶ オブジェクトの属性に基づく表示言語 [P. 66]

このフィールドは、データベースに接続したとき（データベース記述ファイルを開いたときではなく）に使用可能になります。

このフィールドでは以下の言語が選択可能です。

- データベースの作成に用いられたAsset Manager Application Designerの言語、
- データベースの作成後に挿入された言語（ [アクション/データベースに言語を挿入] ）。

▶ Asset Managerクライアント言語を変更する [P. 63]

テーブルを選択する

テーブルを選択するには、ユーザインタフェースの左側枠内でテーブル名をクリックします。

表示するオブジェクトのタイプを選択する

[表示] メニューでは、メイン枠内に表示されるオブジェクトのタイプを選択できます。使用可能なオブジェクトの種類は以下のとおりです。

- フィールド
- リンク
- インデックス
- 画面
- ページ

カスタマイズ枠を使用する

カスタマイズ枠は3つの部分から構成されています。

- 1番目の枠には、選択したテーブルの一般情報が表示されます。
- 2番目の枠には、テーブル内の1タイプの全オブジェクトが表示されます。オブジェクトのタイプを選択するには [表示] メニューを使用します。
- 3番目の枠には、選択したオブジェクトの情報が表示されます。



注意:

カスタマイズできるのは一部の情報のみです。編集不可能な値は灰色のフィールド内に表示されます。

ファイルの操作機能

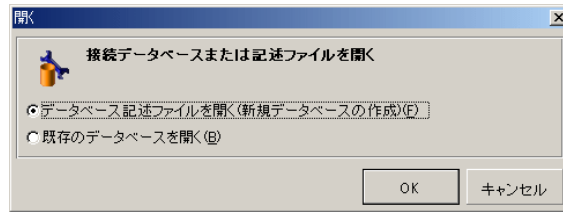
[ファイル] メニューには、ファイルの読み込みと保存に関する機能がまとめられています。

ファイルを開く

[ファイル/開く] を選択します。

次のダイアログボックスが表示されます。

図 2.1. データベースを開く画面



このダイアログボックスでは、Asset Manager Application Designerの以下の機能から1つを選択します。

- **【データベース記述ファイルを開く(新規データベースの作成)】** オプションを選択すると、新規のデータベースを作成するか、またはデータベース記述ファイルを変更することができます。
- **【既存のデータベースを開く】** オプションを選択すると、既存のデータベースをカスタマイズできます。

上記のいずれかのオプションを選択すると、Asset Manager Application Designerセッションが開始します。

注意:

【ファイル】メニューの一番下には、一番最後に開いた4つのドキュメントがデフォルトで表示されます。ここから直接選択するとドキュメントをすばやく開くことができます。最近使用したファイルのリストに表示されるドキュメント数は、[[ファイル]メニューに格納する、最近開いたドキュメントの最大数]オプションを使用して設定できます（[編集/オプション]をクリックし、次に[ドキュメント]プランチを展開）。

【データベース記述ファイルを開く - 新規データベースの作成】

Asset Manager Application Designerで新規のデータベースを作成するには、データベースの記述データが必要です。この記述データはAsset Managerデータベースの構造情報を含むファイルです。このファイルは、データベースの作成時にデータのテンプレートの役割を果たします。

この選択を承認することにより、Asset Managerのインストール先フォルダの「Config」サブフォルダに格納されている「gbbase.xml」記述ファイルを、Asset Manager Application Designerに提供しなければなりません。

ヒント:

標準の記述ファイルのコピーを常に手元に残しておくため、Configフォルダのバックアップコピーを作成することをお勧めします。




注意:

または、[ファイル/データベース構造の出力] メニューを使って、自動的に記述ファイルを生成することをお勧めします。記述ファイルは、データベースを修復するときにも必要です。

[既存のデータベースを開く]

既存のAsset Managerデータベースをカスタマイズするには、データベースに接続する必要があります。このオプションを選択すると、Asset Managerと同じ接続ウィンドウが表示されます。

このドキュメントでは、 ボタンを使った接続設定の編集についての説明は省略します。この機能については、『はじめに』ガイドを参照してください。

ファイルを閉じる

[ファイル/閉じる] メニューを選択すると、1つのAsset Manager Application Designerセッションが終了します。変更を行った場合は、セッションを終了する前に保存を確認するメッセージが表示されます。

データベースに変更が加えられると、Asset Manager Application Designerのタイトルバーに表示されるドキュメント名にアスタリスク (*) が付くため、保存時に参考にできます。

ファイルを保存する

2つの保存方法があります。

- [ファイル/データベース構造の更新] メニューによりデータベース記述ファイルまたはデータベースに対して行った変更を保存できます。
- [ファイル/データベース構造の出力] メニューでは以下の2つのタスクを実行できます。
 - 1 データベース記述ファイルを開いている場合は、データベースの構造を新しいデータベース記述ファイルに保存できます。
 - 2 データベースファイルを開いている場合は、データベースの構造をデータベース記述ファイルに保存できます。

[アプリケーションの終了]

このメニューを選択すると、Asset Manager Application Designerが終了します。セッション実行中に変更があった場合は、保存を確認するメッセージが表示されます。

編集機能

Asset Manager Application Designerには標準の編集機能があります。

コピー、切り取り、貼り付け

【編集】メニューでは次の編集機能を選択できます。

- 【切り取り】（ショートカットキー [Ctrl+X]）：選択部分を切り取ります。
- 【コピー】（ショートカットキー [Ctrl+C]）：選択部分をコピーします。
- 【貼り付け】（ショートカットキー [Ctrl+V]）：切り取ったまたはコピーした情報を貼り付けます。



注意:

これらの機能は、フィールド編集枠以外では使えません。

検索

Asset Manager Application Designerの【編集／検索】メニューでは高度なテキスト検索を実行できます。

データベースの構造はオブジェクト（テーブル、フィールド、リンクなど）から構成されています。オブジェクトは、オブジェクトのプロパティに関する情報で特徴付けられています。例えばテーブルの**SQL名**はテーブルのプロパティの1つであり、フィールドの**タイプ**はこのフィールドのプロパティの1つに当たります。データベースの全体的な構造は階層オブジェクトから成り立っており、各オブジェクトは1つまたは複数のプロパティを有しています。

Asset Manager Application Designerはデータベース構造の全体を検索します。検索を実行すると、データベースの全オブジェクトとプロパティが検索されます。結果リストはメモリに作成され、【編集／次を検索】と【編集／前のレコード】メニュー（または [F3] キーと [Shift+F3] キー）を使うと、このリスト内を参照できます。



注意:

ダイアログボックスの【上へ】と【下へ】オプションで、検索方向を指定できます。

アプリケーションオプション

【編集／オプション】メニューではAsset Manager Application Designer使用時のオプションを指定できます。各オプションの内容は、ダイアログボックス下の説明枠内に説明されています。

3 データベースの標準記述ファイル

このセクションでは、データベースの構造について説明します。

 **注意:**

固有のインデックスを削除または変更したり、外部ツール（INSERT、DELETE、UPDATEステートメントまたはトリガ）を使ってデータベースに直接書き込んだりしないでください。データベースに書き込む場合は、Asset Manager APIをお使いください。

Asset Managerデータベースの定義

Asset Managerデータベースは、管理資産に関するあらゆる情報を記録したファイルの集まりです。インストールを簡単にするためには、これらのファイルをすべてローカルディスクドライブ上またはネットワークファイルサーバ上の1つのフォルダに配置します。

Asset Managerには、デモ用データベース（サンプルデータベース）が付属しています。

Asset Managerでは複数のデータベースを作成でき、1回のユーザーセッションで1つのデータベースを開くことができます。また、複数のコンピュータから、同時に同じデータベースに接続できます。Asset Managerプログラムは、トランザクション処理を使ってデータベース内のデータを更新します。この処理方法により、データベースのデータを確実に更新および最適化することができます。

データベース記述ファイルの概要

はじめに

データベースへのデータのインポート、外部ツールによるデータベースへのアクセス、クエリの記述などの作業を行うには、データベースの構造を理解する必要があります。フィールドの名前、最大長、データ型、さらに固有の値にするかどうかなどの情報が必要になります。

取得方法

Asset Managerデータベースの構造の記述を見るには、次のファイルまたはプログラムを参照します。

- 「**database.txt**」および「**tables.txt**」ファイル：データベースの完全な構造が記述されています。この2つのファイルは、Asset Managerのインストール先フォルダの「**doc\infos**」サブフォルダに格納されています。



注意:

上記のファイルでは、デフォルトのデータベース構造が記述されています。ユーザによるカスタマイズ情報は記述されていません。

これらの記述ファイルに、お使いのデータベースのカスタマイズ情報を反映させるには、Asset Manager Application Designerソフトウェアを使ってお使いのデータベースに接続する必要があります。

- Asset Manager Application Designerプログラム：Asset Managerデータベース（テーブル、フィールド、リンク、インデックス）の記述ファイルを次のようにして作成できます。

以下を利用します。

- Asset Managerのデータベース記述ファイル（「**gbbase*.***」ファイル）を開くか、またはAsset Managerデータベースへ接続します。
- 作成する情報を記述するテンプレート（拡張子が「**.tpl**」のファイル）を使います。Asset Managerプログラムに付属している標準テンプレートを基に、独自のテンプレートを作成することができます。また、**RTF**または**HTML**形式の記述ファイルを作成することもできます。



注意:

Asset Managerの標準バージョンに付属している「dbdict.tpl」テンプレートを使うと、カスタマイズデータ（任意管理項目、特殊フィールド、設定スクリプトなどに関する情報）を使用中のデータベースから標準のテキストファイルにエクスポートできます。ソース管理ツールでこの記述ファイルを使うと、データベースで行ったカスタマイズ情報をトラッキングする際に便利です。

- Asset Managerプログラム

Database.txtとTables.txtの構造

データベースの構造は、Asset Managerのインストール先フォルダの「doc\infos」サブフォルダに格納されている「database.txt」ファイルと「tables.txt」ファイルに記述されています。



注意:

上記のファイルは、Asset Manager Application Designerソフトウェア、および「dbase.tpl」と「tables.tpl」のテンプレートを使って作成されています。これらのテンプレートは、Asset Managerのインストール先フォルダの「doc\infos」サブフォルダに格納されています。

上記のファイルのフォーマットは以下のとおりです。

- データ型：テキスト
- 区切り文字：タブ
- 文字セット：ANSI

これらのファイルを表示する最も簡単な方法は、スプレッドシートでファイルを開く方法です。

ファイルには出来るだけたくさんの情報が含まれていますが、必要に応じて不要な情報を非表示にしたり削除したりすることができます。

「database.txt」ファイルの構造

「database.txt」ファイルの構造は次のとおりです。

- 1行につき1つのフィールド、リンク、またはインデックスの情報が記述されています。
- テーブルはSQL名順に並んでいます。
- 各テーブルの情報は、次の順に並んでいます。
 - 1 フィールド

- 2 リンク
- 3 インデックス

- フィールド、リンク、インデックスはSQL名順に並んでいます
- 1列に1つのデータ項目が入っています。ファイルの1行目は、情報の生成に使ったテンプレートパラメータを示します。2行目は各情報のタイプを示します。

「tables.txt」ファイルの構造

「tables.txt」ファイルの構造は、次のように非常にシンプルです。

- 1行に1つのテーブルの情報が入っています。
- テーブルは、SQLテーブル名順に並んでいます
- 1列に1つのデータ項目が入っています。ファイルの1行目は、情報の生成に使ったテンプレートパラメータを示します。2行目は各情報のタイプを示します。



注意:

SQL名は、テーブル、フィールド、リンク、およびインデックスの固有の名前です。SQL名は、Asset Managerの全言語のバージョンに共通です。

テーブルについて

このセクションでは、Asset Managerデータベースのテーブルについて説明します。内容は次のとおりです。

- Asset Managerテーブルの表記名
- Asset Managerテーブルのレコード名の記述文字列

Asset Managerテーブルの表記名

Asset Managerデータベースの各テーブルには、次の種類の表記名が付きます。

- **SQL名**：Asset ManagerテーブルのSQL名は英語で表記され、先頭に「am」が付きます。SQL名は、Asset Managerの全言語のバージョンに共通です。
- **ラベル**：Asset Managerで表示されるテーブル名です。

Asset Managerでテーブル名を表示するスペースが小さい場合（フィルタとクエリで使うツリービュー、エラーメッセージ、メニュー名など）にこの表記が使われます。

- **説明**：Asset Managerで説明全体を表示できる十分なスペースがある場合（ステータスバーなど）にこの表記が使われます。

テーブルのレコード名の記述文字列

Asset Managerの各テーブルに含まれるレコード名の表記方法（記述文字列）を定義するには、Asset Manager Application Designerを使います。

Asset Managerの各テーブルの記述文字列を定義すると、レコード選択用のドロップダウンリストに表示されるレコード名の表記方法を指定できます。

また記述文字列により、Windowsクライアントのタイトルバーに表示されるレコード名の表記方法も指定できます。

テーブルの記述文字列には、次の要素を使うことができます。

- フィールドのSQL名（括弧または大括弧内）
- テキスト文字列（区切り文字なし）
- 複数の階層のリンク

例：



このテーブルの例では、メーカー（**Brand.Name**）が「Asus」、モデル（**Model.Name**）が「AsusLX512」、資産タグ（**AssetTag**）が「1」です。この場合、詳細ウィンドウのタイトルには、次のように表示されます。

資産 'Asus AsusLX512(1)' の詳細

ドロップダウンリストをリスト形式で表示する場合は、リストにフィルタが適用されている場合を除き、昇順でレコード名が表示されます。

階層構造のあるリストをツリー構造で表示する場合は、レコードは**完全名**に基づいて昇順で表示されます。【**完全名**】は、階層構造のテーブルで使うフィールドです。SQL名は「FullName」です。

フィールドについて

このセクションでは、Asset Managerデータベースのフィールドについて説明します。内容は次のとおりです。

- Asset Managerフィールドの表記名
- フィールドのデータ型と入力フォーマット
- dtLastModifフィールド
- FullNameフィールド

Asset Managerフィールドの表記名

Asset Managerテーブル内の各フィールドには、次の表記名が使われます。

- **SQL名**：SQL名は英語です。Asset Managerのすべての言語のバージョンで同じSQL名を使っています。SQL名には、フィールドのデータ型によって次の接頭コードが付きます。

接頭コード	フィールドのデータ型
"b"	ブール (はい/いいえ)
"d"	日付
"dt"	日付+時刻
"l"	整数 (32ビット)
"m"	金額
"p"	パーセント
"se"	システムリストデータ
"ts"	期間
"mem"	メモフィールド
なし	テキスト

- **ラベル**：Asset Managerでフィールド名を表示するスペースが小さい場合（詳細画面、クエリ作成時のツリー表示用など）に使います。
- **説明**：Asset Managerのステータスバーに表示される長いフィールド名です。

フィールドのデータ型（「タイプ」）と入力フォーマット（「入カタイプ」）

このセクションでは、次の内容について説明します。

- Asset Managerのフィールドのデータ型
- Asset Managerのフィールドのデータ入力時のフォーマットとデータ型
フィールドのデータ型、データ入力時のフォーマットおよびデータのタイプを表示するには、次の方法があります。
- Asset Manager Application Designerで、**[全般]** タブページの **[データ型]** フィールドを使います。
- **「database.txt」** ファイルを読み込みます。このファイルは、Asset Managerのインストール先フォルダの「doc\infos」サブフォルダ内に格納されています。**[Field Type]** 列が、フィールドのデータ型を一覧表示します。
- 各フィールドのヘルプを表示します（Windowsクライアントのみ）。

テーブル内のフィールドのデータ型（タイプ）

フィールドの保存データ型は次のとおりです。

表 3.1. フィールド - タイプ

「database.txt」ファイルの [Field Type] 列の値	Asset Manager Application Designerの [データ型] フィールドの値	説明
Integer (8 bit)	整数 (8ビット)	-128から+127の整数
Integer (16 bit)	整数 (16ビット)	-32,768から+32,767の整数
Integer (32 bit)	整数 (32ビット)	-2,147,483,647から+2,147,483,646の整数
Floating point number	浮動小数点数	4バイト浮動小数点数
Double-precision number	倍精度	8バイト浮動小数点数
Text	テキスト	文字列の長さ制限のあるテキストフィールド。あらゆる文字を入力できます
Date	日付	日付型みのフィールド (時刻なし)
Time	時刻	時刻型みのフィールド (日付なし)
Date and time	日付+時刻	日付と時刻型のフィールド
Variable length binary field	可変長バイナリフィールド	画像またはその他のバイナリファイルの保存に使用します。サイズの制限はありません。
Long text field	メモフィールド	可変長のテキストフィールド。コメント、問題のタイプの解決策や質問の説明などを入力します。 一部のDBMSではこのフィールドの使用方法に制限があります。例えば、Oracle for WorkGroupsでは、この型のフィールドで並べ替えできません。

データの入力フォーマットとデータ型 (「入力タイプ」)

データ入力フォーマットとデータ型は、次のもので表されます。

- Asset Manager Application Designerの [全般] タブページの [入力タイプ] フィールドの値：デフォルトでは、[データ型] フィールドの値になります。
- 「database.txt」ファイルの [Field data display and entry type] 列のフィールドの値が表示されます。

検証時にデータの入力タイプを指定することができます。

表 3.2. データの入力形式とタイプ

「Database.txt」ファイルの [Field data display and entry type] 列の値	Asset Manager Application Designerの [ユーザタイプ] フィールドの値	説明
Default	デフォルト	情報は、データベースに保存されているとおり、つまりフィールドの「データ型」に従って表示または入力されます。

「Database.txt」ファイルの [Field data display and entry type] 列の値	Asset Manager Application Designerの [ユーザタイプ] フィールドの値	説明
Boolean (Yes/No)	ブール (はい/いいえ)	ブール値
Number	数値	数値
Monetary	金額	金額
Date	日付	日付型の値のみが入ります。
Date and time	日付+時刻	日付+時刻型の値が入ります。
Time	時刻	時刻型の値のみが入ります。
System itemized list	システムリストデータ	このタイプのフィールドでは、システムのリストデータから値を選択する必要があります。システムリストデータとは、リストデータがソフトウェア自体で定義されており、ユーザがカスタマイズできないデータのことを指します。
Custom itemized list	カスタムリストデータ	[タイプ] (seType) が [開く] の場合、このタイプのフィールドでは、ユーザがカスタマイズできるリストデータから (ドロップダウンリストを使って) 選択します。
Value of feature	任意管理項目の値	これはプログラム用です。使用できません。
Percentage	パーセント	パーセントが入ります。パーセント形式では、常に10進法2桁の形式で表示します。
Duration	期間	期間型の値が入ります。 表示フォーマットとデフォルトの単位は、 [UserTypeFormat] プロパティで定義します。
Table or field name	テーブルまたはフィールド名	テーブルまたはフィールドのSQL名
BASIC script	BASICスクリプト	これはプログラム用です。使用できません。
N/A	不特定	その他のタイプ

 **注意:**

リスト画面と詳細画面では、データ型が [日付]、[日付+時刻]、[時刻]、または [数値] に設定されているフィールドのフォーマットをクライアントによって変更することはできません。

Asset Manager Application Designerの [入力タイプ] フィールドが次の値に設定されている場合は、追加のフィールド情報を表示できます。

表 3.3. [ユーザタイプ] フィールド値

Asset Manager Application Designerの [ユーザタイプ] フィールドの値	フォーマット
期間	<p>このフィールドでは、表示フォーマットと期間を表すデフォルトの単位を定義できます。</p> <p>シンタックス：</p> <p><code>%.D1]U1[]]d][%.D2]U2[]]d]]...[%.Dn]Un[]]d]]</code></p> <ul style="list-style-type: none"> ■ Uxは次の単位のうちの1つです。 <ul style="list-style-type: none"> ■ Y：年 ■ M：月 ■ D：日 ■ H：時間 ■ N：分 ■ S：秒 ■ 「d」は、単位を指定しない場合に適用する単位を指定します。 このパラメータを1つの単位にのみ割り当てます。 例：期間フォーマットが%M%Dd%Hで、値として10を入力した場合、この値は10日として変換されます。 ■ 「l」は、期間をLong型で表示することを指定します。 lを省略した場合、単位は短縮型で表示されます。 例：%Ylはyearまたはyears、%Yはyと表示されます。 ■ .Dxは、その単位に使用する小数点以下の桁数を指定します。 そのフォーマットの最小単位にのみこのパラメータを割り当てます。 例：期間フォーマットが%M%D%H%.2Nで、保存されている期間が15秒の場合、表示される値は0.25分になります。 <p>期間フォーマットが%Ml%D%Hd%.1Nの場合の例：</p> <ul style="list-style-type: none"> ■ 50を入力すると、Asset Managerは50時間とみなし、180000秒として保存します。 ■ 保存されている値が5,461,220秒の場合、Asset Managerは2 months 3d 5h 2.3minというように表示します。
システムリストデータ	リストデータ（ドロップダウンリスト）に表示される値
カスタムリストデータ	リストデータの名前

[dtLastModif] フィールド

SQL名が「dtLastModif」のフィールドは、Asset Managerデータベースのすべてのテーブルに存在します。

- SQL名：dtLastModif
- ラベル（フィールド名）：更新日
- 説明：変更日

このフィールドは、Asset Manager上で直接データを変更した場合や、データをインポートした時などに、Asset Managerデータベースのレコードが変更された時刻に更新されます。レコードの変更または作成の日付を示します。このフィールドに値をインポートすると、実際にインポートした日付ではなく、この値が使用されます。

【FullName】フィールド

【完全名】は、階層構造のテーブルに含まれているフィールドです。SQL名は「FullName」です。

【FullName】フィールドの構造

階層テーブル内の各レコードのFullNameフィールドでは、そのレコードの値の前に、親レコードからルートまでのフィールド値で構成される階層が付きます。各フィールドの値はスラッシュ (/) で区切られます（スペースは入りません）。階層の先頭と末尾にもスラッシュが付きます。

例：

- 場所のテーブル：場所の完全名は、場所名の前に親場所（複数可）の名前が付いた階層になります。

例: "/日本/東京/府中支社/"

- 部署と従業員（amEmplDept）のテーブル：従業員の完全名は、従業員の姓、名、IDの前に親レコード（複数可）の名前が付いた階層になります。

例: "/営業/テレマーケティング/田中,俊彦,P223/"

- 資産のテーブル：資産のフルネームは、その資産タグの前に親資産（複数可）の資産タグが付いた階層になります。

例: "/P123/DD456/CM0125/"

注意:

FullNameフィールドに直接入力することはできません。FullNameフィールドは、Asset Managerによって自動的に管理されます。

特殊なケース

FullNameフィールドの値に既にスラッシュ (/) 記号が使われている場合、Asset Managerはこの記号をハイフン (-) に置き換えます。

例：部署と従業員のレコードで、部署名が「営業/マーケティング」の場合は、FullNameフィールドの値は次のようになります。/A.../営業-マーケティング/B.../

リンクについて

このセクションでは、Asset Managerデータベース内のリンクについて説明します。内容は次のとおりです。

- リンクの表記名
- リンクのタイプ
- リンク数
- 中間テーブル
- 状況依存リンク

リンクの表記名

Asset Managerテーブル内の各リンクには、次の表記名が使われます。

- **SQL名**：Asset Managerのすべての言語のバージョンで同じSQL名を使っています。
- **ラベル**
- **説明**

[ポートフォリオ品目] テーブルから [設置場所] (SQL名：Location) へのリンクの例は次のとおりです。

- ソーステーブル ([ポートフォリオ品目] テーブル) のソースフィールドのSQL名は「ILocaId」です。
- ターゲットテーブル ([場所] テーブル) のSQL名は「amLocation」です。
- ターゲットテーブル ([場所] テーブル) のターゲットフィールドのSQL名は「ILocaId」です。
- 「database.txt」ファイルでリンク数は「1」に指定されています。これは、1件のポートフォリオ品目につき1つの設置場所のみが対応することを示しています。

このリンクの場合は、「ソースフィールド=ターゲットフィールド」と解釈することができます。

**警告:**

リンクのラベルおよび説明は、ターゲットテーブルのラベルおよび説明とは異なります。Asset Managerのテーブル間には複数のリンクが存在する場合があります。例えば、[ポートフォリオ品目] テーブルと、[部署と従業員] (amEmplDept) テーブル間には、資産のユーザを定義する [ユーザ] (SQL名: User) というリンクと、品目の責任者を定義する [責任者] (SQL名: Supervisor) というラベルのリンクが存在します。

リンクのタイプ

Asset Managerの既存のリンクのタイプは、次の表のとおりです。

表 3.4. 既存のリンクのタイプ

タイプ	説明
Normal	ソースレコードを削除すると、リンクも削除され、ターゲットレコードがソースレコードを参照しなくなります。
Own	ソースレコードを削除すると、リンク先のターゲットレコードも削除されます。
Define	ソースレコードとターゲットレコードがリンクしている限り、ソースレコードを削除できません。
Neutral	ソースレコードを削除すると、リンクも削除されます。ターゲットレコードの情報は更新されません。
Copy	ソースレコードを複製すると、このタイプのリンクも複製されます。
Owncopy	「Own」と「Copy」の両方のリンク属性が適用されます。

リンクによって保存される情報の属性は、次の表のとおりです。

表 3.5. リンクにより保存される情報の属性

情報の属性	説明
Normal	他の「UserType」のリンクが保存する情報以外のデータを保存します。
Comment	コメントフィールドを保存します。
Image	画像を保存します。
History	コンテキストリンクの一部を表します。
Feature values	任意管理項目の値を保存します。

リンク数

Asset Managerでは、1つのテーブルAに対して次の2つのタイプのリンクを定義できます。

- 「1-->1」リンク：テーブルAの1つのレコードはテーブルBの単一のレコードにしかリンクできません。例えば、[部署と従業員] テーブルを [場所] テーブルに「1-->1」リンクでリンクすると、1人の従業員が1つの場所に関連づけられます。
- 「1-->N」リンク：テーブルAの1つのレコードはテーブルBの複数のレコードにリンクできます。例えば、[契約] テーブルを [資産] テーブルに「1-->N」リンクでリンクすると、1人の契約が複数の資産に関連づけられます。

注意：理論的には、データベース内のテーブル間には次の3種類のリンクが存在します。

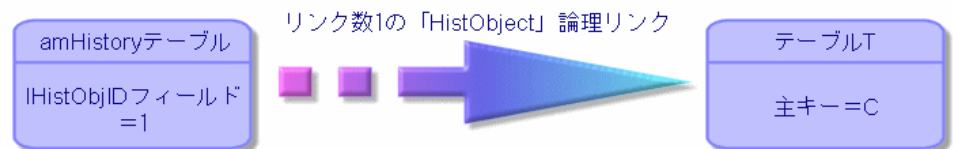
- 1リンク：テーブルAの1つのレコードとテーブルBの1つのレコードを双方向にリンクできます。この場合は、2つの「1-->1」リンクが存在することになります。
- nリンク：テーブルAの1つのレコードをテーブルBの複数のレコードにリンクできますが、テーブルBの各レコードはテーブルAの単一のレコードにしかリンクできません。この場合は、1つの「1-->1」リンクと1つの「1-->N」リンクが存在することになります。
- n-nリンク：テーブルAの1つのレコードをテーブルBの複数のレコードとリンク、または逆にテーブルBの1つのレコードをテーブルAの複数のレコードにリンクできます。この場合は、2つの「1-->N」リンクが存在することになります。

状況依存リンク

ターゲットテーブルが事前に定義されていないのに、ソーステーブル内で指定されることがあります。これは状況依存リンクと呼ばれます。このリンクのリンク数は1で、逆方向のリンクはありません。

例：[履歴] テーブルとターゲットテーブル間のリンク

図 3.1. [履歴] テーブルとターゲットテーブルとのリンク



リンクから得られるテーブル

中間テーブル

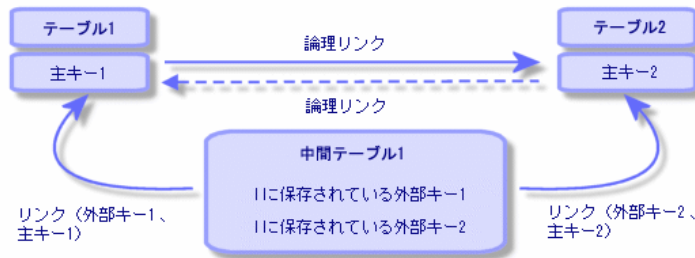
中間テーブルは、「n-n」タイプの論理リンクの場合にのみ使われます。

このテーブルは、Asset Managerのインタフェースには表示されず、テーブル間を論理的に関連付けるテーブルとして存在します。

通常のテーブルとは異なり、中間テーブル（リレーショナルテーブルとも呼ばれます）には主キーがありません。

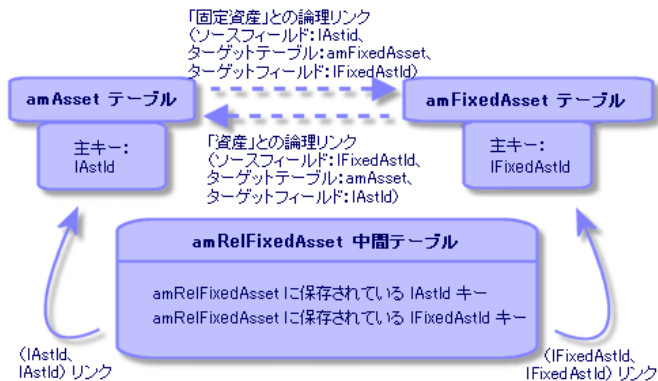
次に、中間テーブルと2つのテーブルの関係図を示します。

図 3.2. 中間テーブルの役割



例えば、「資産」テーブルと「固定資産」テーブルとのリンクは次のようになります。

図 3.3. 「資産」テーブルと「固定資産」テーブルとのリンク



この場合は次のようになります。

- 1件の資産に、次のように複数の固定資産をリンクできます（「**関連固定資産**」論理リンク（オブジェクトのSQL名：**FixedAssets**））。
 - [資産] テーブルの各レコードは、中間テーブル内の複数のレコードにリンクできます。
 - 中間テーブルの各レコードは、それぞれ [固定資産] テーブルの1つのレコードにリンクします。
- 1件の固定資産は、次のように複数の資産に関連付けることができます（「**固定資産**」論理リンク）。
 - [固定資産] テーブルの各レコードは、中間テーブル内の複数のレコードにリンクできます。
 - 中間テーブルの各レコードは、それぞれ [資産] テーブルの1つのレコードにリンクします。

注意:

中間テーブルは、Asset Managerのユーザインタフェースには表示されませんが、複雑なクエリを作成する場合に時々必要になります。

オーバーフローテーブル

これらのテーブルは以下のものを格納します。

- 元のテーブルに含まれない追加情報
- オーバーフローリンク（**オーバーフロー**タイプのリンク）の外部キー

オーバーフローテーブルの概要

テーブルがオーバーフローテーブルとなるのは、オーバーフローリンクを参照している場合です。

データベース構造には4つのオーバーフローテーブルがあります。

- 1 [ソフトウェアインストール] (**amSoftInstall**) テーブル
- 2 [電話] (**amPhone**) テーブル
- 3 [コンピュータ] (**amComputer**) テーブル

これらのテーブルはポートフォリオ品目テーブル (**amPortfolio**) の拡張であり、その外部キーを持ちます。

- 4 [資産] (**amAsset**) テーブル

このテーブルはポートフォリオ品目テーブル (**amPortfolio**) の拡張です。外部キーは**amPortfolio**テーブルにあり、ポートフォリオ品目リンクは外部キーを共有しています。

オーバーフローテーブルのリンクは、参照テーブルを指すかどうかによって動作が異なります。

参照テーブルは次のとおりです。

- [ポートフォリオ品目] (amPortfolio) テーブル
- [契約] (amContract) テーブル
- [作業指示] (amWorkOrder) テーブル
- [ケーブル] (amCable) テーブル
- [研修] (amTraining) テーブル

オーバーフローテーブルの詳細については、『ポートフォリオ』ガイドを参照してください。

注意:

オーバーフローリンクの場合、オーバーフローリンクが入力されていない場合に対応した妥当性スクリプトを追加することを強くお勧めします。例：
amPortfolio:Computerが無関連になる場合

```
Retval = [amPortfolio:Model.Nature.OverflowTbl]<>"amComputer"
```

参照テーブルを指しているオーバーフローリンクの動作

参照テーブルを指しているオーバーフローテーブルのそれぞれに対して：

- オーバーフローリンクが無関連の場合を除いて、オーバーフローテーブルでレコードを作成すると、参照テーブルでレコードが作成されます。
例えば、コンピュータ (amComputer) オーバーフローテーブルでは、オーバーフローリンクはコンピュータグループに無関連です。したがって、ポートフォリオ品目テーブルにレコードは作成されません。
- 以下の場合に、参照テーブルでレコードを作成したときにオーバーフローテーブルでレコードが作成されます。
 - このレコードのモデルの属性のオーバーフローテーブルフィールドは、オーバーフローテーブルを参照します。

他のテーブルを指しているオーバーフローリンクの動作

他のテーブルを指しているオーバーフローテーブルのそれぞれに対して：

- オーバーフローテーブルでレコードを作成しても、参照テーブルでレコードが作成されることはありません。
- 参照テーブルでレコードを作成しても、オーバーフローテーブルでレコードが作成されることはありません。

オーバーフローテーブルの外部キー

参照テーブルを指しているオーバーフローテーブルのそれぞれに対して：

- 外部キーはオーバーフローテーブルにあります

オーバーフローテーブルに対するクエリを実行するスクリプトを参照テーブルに作成する場合、対応するオーバーフローリンクをチェックするテスト用クエリを必ず最初に実行してください。

実装に関するアドバイス

オーバーフローテーブルを作成する際には、以下のことを守ってください。

- オーバーフローリンクに対応する属性とモデルを作成します。
- デフォルトで入力されない必須フィールドが、オーバーフローリンクに依存するページに表示されていることを確認します。

オーバーフローテーブル作成の例

自動車テーブル（amCar）をポートフォリオ品目（amPortfolio）テーブルのオーバーフローテーブルとして作成します。

手順は次の通りです。

- 新しいテーブルとオーバーフローリンクをAsset Manager Application Designerで作成します。
 - その動作をAsset Managerで確認します。
- 1 自動車テーブルを作成するには：
 - 1 データベースにテーブルを追加します（[データベース/テーブルを追加]）。
 - 2 次のようにフィールドに入力します。
 - **SQL名**：am Car
 - **ラベル**：自動車
 - 3 必要なフィールドを追加します
 - 2 オーバーフローリンクを作成するには：
 - [amCar] テーブルで、リンク [ポートフォリオ] を作成します（[データベース/リンクを追加]）。
 - 次のようにフィールドに入力します。
 - **リンクタイプ**：オーバーフロー
 - **ソースフレーム**、**SQL名**：Portfolio
 - **ターゲットフレーム**、**テーブル**：ポートフォリオ品目（amPortfolio）、**SQL名**：Car
 - [このリンクのインデックスを作成] を選択します。
 - 3 オーバーフローテーブルのページを作成します。

次のようにフィールドに入力します。

 - [全般] タブ：
 - 1 **SQL名**：scrGenCar
 - 2 **ラベル**：全般
 - [内容] タブ：
 - 1 [フィールドのリスト] フレームに、リンク「Portfolio.Model」と、必要なフィールドを追加します。

- 4 オーバーフローリンクの画面を作成するには：
次のようにフィールドに入力します。
 - [全般] タブ：
 - 1 **SQL名**：scrCar
 - 2 **ラベル**：自動車
 - 3 **説明**：Cars
 - [ページ] タブ：
 - 1 [詳細ページ] フレームに、scrGen carとPgGenDocのページを追加します。
- 5 データベースを保存します。
- 6 Asset Managerを開き、[属性] 画面を表示します。
- 7 [オーバーフローテーブル] フィールドに新しいリストデータ値「**Car**」が入っていることを確認します。
- 8 自動車テーブル (amCar) に作成された項目のそれぞれに対して、対応する項目がポートフォリオ品目テーブル (amPortfolio) に作成されていることを確認します。
- 9 作成されたポートフォリオ品目のそれぞれに対して、対応する項目が自動車テーブル (amCar) に作成されていることを確認します。

インデックスについて

このセクションでは、Asset Managerデータベースのインデックスについて説明します。内容は次のとおりです。

- インデックスの表記名
- インデックス付きフィールドの値の属性

インデックスの表記名



インデックスは、次の情報で表記されます。

- **SQL名** (末尾に「Id」が付きます) : SQL名は英語です。Asset Managerのすべての言語のバージョンで同じSQL名を使っています。
- ラベル
- 説明

インデックス付きフィールドの値の一意性

インデックス付きフィールドの値は、インデックスの属性によって複製が可能な場合と不可能な場合があります。

Asset Manager Application Designerでは、インデックスの左側にあるアイコンからインデックスの属性が分かります。

- アイコンなし：制限はありません。
- ：同じインデックスが付いたフィールド一式と同じフィールドが、テーブル内に作成されることはありません。
- ：NULL値が繰り返される場合を除き、同じインデックスが付いたフィールド一式と同じフィールドがテーブル内に作成されることはありません。

例えば、次のようなレベルでユーザ権限を設定できます。

【フィールドのヘルプ】（SQL名：amHelp）テーブルでは、次のフィールドに「Help_TableNameFiel」インデックスが付きます。

- 【テーブル】（SQL名：TableName）
- 【フィールド】（SQL名：FieldName）

インデックスは固有の値かまたはNULL値です。このため、同じテーブル内のあるフィールド用にヘルプが2つ検出されることはありません。

一方、これらのフィールドがすべてNULL値の場合は、そのレコードを複製することができます。

4 Asset Managerデータベースの作成、変更、削除

この章では、次の方法について説明します。

- 次の手順でAsset Managerデータベースを作成します。
 - 1 Asset Managerデータベースの作成の前提条件 [P. 45]
 - 2 DBMSを使って空のシェルを作成する [P. 46]
 - 3 データベースパスワードの暗号化 [P. 55]
 - 4 Asset Managerへの接続を作成する [P. 56]
 - 5 Asset Manager Application Designerを使ってデータベース構造を作成する [P. 57]
 - 6 Asset Manager Automated Process Managerを使用してデータベースに署名する [P. 62]
- 次の手順でAsset Managerデータベースの特定のパラメータを変更します。
 - Asset Managerクライアント言語を変更する [P. 63]
 - テキストフィールドで使用できる言語 [P. 68]
 - DBMSを変更する [P. 72]
- 次の手順でAsset Managerデータベースを削除します。

Asset Managerデータベースを削除する [P. 74]

Asset Managerデータベースの作成の前提条件

インストールが正常に終了するには、次のコンポーネントが次の2つの方法のいずれかでインストールされている必要があります。

- Asset Manager Application Designer（およびAsset Manager）が1つのマシンにインストールされ、Asset Manager Automated Process ManagerとHP AutoPassが別のマシンにインストールされている。
- Asset Manager Application Designer、Asset Manager Automated Process ManagerおよびHP AutoPassが1つのマシンにインストールされている。
- ▶ Asset Managerの『インストールとアップグレード』ガイド
また、「ライセンスキーをインストールする [P. 13]」で説明するように、ライセンスキーを取得しインストールする必要があります。

DBMSを使って空のシェルを作成する

作成の前に

警告および注意事項

- データベースの作成とは、高度なデータベースの管理能力を要する複雑な作業です。
このガイドは、データベース作成に関するすべての情報を記載している訳ではありません。詳細はDBMSのドキュメントを参照してください。ただし、データベース管理者が必要とする最低限の情報は以下で説明されています。

警告:

DBMSの使用を完全に習得していない場合は、専門のコンサルタントのサービスを使用する必要があります。データベースの作成、設定方法によっては、データのセキュリティ、Asset Managerの性能、およびアーキテクチャのスケラビリティに直接的な影響をもたらします。

データベースの作成方法は多数ありますが、以下の条件に応じて適切なものを選択します。

- DBMSがインストールされているプラットフォーム
- 企業内の規則
- 通常使用しているデータベース管理ツール
- 複数のユーザがデータベースを同時に使用する場合は、DBMSの性能を最適化するよう努めてください（カウンタレベルおよび識別子の生成）。データベース管理者がこの作業を行います。データベース管理者は、HPの技術サポート部に連絡を取り、指示を得る必要があります。
- 多数のサービス（例：ドメインコントローラ、DHCPコントローラ、DNS、メッセージサービス）をホストするサーバにDBMSをインストールしないようお勧めします。



注意:

後述の例および予測は、DBMSと共に提供される管理ツールを使った上で記述されています（サードパーティ社製のツールも存在します）。

データベースサーバの設定

デフォルトで「**RAID 10**」設定を選択します。これが不可能な場合、またはサーバが複数の「**RAID**」ディスククラスタで構成されている場合は、データベースを複数のディスクまたはディスククラスタに分散します。

複数の論理ボリュームにデータを作成し、ファイルグループを作成することにより、これを実行できます。

DBMSを正しく使用するには、書き込みロードと読取りロードをすべてのハードディスクに均一に分配する必要があります。

ハードディスクの速度が異なる場合、最も高速のハードディスクにインデックスを格納します。

RAID 0は、複数のハードディスクにアウトプット/インプットを分配するために最も容易な手段です。

参考情報：MS SQL Serverのパフォーマンスの比較結果は、数個の**RAIDS 10**クラスタを持つメモリが数GBのコンピュータを基にしています。

ログファイル(.log)を含むディスクには、**RAID 5**の代わりに一貫して**RAID 1**および**RAID 10**を使用します。

6個以上のハードディスクにデータが格納されている場合は、**RAID 5**を使用します。ただし、**RAID 10**を使用する場合よりも性能は低下します。

DBMSレベルでデータベースを作成する



注意:

Microsoft SQL ServerとUDB DB2では「データベース」という用語が使われていますが、Oracleでは「スキーマ」という用語が使われています。

データベースの作成手順

データベースの作成手順は次のとおりです。

- 1 データベース用のスペースを作成、確保する。
- 2 DBMSレベルでログインを作成する。
- 3 ユーザに権限を割り当てる。DBMSレベルでは、ログイン1つで十分です。ただし、他のDBMSログインを指定し、Asset Managerの接続で使用することも可能です。この場合、Asset Managerデータベースの各テーブルで変更権限を割り当てる必要があります。

サポートされる各DBMSのデータベースの作成方法は、下記を参照してください。

データベースの領域を確保する際の考慮事項

Asset Managerデータベースの構成要素は、おおよそ以下のとおりです。

- テーブル370個
- インデックス1600個

ヒント:

Asset Manager Application Designerを使って作成するデータベースを管理するためには、同時にオープン可能なオブジェクト数を指定するためのパラメータを、適切な値に設定しなければなりません。

Oracle

表 4.1. Oracleインスタンスを作成するために重要なサーバパラメータ

パラメータ	推奨値
CHARACTER SET	<ul style="list-style-type: none">■ アジア言語以外（英語、フランス語、ドイツ語、イタリア語、スペイン語、ロシア語など）でテキストを保存するには、非Unicode文字セット（UTF8の代わりに）を使用できます。■ アジア言語（日本語、簡体字中国語、韓国語）を含むすべての言語でテキストを保存するには、UTF8文字セットを使用：（▶テキストフィールドで使用できる言語 [P. 68]）■ 指定のアジア言語（日本語、簡体字中国語、韓国語）でテキストを保存するには、UTF8の代わりに、ユーザの言語に対応する文字セットを使用できます。<ul style="list-style-type: none">■ 欠点：テキストフィールドに保存できる文字数は、半分になります。■ 利点：UTF8文字セットを使用する場合よりもパフォーマンスが向上します。
	注意: Oracleの特定のバージョンでは、データベースの作成後にこのパラメータを変更することは不可能です。
db_block_size	このパラメータの単位はバイトです。 「8192」をお勧めします。
	注意: Oracleの特定のバージョンでは、データベースの作成後にこのパラメータを変更することは不可能です。

パラメータ	推奨値
db_block_buffers	<p>データベースのメモリキャッシュのサイズ</p> <p>このサイズは、db_block_sizeの単位で表記されています。</p> <p>例えば、必要なキャッシュが200 MBの場合：</p> <ul style="list-style-type: none"> ■ db_block_buffers = 25000 if db_block_size = 8192 ■ db_block_buffers = 100000 if db_block_size = 2048 <p>注意:</p> <p>中サイズのサーバの場合、メモリ全体の20から25%をデータベースのキャッシュ用のメモリに割り当てるようお勧めします。</p> <p>複数のユーザが同時にシステムにアクセスする場合は、上記の割合を調整してください。（サーバのメモリがスワップしないよう、データベースのキャッシュを多少縮小できます。）</p>
shared_pool_size	<p>このパラメータの単位はバイトです。</p> <p>このサイズは、Oracle 8.0.xの場合、ユーザ数（10～50）に応じて9,000,000から15,000,000バイトの範囲で変化します。</p> <p>ユーザがマルチスレッドサーバにインストールされている場合、このパラメータの値を増加する必要があります（Oracle 8iのlarge_pool_sizeと同様）。</p>
log_buffer	<p>このパラメータの単位はバイトです。</p> <p>「163840」をお勧めします。</p>
processes	<p><同時ユーザ数> + <Oracleシステムプロセス数>。</p> <p>注意:</p> <p>この値は、Oracleのライセンスで許可されている権限に応じて規定されます。</p> <p>例えば、10ユーザの場合、最大で「50」まで許容されます。</p>
dml_locks	500
open_cursors	<同時ユーザ数> * Max [30; <複数選択で更新するレコードの最大数>]
optimizer_mode	最初のインポート時には RULE モードを使用します。インポートが完了し、テーブル/インデックスの統計が計算された（Asset Manager Automated Process Managerの機能）時点で、 FIRST_ROWS モードまたは ALL_ROWS モードを使用します。
sort_area_size	<ul style="list-style-type: none"> ■ MTS以外のサーバの場合 <ul style="list-style-type: none"> ■ 「sort_area_size」はメモリのソート用のメモリのバイト数に当ります。 ■ 「sort_area_retained_size」は、メモリのソートの終わりに保存されるメモリに相当します。 <p>「65256」とまず指定し、必要に応じてこの値を増加します。</p>
sort_area_retained_size	<ul style="list-style-type: none"> ■ MTSサーバの場合、ソート用メモリ領域はすべての多重ユーザ接続を包括し、ソートが同じエリアで行われます。65536* 同時ユーザ数とまず指定し、必要に応じてこの値を大きくします。

Asset Managerを使ってデータベースを作成する前の確認事項

- 1 Oracleサーバを使用します。
- 2 Oracleのユーティリティを使って（例：Database Expander、Storage ManagerまたはDBA Studio）、Asset Managerデータベースの作成に必要な空き容量をテーブルスペースに確保します。例えば、Asset Managerをテーブルスペース1つ（Storageの値：INITIAL 10K、NEXT 10K）のみにインストールする場合、小さいデータベース用には150MBを確保し、資産5000件程度のデータベース用には450MBを確保します。
- 3 ロールバックセグメントは大幅に拡張できるものでなければなりません。同時ユーザ4人に対してロールバックセグメントを割り当てます。各ロールバックセグメントは、およそ25MBまで拡張可能である必要があります。INITIAL値（OPTIMAL：5MB）とNEXT値（OPTIMAL：20MB）には、256KBから10MBまでの差があります。これらの情報を、ユーザ企業の方針とAsset Managerデータベースのサイズに応じて調整しても構いません。
- 4 データベース用に30MBのメモリキャッシュを使用するようにOracleインスタンスを設定します。

Microsoft SQL Server

Microsoft SQL Server 7.0、2000または2005を準備する

SQL Server 7.0または2000を準備するには、Microsoft SQL Enterprise Managerユーティリティを使用します。SQL Server 2005に使用する管理ユーティリティは、**SQL Server Management Studio Express**です。

表 4.2. サーバとデータベースの重要なパラメータ

パラメータ	デフォルト値	推奨値
max server memory	OS memory	OSカーネルとプロセス用にOSメモリを確保します（Windows 2000の場合は128から256MB）。システムがスワップしてはなりません。
auto create statistics	データベースの作成時にデータベースのテンプレートで設定されている値	すべてのAsset Managerデータベースで有効にする必要があるデータベース設定
auto update statistics	データベースの作成時にデータベースのテンプレートで設定されている値	すべてのAsset Managerデータベースで有効にする必要があるデータベース設定

Microsoft SQL Server 7.0データベースを作成する

- 1 Asset Managerデータベースを作成するのに十分なサイズのデータファイルを使って、データベース（データベース名は任意）を作成します（サイズの小さいデータベース用には最低150MBのDATA、20MBのLOG）。

- 2 ログインの詳細で、データベースへの接続権限を割り当てます。

 **ヒント:**

どの接続権限を使用すべきか分からない場合は、ユーザログインに「**Database Owner**」を割り当てようお勧めします。

- 3 データベースのプロパティを表示し、次の操作を行います。
 - **[Permissions]** タブで、ログインに権限を割り当てます（少なくとも「**Create Table**」と「**Create Stored Procedures**」）。
 - ログファイルを保存しない場合は、**[Options]** タブの **[Truncate Log on Checkpoint]** チェックボックスをオンにします。保存する場合は、ディスクスペースの飽和状態を防止するプロセスを設置する必要があります。
- 4 **Tempdb**データベースの領域が少なくとも**20MB**であることを確認します。

データベースオプションを設定する

ログファイル (**trunc. log on chkpt**) に負荷がかからないようにする場合や、SQL Query Analyzer (**ANSI null default**) を使ってテーブルの作成スクリプトを作成するためには、以下のデータベースオプションが推奨されています。

- Sp_dboption <dbname>,'trunc. log on chkpt', true
- Sp_dboption <dbname>,'ANSI null default', true
- Sp_dboption <dbname>,'auto create statistics', true
- Sp_dboption <dbname>,'auto update statistics', true

サーバの標準設定を回復する

以下のSQLクエリを使って、サーバの標準設定を回復します（例えば、SQL Query Analyzerで実行）。

- Sp_configure
標準設定を回復します（割り当てられたメモリなど）。
- Select @@version
サーバのバージョンを回復します。
- Sp_helpsort
インデックスが物理的に使用するソート順を回復します。

クライアントコンピュータを準備する

SQL Serverのクライアント層を各クライアントワークステーションにインストールします。

- SQL Server 7.0または2000がインストールされている場合、ODBC SQL Serverドライバを使用します。

- SQL Server 2005がインストールされている場合、SQL Native ClientドライバまたはSQL Server ODBCドライバを使用します。SQL Native Clientドライバは、標準SQL Server ODBCドライバよりも優れたパフォーマンスを提供し、かつSQL Server 2005に特有の機能にアクセスできます。

 **重要項目:**

SQL Native Clientドライバの現行バージョンでは、特定のSQL機能（クエリ実行時）に影響するエラーが報告されています。

Asset Managerでこの問題に対処するには、[StmtCache] の値を0に設定します。[StmtCache] パラメータを使用して、SQL Serverに送信されるクエリのパフォーマンスを向上します。この値を0に設定すると、サーバパフォーマンスが低下します。

SQL Native Clientドライバを選択した場合、この処理は自動的に行われますが、クライアントワークステーションとこのドライバを使用しているサーバとの接続後であれば、Asset Manager「amdb.ini」ファイルの[接続]セクションで、[StmtCache] パラメータを手動で追加できます。これにより、パフォーマンスが向上します。

例：

```
StmtCache=30
```

DB2 UDB

ここで説明する情報は、DBMSの設定後、そしてDBMSでAsset Managerデータベースを作成する前に使用します。

- 以下で説明する操作は、DB2 UDBを使ってAsset Managerを正しく機能させるために必要な操作です。
- 以下のコマンドは、DB2 UDBのCommand Centerのシンタックスに対応します。次のソリューションを使用する場合は、コマンドを調整してください。
 - DOSセッション
 - DB2 UDBのCommand Center

サーバでデータベースを作成する

高品質のデータベースを維持するために、データベース管理者がデータベースエンジンを設定するようお勧めします。管理者は以下のSQLステートメントを使って、データベースのAPPLHEAPSZおよびAPP_CTL_HEAP_SZパラメータを増加することが必要です。

```
1 CREATE DATABASE <データベース名>
```



注意:

アジア言語（日本語、簡体字中国語、韓国語）を含むすべての言語でテキストを保存するには、**UTF-8文字セット**を**CODESET**パラメータで指定します（▶ **テキストフィールドで使用できる言語** [P. 68]）。例：

```
CREATE DATABASE <データベース名> USING CODESET UTF-8
```

1種類のアジア言語（日本語、簡体字中国語、韓国語）のみでテキストを保存するには、ユーザの言語（CODESETパラメータ）に対応する文字セットを使用できます。

- 欠点：テキストフィールドに保存できる文字数は、半分になります。
- 利点：UTF-8文字セットを使用する場合よりもパフォーマンスが向上します。

```
2 Update database configuration for <データベース名> using APPLHEAPSZ 2048
```

```
3 Update database configuration for <データベース名> using APP_CTL_HEAP_SZ 2048
```

```
4 Update database configuration for <データベース名>; using DBHEAP 4096
```

```
5 Update database configuration for <データベース名>; using LOGFILSIZ 500
```

```
6 Update database configuration for <データベース名>; using DFT_QUERYOPT 2
```

7 Asset Managerデータベースでアーカイブ機能を使用する場合、DB2は8Kページを処理できる必要があります。



注意:

デフォルトでは、DB2は4Kページしかサポートしません。

アーカイブ機能をサポートするには、「**SYSTEM**」タイプのテーブルスペースと、8Kの「**REGULAR**」タイプのテーブルスペースを作成する必要があります。

例：

```
db2 CREATE Bufferpool BP8K IMMEDIATE SIZE -1 PAGESIZE 8 K ;
```

```
db2 CREATE REGULAR TABLESPACE TB8K PAGESIZE 8 K MANAGED BY AUTOMATIC STORAGE BUFFERPOOL BP8K ;
```

```
db2 CREATE SYSTEM TEMPORARY TABLESPACE TB8KTMP PAGESIZE 8 K MANAGED BY AUTOMATIC STORAGE BUFFERPOOL BP8K ;
```

- 8 IDをデフラグするには、データベースが特定の一時ワークスペースを使用する必要があります。

一時ワークスペースをDB2で起動するには、次のコマンドを使用します。

```
db2 CREATE USER TEMPORARY TABLESPACE TEMPTBSPACE MANAGED BY AUTOMATIC STORAGE
```

 **注意:**

データベースIDのデフラグにより、レコードのキーとして使用されるID数が制限されます。

Asset Manager Application Designerのコマンドラインバージョンを使用してIDをデフラグします。

シンタックスは次のとおりです。

```
amdbal -iddefrag:<接続名>;<管理者権限を有するユーザのログイン>;<ログインに関連付けられたパスワード>
```

DB2でAsset Managerクエリを処理できるようにするには、**STMTHEAP**パラメータ (SQL命令のヒープサイズ)に少なくとも**12288** (12KB) の値を設定する必要があります。

クライアントを準備する

まず、DB2のクライアント層をインストールします (DB2 Cli)。

DB2接続用にODBCソースを作成する必要はありません。ただし、他のアプリケーション用にODBCソースを宣言する必要がある場合は、**user data source**ではなく**system data source**を使用するように注意してください。**user data source**を使用すると、重要な最適化がキャンセルされてしまいます。

ODBCソースを使用せずに接続を作成するためには、以下のコマンドを検討します。

- CATALOG TCPIP NODE
- CATALOG DATABASE
- db2icrt

例

データベースサーバがWindowsサーバ「**CALIFORNIUM**」にインストールされています。データベース自体は「**INST_1**」インスタンスにあり (DB2は同時に複数のインスタンスを処理します)、名前は「**SAMPLE**」です。このデータベースにアクセスする前に実行するコマンドは、以下のとおりです (**DB2**

Command Line Processorで実行)

- 1 `CATALOG TCPIP NODE MyNode REMOTE CALIFORNIUM SERVER 50000 REMOTE_INSTANCE INST_1 SYSTEM CALIFORNIUM OSTYPE NT`
- 2 `CATALOG DATABASE SAMPLE AS SAMPLE AT NODE MyNode`
- 3 次に実行する標準コマンドライン：
`db2icrt MyNode`

DBMSログインを作成する

Asset Managerのすべてのユーザが共通のDBMSログインを使用できます。この場合、Asset Managerでユーザプロファイルを使ってアクセス権限を管理します。

また、データベースに対して制約された権限を持つ複数のDBMSログインを使うようにAsset Managerを設定することもできます。これにより、外部ツールを使ってデータベースへのアクセスを制御できるようになります。このガイドでは、複数ログインの使用の説明は省略されています。

Oracleの例

少なくとも**CONNECT**、**RESOURCE**および**CREATE SESSION**権限を含むOracleログインを宣言します。既にOracleでAsset Managerデータベースが作成されている場合は、新しいデータベースをホストするために新しいOracleスキーマを作成します。

スキーマを作成するためのOracle SQLスクリプトの例

```
connect system@ORASERV; create user Asset Manager identified by <password> default tablespace <asset> temporary tablespace <temporary_data> profile default; grant connect, resource, create session to Asset Manager;
```

データベースパスワードの暗号化

Asset Managerは、これまで独自のセキュリティアルゴリズムを使用して、「amdb.ini」ファイルに格納されているデータベースパスワードを暗号化していました。バージョン9.30では、サインインセキュリティがより強い業界標準の**AES 256**アルゴリズムで強化されています。

- Windowsクライアントメニューの【ファイル／接続の管理...】を使用してデータベースへの新規接続を作成するか、既存の接続のサインイン詳細を変更する場合、「amdb.ini」ファイルに格納されているパスワードは、AES 256を使用して暗号化されます。

- 「amdb.ini」から既存のデータベース接続を開く場合、Asset Managerは以前の暗号化システムと後方互換性があり、古いパスワードを使用して接続できます。

AES 256暗号化を使用して既存のデータベース接続のセキュリティを強化する場合、古いパスワードを変更する必要があります。このため、弱い方法で暗号化された古いパスワードは使用できなくなります。

新しい方法の詳細は、

http://en.wikipedia.org/wiki/Advanced_Encryption_Standardを参照してください。

Asset Managerへの接続を作成する

空のシェルへの接続を指定する手順

- 1 Asset Managerを起動します
- 2 [データベースに接続] ウィンドウが表示されます。
データベースに接続せずにこのウィンドウを終了します（[キャンセル] ボタン）。
- 3 接続リストを表示します（[ファイル/接続の管理...]）。
- 4 接続の作成ページを表示します（[新規作成] ボタン）。
- 5 接続の詳細を入力します。
- 6 接続を作成します（[作成] ボタン）。
 - ▶ MS SQL Serverを使用してデモ用データベースへの接続を作成するための情報 [P. 57]
- 7 データベース構造がまだ作成されていないため、この段階で接続をテストしないでください（[テスト] ボタン）。
- 8 データベース構造がまだ作成されていないため、この段階で接続を開かないでください（[開く] ボタン）。
- 9 すべてのウィンドウを閉じます。
- 10 Asset Managerを終了します。

注意:

Windows XPを使用している場合、接続を宣言するためには、書き込み権限で接続しなければなりません。

MS SQL Serverを使用してデモ用データベースへの接続を作成するための情報

フィールド	値
名前	<名前>
エンジン	Microsoft SQL Server
データソース	AMDemo93en (SQL Server)
データベース	<MS SQL Serverエディタで作成した空のシェルの名前>
ユーザ	itam
パスワード	password
所有者	<このフィールドは空にしておくこと>

Asset Manager Application Designerを使ってデータベース構造を作成する

Asset Manager Application Designerを使用して、DBMSで作成した空のシェルの中にAsset Managerデータベース構造を作成します。

この構造は、Asset Managerとともにインストールされている記述ファイルに基づいています。

データベース構造の作成手順

- 1 Asset Manager Application Designerを起動します。
- 2 次の手順でデータベース記述ファイルを開きます。
 - a **【開く】** ウィンドウを表示します（**【ファイル／開く...】**）。
 - b **【データベース記述ファイルを開く（新規データベースの作成）】** オプションを選択します。
 - c オプションを確定します（**【OK】** ボタン）。
 - d 「gbbase.xml」ファイルを選択します（Asset Manager インストールフォルダの「config」サブフォルダ、通常は「C:\Program Files\HP\Asset Manager 9.30 xx\config」にあります）。
 - e ファイルを開きます（**【開く】** ボタン）。
- 3 アジア言語（日本語、簡体字中国語、韓国語）を含むすべての言語でテキストを保存するには、次の手順に従います。
 - 1 **【データベース／UNICODEテキストフィールドの使用】** メニューを選択します。
 - 2 「gbbase.xml」ファイルを保存します（**【ファイル／保存】** メニュー）。

 **警告:**

OracleおよびDB2では、このオプションは、データベースがUTF8文字セットを使用する場合のみ使用可能になります。▶ [DBMSレベルでデータベースを作成する \[P. 47\]](#)

 **注意:**

1種類のアジア言語（日本語、簡体字中国語、韓国語）のみでテキストを保存する場合、このオプションは必要ありません。

- 欠点：テキストフィールドに保存できる文字数は、半分になります。
- 利点：UTF8文字セットを使用する場合よりもパフォーマンスが向上します。

- 4 データベース構造を作成するウィザードを起動します（[アクション/データベースを作成...]メニュー）。
- 5 ウィザードの各ページで必要な情報を入力します（[次へ] および [前へ] ボタンでウィザードページを移動します）。
▶ [データベースの作成ウィザードのパラメータ \[P. 59\]](#)

 **注意:**

ユーザのDBMSとしてOracleを使用し、かつデータのインポート時にエラーメッセージが表示される場合、Windowsのレジストリにある [NLS_LANG] パラメータの値をチェックします。

NLS_LANGはAsset Manager Application Designerインストール言語と一致する必要があります。手順は次のとおりです。

- 1 DOSコマンドプロンプトを開き（Windowsメニューの [スタート/実行...] ）、次のコマンドを入力します。

```
regedit
```

これにより、レジストリエディタが開始します。

- 2 例えば、Oracleバージョン10gを使用している場合、次のブランチを展開：
「My computer\ HKEY_LOCAL_MACHINE\ SOFTWARE\ ORACLE\ KEY_OraClient10g_home2」

他のOracleバージョンに使用するブランチ名については、次のサイトの情報を参照してください。

http://www.oracle.com/technology/tech/globalization/htmldocs/nls_lang%20faq.htm#_Toc110410552

- 3 変更するには、 [NLS_LANG] パラメータをダブルクリックします。
例えば、Asset Manager Application Designerが英語でインストールされている場合、値「**AMERICAN_AMERICA.WE8MSWIN1252**」を入力します。
- 4 変更を保存するには、 [OK] をクリックします。

- 6 ウィザードで定義したオプションを実行します（ [完了] ボタン）。

データベースの作成ウィザードのパラメータ

[SQLスクリプトの生成/データベースの作成] ページ：

フィールド	値
データベース	Asset Managerで作成した空のシェルへの接続を選択します。 ▶ Asset Managerへの接続を作成する [P. 56]

フィールド	値
作成	<ul style="list-style-type: none"> ■ 【データベースを作成】：標準のSQLスクリプトでデータベースを作成したい場合は、このオプションを選択します。 ■ 【SQLスクリプトを生成】：標準のSQLデータベース作成スクリプトを生成してスクリプトを変更したい場合は、このオプションを選択します。 この場合は次のようになります。 <ol style="list-style-type: none"> 1 データベース作成ウィザードの実行を完了します。ここでは単にSQLスクリプトファイルが作成されます。 2 SQLスクリプトを変更します。 3 データベース作成ウィザードをもう一回実行します。ここではオプションの【スクリプトを使用してデータベースを作成】を選択します。 ■ 【スクリプトを使用してデータベースを作成】：【SQLスクリプトの生成】オプションで作成し、その後変更を加えたSQLスクリプトを使用してデータベースを作成したい場合は、このオプションを選択します <p>注意: ウィザードの後続のページは、ここで選択したオプションによって変わります。</p>
高度な作成オプションを使用	<p>テーブル、インデックス、およびLOBのテーブルスペースを定義する必要がある場合は、このオプションを選択します。</p> <p>注意: DBMSを使用してテーブルスペースをあらかじめ作成しておく必要があります。</p>

前のページで **【SQLスクリプトを生成】** を選択した場合は、**【スクリプトの選択】** ページが表示されます。

フィールド	値
ファイル	作成するSQLスクリプトファイルのフルパスを入力します。
区切り文字	<p>次のようなSQLステートメントの区切り文字を選択します。</p> <ul style="list-style-type: none"> ■ /：Oracleデータベースの場合 ■ GO：その他のすべてのデータベース ■ その他の区切り文字：有効な区切り文字のみ（;など） <p><MySeparator>というように、ユーザ定義の区切り文字を定義できますが、データベース作成スクリプトは機能しなくなります。</p>

前のページで **【データベースを作成】**、**【スクリプトを使用してデータベースを作成】**、または **【専門分野データをインポート】** を選択した場合は、**【作成パラメータ】** ページが表示されます。

フィールド	値
作成スクリプト	作成するSQLスクリプトファイルのフルパスを入力します（ウィザードの最初のページで 【データベースを作成】 オプションを選択すると、このフィールドは表示されません）。
パスワード	管理者ログインに関連付けるパスワード。 注意: Asset Managerデータベース管理者は、 【部署と従業員】 (amEmplDept) テーブルに記録され、その 【名前】 (Name) フィールドは Admin になります。 データベース接続ログインは 【ユーザ名】 (UserLogin) フィールドに保存されます。管理者名は「 Admin 」です。
確認	【パスワード】 フィールドに入力した値を確認します。

前のページで**【データベースを作成】** または **【スクリプトを使用してデータベースを作成】** を選択した場合は、**【システムデータの作成】** ページが表示されます。

フィールド	値
タイムゾーンの使用	このオプションを選択すると、データベースサーバ、日付フィールド、Asset Manager クライアントのタイムゾーンの時間差を管理できます。 ▶『 高度な使い方 』ガイドの 「タイムゾーン」 の章。
サーバ	Asset Managerデータベースサーバのタイムゾーンを選択します。 この場合は、例えば、サーバのWindowsコントロールパネルで定義されたタイムゾーンになります。
データ	データベースに保存された日付を表示するタイムゾーンを選択します。
フィールドのヘルプの使用	このオプションを選択すると、データベースフィールドとリンクに関する情報が挿入されます。

前のページで**【専門分野データをインポート】** を選択した場合は、**【インポートするデータ】** ページが表示されます。

フィールド	値
使用可能データ	データベースにインポートするサンプルデータまたは専門分野データを選択します。 本番データベースの場合は、以下を推奨します。 <ul style="list-style-type: none"> ■ 使用したいモジュールの専門分野データを選択します。 ■ サンプルデータを選択しないでください。 注意: データベースの作成時にデータをインポートしなくても、あとから簡単にインポートすることができます。

フィールド	値
エラー発生時にインポートを中止	問題が発生した場合にインポートを中止するには、このオプションを選択します。
ログファイル	エラーや警告などすべてのインポート操作を記録するファイルの完全名。

データベース構造の更新

Asset Manager Application Designerでは、マイグレーションインポートを実行することなしに、データベースを旧バージョンから、新バージョンにアップグレードできるようになりました。

Asset Managerデータベースのアップグレードに関する詳細は、『マイグレーション』ガイドを参照してください。

Asset Manager Automated Process Managerを使用してデータベースに署名する

Run Asset Manager Automated Process Managerを実行し、データベースが定期的に署名されるように、データベースサーバに信号送信 (UpdateToken) モジュールを設定します ([ツール/モジュールの設定] メニュー) 。

注意:

これを実行するのは、データベース用にアクティブになっているライセンスキーと一致するライセンスキーのセットを使用して、HP AutoPass License Management Toolがインストールされているマシン上である必要があります (HP AutoPassはAsset Manager Automated Process Managerで自動的にインストールされますが、ライセンスキーをHP AutoPassのこのインスタンスにインストールするのは手動の手順です (▶ [HP AutoPassライセンスキーをインストールする \[P. 14\]](#))) 。

署名プロセスでは、HP AutoPassによって付与されたライセンスキーとAsset Manager Automated Process Managerが接続されているデータベースでアクティブにされたライセンスキーが比較されます。前者のキーが後者のキーより少ない場合、データベースは署名されません。

データベースが1週間以上未署名のままであると、データベースはロックされます。

データベースをアンロックするには、次の手順に従います。

- 1 HP AutoPassとデータベース間のライセンスキーの不一致を解決します。
これを行うには、次のいずれかの手順を実行できます。

- 不足しているライセンスキーをAsset Manager Automated Process ManagerマシンのHP AutoPassインスタンスに追加します。
 - ▶ HP AutoPassライセンスキーをインストールする [P. 14]
 - またはデータベースから余分なライセンスキーを削除します。
 - a Asset Manager Application Designerでデータベースをアクティブにするマシン上で、HP AutoPass License Management Toolを実行し、**【ライセンスキの削除】**を使用します。
 - b 削除するライセンスキーを選択します。
 - c **【削除】**をクリックします。
 - d 正しいセットのライセンスキーでデータベースを再度アクティブにします (▶ HP AutoPassライセンスキーファイルを変更する [P. 17]) 。
 - または一からライセンスキーを再インストールします。
 - a Asset Manager Application Designerでデータベースをアクティブにするマシン上で、HP AutoPass License Management Toolインストールフォルダの「data」フォルダ (デフォルトでは、「C:\Program Files\HP\Asset Manager 9.30 xx\autopass\data」) にある「LicFile.txt」を削除します。
 - b 必要なライセンスキーをAsset Manager Automated Process ManagerマシンのHP AutoPassインスタンスに追加します。
- 2 Asset Manager Automated Process Managerを実行し、データベースサーバに信号送信モジュールを再度実行します。
データベースはアンロックされている必要があります。

Asset Managerクライアント言語を変更する

概要

Asset Managerのデータベースオブジェクトの一部はマルチリンガルです。

▶ オブジェクトの属性に基づく表示言語 [P. 66]

Asset Managerデータベースのマルチリンガルオブジェクトをクライアントレベルで使用するためには、以下を実行する必要があります。

- 1 最初に、対応する言語パラメータをデータベースに追加します。
 - ▶ 言語パラメータの追加 [P. 64]
- 2 次の手順を実行します。
 - Windowsクライアント：対応する言語のバージョンのAsset Managerクライアントをユーザのワークステーションにインストールします。マルチリンガルデータベースオブジェクトは自動的にクライアントの言語で表示されます。

- ▶ **Asset Manager**クライアントレベルでの特定言語の表示 [P. 65]
- **Webクライアント**：対応する言語のバージョンの**Asset Manager Web**（**Asset Manager Web Service**で使用される**Asset Manager Web Tier**および「**aamapi93.dll**」）をインストールします。マルチリンガルデータベースオブジェクトは自動的に**Webクライアント**で表示されます（URLは言語固有のバージョンの**Asset Manager Web**をポイントします）。
 - ▶ **Asset Manager**クライアントレベルでの特定言語の表示 [P. 65]
 - ▶ 『インストールとアップグレード』ガイドの「**Asset Manager Web**のインストール、設定、削除、更新」の章、「**Asset Manager Web**のインストール/**Asset Manager Web**のインストールの準備/インストールするコンポーネント」のセクションを参照してください。

Asset Managerソフトウェアスイートでサポートされる任意の言語が使用できません。

例えば、英語クライアントからもスペイン語クライアントからもデータベースにアクセスできます。

言語パラメータの追加

Asset Manager Application Designerでデータベースを作成する際には、**Asset Manager Application Designer**の言語バージョンに対応するパラメータが用いられます。

他の言語パラメータを追加するには：

- 1 どの言語をサポートするかを決めます。
- 2 追加する言語のそれぞれに対して**Asset Manager**をインストールします。

ヒント:

Asset Manager Application Designerだけをインストールすることもできます。

- 3 選択した言語で**Asset Manager Application Designer**を開始します。
- 4 データベースを開きます（[ファイル/開く.../既存のデータベースを開く]メニュー）。
- 5 追加する言語のそれぞれに対して：
 - 1 [アクション/データベースに言語を挿入...]を選択します。
 - 2 適切な言語のバージョンの**Asset Manager**がインストールされているフォルダを選択します。
例：C:\Program Files\HP\Asset Manager 9.30 xx-English
 - 3 [OK] をクリックして確定します。



警告:

すでに使用されている言語のパラメータを追加すると、追加されたオブジェクトに関するものを含めて、すべてのカスタマイズが上書きされます。

WindowsクライアントでのANSIコードページを使用しない言語を表示するフォントの置換

すべてのユーザインタフェース項目（データ、メニュー、ウィザード、スクリプト、ナビゲーションバーなど）をANSIコードページを使用しない言語（非西洋言語）で正しく表示するには、次を行います。

- ユーザインタフェース項目の表示に使用されるフォントに置換：

- 1 テキストエディタで「am.ini」ファイルを開きます。

このファイルの場所：▶ 『インストールとアップグレード』ガイドの「**.ini** および**.cfg**ファイル」の章。

- 2 対応するセクションに以下の行を追加します。

```
[FontSubstitutes] Microsoft Sans Serif=Arial Unicode MS MS Sans Serif=Arial Unicode MS Times New Roman=Arial Unicode MS Verdana=Arial Unicode MS Arial=Arial Unicode MS
```

```
[Option] /ShortcutFavorite/Extended/SHORTCUT_FAVORITE_FONTNAME=Microsoft Sans Serif /Advanced/UseHardCodedFontMapper=1 /Advanced/ScriptFont=Arial Unicode MS, 12
```

- 3 変更を保存します。

- WindowsシステムフォントがUnicode対応であることを確認します。

- 1 Windowsの表示プロパティ（ [コントロールパネル/画面] ）、 [デザイン] タブで、 [詳細設定] をクリックします。
- 2 異なるインタフェース項目に使用されるフォントがUnicodeをサポートすることをチェックします（例えば、 [Arial Unicode MS] フォントを選択）。

Asset Managerクライアントレベルでの特定言語の表示

言語パラメータが管理者によってAsset Managerデータベースに追加されたら：

- 1 ■ Windowsクライアント：必要な言語でAsset ManagerのWindowsクライアントをインストールします。
 - Webクライアント：対応する言語のバージョンのAsset Manager Web（Asset Manager Web TierとAsset Manager Web Service）をインストールします。

- 2 ■ Windowsクライアント : Asset Manager Windowsクライアントを起動します。
 - Webクライアント : Internet Explorerを起動し、必要な言語の Asset Manager Web TierのインスタンスをポイントするURLを入力します。
- 3 データベースに接続します。
マルチリンガルデータベースオブジェクトは自動的にクライアントの言語で表示されます。

 **注意:**

Asset Managerクライアントのレベルでは、マルチリンガルオブジェクトは1つの言語でしか表示できません。すなわちWindowsクライアントまたはAsset Manager Web Tier自身の言語です。

オブジェクトの属性に基づく表示言語

Asset Managerでのオブジェクトの表示言語は、オブジェクトの属性に依存します。

表 4.3. Asset Manager Windows クライアント - ラベル、画面、ページなどの表示言語

オブジェクトの属性	<ul style="list-style-type: none"> ■ テーブル、フィールド、リンク、インデックスのラベルと説明 ■ 画面ラベル ■ ページラベル ■ システムリストデータ値 ■ フィールドまたはリンクの状況依存ヘルプ
オブジェクトのテキストの保存	データベース
Asset Manager Windows クライアント別の表示言語	データベースに接続されたAsset Managerクライアントの言語
Asset Manager Webクライアント別の表示言語	WebクライアントのURLがポイントするAsset Manager Webインスタンスの言語

表 4.4. Asset Manager Windows クライアント - データの表示言語

オブジェクトの属性	データベースに保存されたデータ
オブジェクトのテキストの保存	データベース
Asset Manager Windows クライアント別の表示言語	ユーザがデータを入力した言語

オブジェクトの属性	データベースに保存されたデータ
Asset Manager Webクライアント別の表示言語	<p>以下を除くユーザがデータを入力した言語</p> <ul style="list-style-type: none"> ■ 詳細画面の【金額】フィールドと【時刻】フィールド：読取りと編集モードでは、展開されたAsset Manager Web Tierの「AssetManager.war」ファイルの「webclient-faces-config.xml」ファイル内の<default-locale>%locale%</default-locale>パラメータに従って表示されます。 ■ 詳細画面の【日付+時刻】フィールド： <ul style="list-style-type: none"> 編集モード：フィールドの表示方法は、ユーザが変更できず、すべての言語バージョンのAsset Managerで同じです。 読取りモード：展開されたAsset Manager Web Tierの「AssetManager.war」ファイルの「webclient-faces-config.xml」ファイル内の<default-locale>%locale%</default-locale>パラメータに従って表示されます。 ■ 詳細画面の【期間】フィールド：読取りと編集モードでは、Asset Manager Web Tierの言語バージョンに依存します。 ■ リスト（メインリスト、ページリスト、およびウィザードリスト）内の【日付】フィールドと【日付+時刻】フィールド：フィールドの表示方法は、ユーザが変更できず、Asset Manager Web Serviceを実行しているユーザアカウントセッションに関連付けられている地域設定に依存します。 <p>警告:</p> <p>Asset Manager Web Serviceを実行しているユーザアカウントセッションに関連付けられている地域設定は、Asset Manager Web Serviceの言語と一致する必要があります。それ以外の場合、一部の文字列が望まないフォーマットで表示される可能性があります。</p>

表 4.5. Asset Manager Windows クライアント - ワンポイントの表示言語

オブジェクトの属性	ワンポイント
オブジェクトのテキストの保存	データベース
Asset Manager Windows クライアント別の表示言語	データベースの作成に用いられたAsset Manager Application Designerの言語
Asset Manager Webクライアント別の表示言語	なし

表 4.6. Asset Manager Windows クライアント - プログラムメッセージの表示言語

オブジェクトの属性	プログラムメッセージ
オブジェクトのテキストの保存	クライアントプログラム
Asset Manager Windows クライアント別の表示言語	データベースに接続されているAsset Managerクライアントの言語（データベースレベルで生成される一部のメッセージを除く）
Asset Manager Webクライアント別の表示言語	WebクライアントのURLがポイントするAsset Manager Webインスタンスの言語（データベースレベルで生成される一部のメッセージを除く）

一部のデータベースレコードには追加情報が必要です。

表 4.7. Asset Manager Windowsクライアント - レコードタイプに基づく表示言語

レコードタイプ	コメント
SAP Crystal Reports	レポートは、作成に用いられた言語が何であっても表示できます。ただし、ラベルはレポートの作成に用いられた言語で表示されます。ラベルを他の言語で表示したい場合は、レポートを複製して翻訳する必要があります。一部の文字列は、使用している「aamapi93.dll」ファイルの言語に依存する場合があります。
アクションとウィザード	レポートと同じ原則
クエリ	言語とは無関係
デモ用データベース	単一言語（Asset Managerのインストール言語）
ワークフロースキーム	他の言語パラメータも、本番データベースと同じ方法で追加できます。ワークフロースキームは、作成に用いられた言語と無関係に実行できません。ただし、ラベルはチャートの作成に用いられた言語で表示されます。ワークフロースキームを複製することはできません。ワークフローインスタンスも複製されてしまうからです。
オンラインヘルプ	オンラインヘルプはどの言語でも使用できます。デフォルトの言語は、ユーザのデスクトップにインストールされたクライアントの言語です。
ドキュメント	クライアントのデスクトップにAsset Managerをインストールする際には、インストール言語に一致する「pdf」ファイルだけがインストールされます。他の言語の「pdf」ファイルも必要な場合は、それらがインストールされているデスクトップから単にコピーしてください。

テキストフィールドで使用できる言語

以下を最初に実行することで、WindowsクライアントまたはWebクライアントを介して入力できるすべての言語（西洋、アジア）からの任意の文字を保存および表示できます。

- 既存データベースを適用するには、データベースの形式を9.3形式のデータベースに移行する必要があります。次に実行するコマンド：
 - DBMSがSQL Serverの場合：
 - 1 Asset Manager Application Designerを起動します。
 - 2 データベースに接続します。
 - 3 [データベース/**UNICODE**テキストフィールドの使用] メニューを選択します。
 - 4 保存します（[ファイル/保存] メニュー）。
 - 新規データベースを作成するには：
 - 1 DBMSを介して空のシェルを作成する場合、使用しているDBMSにより要求されるアクションを実行します。
 - ▶ DBMSレベルでデータベースを作成する [P. 47].
 - 2 Asset Manager Application Designerでデータベースを作成する場合、[データベース] メニューで[UNICODEテキストフィールドの使用] オプションを有効にします。
 - ▶ Asset Manager Application Designerを使ってデータベース構造を作成する [P. 57]

 **注意:**

この機能により、データベースのサイズが2倍になるため、パフォーマンスを阻害する可能性があります。

 **注意:**

データベースがUnicodeテキストフィールドを保存する設定でない場合でも、データベースに文字を入力および保存できますが、正しく保存されません。その結果、レコードを表示する場合、テキストフィールドの情報は正しく表示されません。

 **ヒント:**

この機能は、日本語、韓国語、簡体字中国語バージョンもフルサイズのフィールドで提供します。[UNICODEテキストフィールドの使用] オプションが有効でない場合、フィールドの半分のみが保存されます。

すべてのユーザインタフェース項目をANSI コードページを使用しない言語（非西洋言語）で正しく表示するため、「WindowsクライアントでのANSIコードページを使用しない言語を表示するフォントの置換 [P. 65]」も参照してください。

Oracleを使用しているAsset Manager DBMSに対してUnicodeのサポートを追加する

Oracleに基づくAsset Managerデータベースに対してUnicodeのサポートを追加するには、次の手順に従います。

- 1 OracleツールのDatabase Configuration Assistantと次の値を使用して、新しいOracleインスタンスを作成します。
- 2 Oracleエクスポートツール (expdp) を使用して、現在のデータベースをエクスポートします。
- 3 Oracleインポートツール (impdp) を使用して、既存のAsset Managerデータベースを手順1で作成したUnicodeインスタンスにインポートします。データのインポートが正常に終了したことを確認します。

警告:

エクスポートとインポートには、数時間かかる場合があります。

エクスポートとインポートのプロセスはOracleツールを使用して実行されるため、プロセス中、特に変換が行われるインポート中に何らかのエラーが発生した場合、HP Softwareではツールのサポートを提供できません。

- 4 「C:\Documents and Settings\All Users\Application Data\HP\AssetManager\conf\amdb.ini」 ファイルを開きます。
- 5 データベースへの接続を記述するセクションに、次の行を追加します。

```
TableSpace=MY_DATA_TAB_SPC TableSpaceIndex=MY_INDEX_TAB_SPC  
TableSpaceLong= MY_DATA_TAB_SPC TableSpaceArchive=MY_ARCHIVE_TAB_SPC
```

ここで、

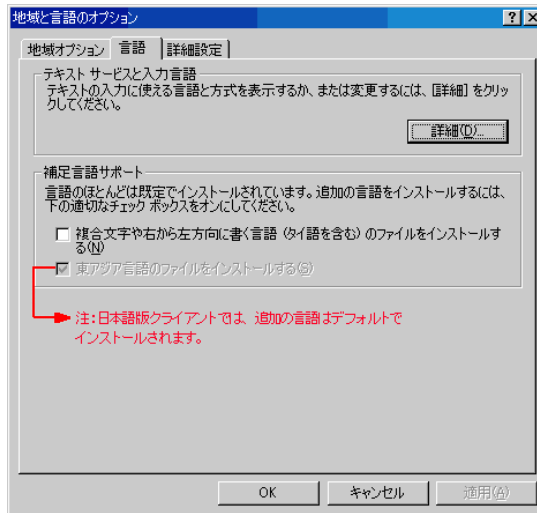
```
MY_DATA_TAB_SPC MY_INDEX_TAB_SPC MY_ARCHIVE_TAB_SPC
```

は、データ、インデックス、およびアーカイブがある場合に、それらに対して定義したテーブルスペースです。

警告:

この部分には、数時間かかる場合があります。

- 6 以下のスクリーンショットのように、Oracleツールを使用しているAsset Managerクライアントコンピュータの追加言語をインストールします。



- 7 Oracle SQL Plusに関する次のクエリを実行して、データベースがUnicodeをサポートするように正しく設定されていることを確認します。

1 `SELECT * FROM NLS_DATABASE_PARAMETERS;`

次の結果が取得されるはずですが。

```
NLS_CHARACTERSET AL32UTF8 NLS_NCHAR_CHARACTERSET
AL16UTF16
```

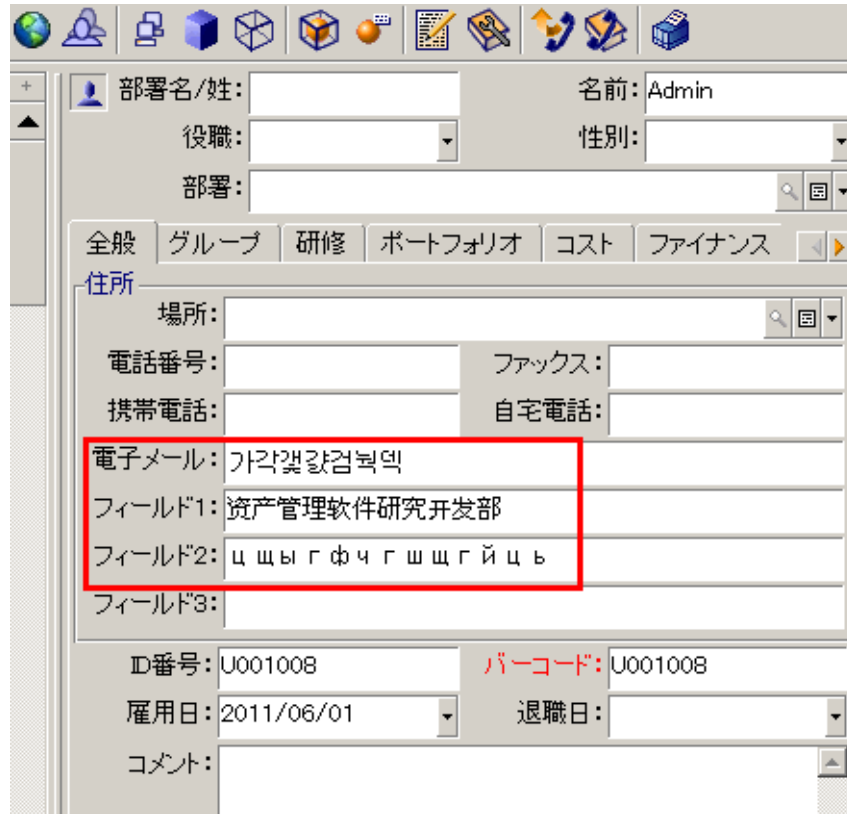
- 8 例えば、次のようにUnicode文字列を含むソースに対してスクリーンテキストダンプを実行します。

`SELECT DUMP (email,1017) FROM amEmplDept;`

これにUnicode文字列が含まれている場合、次のような結果が取得されます。

```
DUMP(EMAIL,1017) -----
----- NULL Typ=1 Len=17 CharacterSet=UTF8: 89,e4,b8,d0,d1,8b,d1,89,
e4,b8,d1,b8,d1,89,e4,b8,b9
```

Unicodeが正しくサポートされているとすると、次の例に示すように、UnicodeベースのテキストをAsset Managerに入力できるようになります。



The screenshot shows a user profile form in Asset Manager. The form includes fields for name, job, department, address, phone numbers, and email. The email field contains the Korean text "가각갯갯검늑덱". Below it, two address fields contain Chinese characters: "资产管理软件研究开发部" and "ц щы г фч г шщ г й ц ь". A red box highlights these two fields. The form also shows fields for ID number, barcode, employment date, and resignation date.

DBMSを変更する

Asset Managerでは、Asset Managerのデータベースを管理するのに使用しているDBMSを変更することができます。

DBMSを変更するには、旧DBMSのデータベースの構造と内容を出力し、新規DBMS内へインポートする必要があります。

警告:

データベースバージョンは同じでなければなりません。例えば、Asset Manager 3.xをAsset Manager 9.3へ出力することは不可能です。

次の手順に従います。

- 1 旧DBMSでデータベースの出力ファイルを作成します。
- 2 新DBMS用に空のシェルを作成します。
- 3 空のシェルへの接続を宣言します。
- 4 作成された空のシェル内で出力ファイルを復元します。

 **注意:**

手順2と3はこのセクションでは説明されていません。詳細についてはこの章の次のセクションを参照してください。

- DBMSを使って空のシェルを作成する [P. 46],
- Asset Managerへの接続を作成する [P. 56],
- Asset Manager Application Designerを使ってデータベース構造を作成する [P. 57].

データベース構造およびデータの出力

出力するには：

- 1 Asset Manager Application Designerを起動します。
- 2 [ファイル/開く.../既存のデータベースを開く] メニューでAsset Manager Application DesignerのAsset Managerデータベースを開きます。
- 3 [アクション/データベース構造とデータの出力...] メニューを選択します。
- 4 出力名を指定し、[保存] をクリックします。

Asset Manager Application Designerは出力ファイルを生成します。

 **注意:**

データベースの出力は、「xxx01.ar」、「xxx02.ar」、「xxx03.ar」などのように自動的に番号が付けられた複数のファイルから構成されます。ここでxxxは出力の名前です。

データベース構造とデータをインポートする

出力の実行、空のシェルの作成、接続の宣言がなされた後、出力されたファイルをインポートできるようになります。

- 1 [アクション/データベース構造とデータのインポート...] メニューを使用します。
- 2 表示されるダイアログボックス内で、上記で作成した空のシェルを選択します。
- 3 Asset Manager Application Designerが生成した最初の出力ファイル (xxx01.ar) を選択します。

Asset Manager Application Designerは新規データベース内に出力ファイルをインポートします。

SQL 2000データベースのSQL 2005 Serverへのリンク

SQL 2000システムをSQL 2005に移行するには、Asset Managerでデータベースを100%利用可能にするため、以下を行う必要があります。

- 1 旧DBMS (SQL Server 2000) のデータベース互換性レベルを変更します。データベースが"SQL Server 2000"として記されており、SQL 2005モードと互換していないため、この作業を行う必要があります。
 - 1 SQL Server 2005管理ツールを起動します。
 - 2 データベースプロパティを編集し、次に [互換性レベル] オプションの値を**SQL Server 2005**に設定します。
- 2 データベーストリガのSQLシンタックス変換：
 - 1 Asset Manager Application Designerを起動します。
 - 2 [アクション/データベースの診断/修復...] メニューを選択します。
 - 3 [トリガのチェック] タスクを選択し、[修復] オプションを選択して、[実行] をクリックします。

Asset Managerデータベースを削除する

セキュリティ上の理由から、Asset Managerにはデータベースを削除するコマンドがありません。Asset Manager外で削除を実行しなければなりません。

Asset Managerデータベースを削除する手順は、以下のとおりです。

- 1 データベース自体を削除します。

DBMSのデータベース管理ツールを使用します。

 - Microsoft SQL Server 7.0の場合は、**Microsoft SQL Enterprise Manager**を使用します。
 - Oracleの場合は、**User Manager Oracle**を使用します。

Microsoft SQL Serverの場合：

 - 1 削除するデータベースがインストールされているSQL ServerインスタンスのWindowsサービス (**MSSQL\$ASSETMANAGER**など) が起動されていることを確認します。
 - 2 DOSコマンドプロンプトを開きます。
 - 3 次のコマンドを実行します。
 - a SQL ServerインスタンスのSQLエディタを起動するコマンドを入力します。

シンタックス：

```
osql -S <サーバ名>\<SQL Server インスタンス名> -U <ユーザID> -P <ユーザパスワード>
```

例：

```
osql -S (local)\ASSETMANAGER -U sa -P saampassword
```

- b 次のコマンドを実行します。

```
go
```

- c データベースを切断するコマンドを入力します。

シンタックス：

```
exec sp_detach_db '<データベース名>', 'false'
```

例：

```
exec sp_detach_db 'ACBase', 'false'
```

- d 次のコマンドを実行します。

```
go
```

- e データベースに関連付けられた「.mdf」ファイルと「.log」ファイルを削除します。

- 2 各クライアントコンピュータで、Asset Managerレベルで定義された接続を削除します。
- 3 Microsoft SQL Serverの場合、各クライアントコンピュータでODBC接続を削除します。

5 データベース記述ファイルの作成

Asset Manager Application Designerでは、データベースの情報を抽出する際に、抽出する情報の属性とフォーマットを制御することができます。

Asset Managerデータベースの構造の記述を見るには、次のファイルまたはプログラムを参照します。

- 「database.txt」および「tables.txt」ファイル：データベースの完全な構造が記述されています。この2つのファイルは、Asset Managerのインストール先フォルダの「doc\infos」サブフォルダに格納されています。

注意:

上記のファイルでは、デフォルトのデータベース構造が記述されています。ユーザによるカスタマイズ情報は記述されていません。

これらの記述ファイルに、お使いのデータベースのカスタマイズ情報を反映させるには、Asset Manager Application Designerソフトウェアを使ってお使いのデータベースに接続する必要があります。

- Asset Manager Application Designerプログラム：Asset Managerデータベース（テーブル、フィールド、リンク、インデックス）の記述ファイルを次のようにして作成できます。以下を利用します。
 - Asset Managerのデータベース記述ファイル（「gbbase*.*」ファイル）を開くか、またはAsset Managerデータベースへ接続します。
 - 作成する情報を記述するテンプレート（拡張子が「.tpl」のファイル）を使います。Asset Managerプログラムに付属している標準テンプレートを基に、独自のテンプレートを作成することができます。また、RTFまたはHTML形式の記述ファイルを作成することもできます。

 **注意:**

Asset Managerに付属の標準テンプレートのうちの1つ、「Dbdict.tpl」ファイルを使うと、カスタマイズデータを使用中のデータベースから標準のテキストファイルにエクスポートできます。これらの情報（任意管理項目、特殊フィールド、設定スクリプトなど）には、任意管理項目、特殊フィールド、設定スクリプトなどがあります。「ソース管理」ツールでこの記述ファイルを使うと、データベースのカスタマイズ情報をトラッキングする際に便利です。

■ **Asset Managerプログラム**

この機能にアクセスするには【アクション／テンプレート】メニューを使用します。このメニューには以下のサブメニューがあります。

- **【フォルダの選択】**：Asset Manager Application Designerが記述テンプレートを検索するフォルダを指定できます。選択したフォルダのすべてのサブフォルダを検索します。
- **【リストの更新】**：前回指定したフォルダから記述ファイルの検索を再開します。
- **その他のサブメニュー**：Asset Manager Application Designerがフォルダ内で検出したすべての記述テンプレートが表示されます。ここで記述テンプレートの名前を選択すると、そのテンプレートを直接実行できます。

 **注意:**

データベース記述テンプレートの実行中に、Asset Manager Application Designerがテンプレート内に値の定義がない変数を検出すると、この変数の値を入力する画面が表示されます。

はじめに

データベースの内部構造を、オブジェクトの階層型コレクションとして表示することができます。階層構造では、データベースはテーブルを含み、テーブルはフィールド、リンク、インデックスなどを含みます。

データベースを記述するという事は、この構造を検索し、必要な情報を適切な形で抽出するという事です。Asset Manager Application Designerが情報を抽出する方法（つまり抽出する情報の内容とフォーマット）は、テンプレートと呼ばれるファイルに記述されています。これらのファイルは小さなプログラムであり、プログラムのシンタックスは、プログラミングの経験が少しでもあれば容易に理解できます。このシンタックスについてはこの章の後の方のセクションで説明します。

データベース記述パラメータ

次のパラメータを使ってデータベースを記述します。

Instance DATABASE Property P1-n Collection TABLES as TABLE Collection CALCFIELDS as CALCFIELDDESC Collection FEATURES as FEATPDESC Collection PARAMS as FEATPARAMDESC Collection CLASSES as FEATCLASSDESC Collection SCREENS as VIEW

Instance TABLE Property P1-n Collection FIELDS as FIELD Collection LINKS as LINK Collection INDEXES as INDEX Collection RELEVANTSCRIPT as SCRIPT Collection PROCESSES as BGPROC Collection FEATURES as FEATPARAMDESC Object O1-n as <インスタンス名>

Instance FIELD Property P1-n Collection DEFVALDEPENDENCIES as DEFVALSCRIPT Collection SYSENUMVALUES as SYSENUMVALUE Object O1-n as <インスタンス名>

Instance LINK Property P1-n Object O1-n as <インスタンス名>

Instance INDEX Property P1-n Collection FIELDSINDEX as FIELD Object O1-n as <インスタンス名>

Instance SCRIPT Property P1-n Collection REFERENCEDFIELD as SCRIPTFIELD Collection REFERENCEDSTORAGEFIELDS as STRING Object O1-n as <インスタンス名>

Instance FEATDESC Collection PARAMS as FEATPARAMDESC Object O1-n as <インスタンス名>

Instance FEATPARAMDESC Property P1-n Object O1-n as <インスタンス名>

Instance FEATCLASSDESC Property P1-n Object O1-n as <インスタンス名>

Instance CALCFIELDDESC Property P1-n Object O1-n as <インスタンス名>

Instance VIEW Property P1-n Collection PAGES as PAGE Collection FIELDSINLISTCONFIG as PAGEITEM Object O1-n as <インスタンス名>

Instance PAGE Property P1-n Collection FIELDS as PAGEITEM Object O1-n as <インスタンス名>

Instance PAGEITEM Property P1-n Object O1-n as <インスタンス名>
Instance SYSENUMVALUE Property P1-n Object O1-n as <インスタンス名>
>
Instance STRING Property P1-n Object O1-n as <インスタンス名>
Instance SCRIPTFIELD Property P1-n Object O1-n as <インスタンス名>
Global Values Property P1-n

Asset Managerデータベースの構造は、次のインスタンスで構成されています。

- Database：データベース自体
- Table：データベースのテーブル
- Field：テーブル内のフィールド
- Link：テーブル内のリンク
- Index：テーブル内のインデックス
- Script：フィールド値を計算するスクリプト

各インスタンスは、次の情報で記述されます。

- Property：インスタンスのプロパティ

例：

Instance Table Property SqlName

SqlNameプロパティは、テーブルのSQL名を表します。

- Collection：インスタンスのコンポーネントの1つを構成する項目の集まり（コレクション）

例：

Instance Index Collection FieldsIndex as Field

インデックス（Indexインスタンスのコンポーネントの1つ）は、主にフィールドのセット（FieldsIndexコレクション）で定義されます。各フィールドは、Fieldインスタンス内の項目です。

- Object：インスタンスの1つのコンポーネント

例：

Instance Link Object SrcField as Field

リンク（Linkインスタンスのコンポーネントの1つ）は、主にソースフィールド（SrcFieldオブジェクト）で定義されます。このフィールドは、Fieldインスタンスのコンポーネントです。

記述テンプレートのシンタックス

Asset Manager Application Designerで使うテンプレートには、抽出すべき情報と、その処理方法および表示方法が定義されています。

テンプレートファイルのフォーマットは次のとおりです。

- データ型：テキスト
- 文字セット：ANSI
- 拡張子：.tpl

シンタックスは次のとおりです。

- 固定テキスト
- コメント
- 別のテンプレートを挿入する
- コンポーネントの参照、並べ替え、および抽出
- `$if...$else...$elseif...$endif`条件
- 記述テンプレートで使用可能な関数
- テンプレートで定義した関数を使ってプロパティの値を処理する
- 改行マークを無視する
- 参照したコンポーネントを数える
- テンプレートのグローバル変数を定義する

固定テキスト

「\$」記号以外で始まり、関数の一部でない文字列は、Asset Manager Application Designerでそのまま出力されます。

 **注意:**

「\$」を出力する場合は、テンプレートに「\$\$」と入力する必要があります。

例：

テンプレートに次のように入力します。

```
List of tables.SQL NAME $$
```

次のよう出力されます。

```
List of tables.SQL NAME $
```

コメント

テンプレートのコメントとしてテキストを追加する場合は、行の最初に「\$」記号とスペース1つを挿入して、Asset Manager Application Designerが認識できないようにします。

例：

```
$コメント行
```

別のテンプレートを挿入する

外部のテンプレートを挿入するには、次のシンタックスを使います。

```
$include "<挿入するテンプレートの完全パス>"
```

例：

```
$include "e:\models\dbscript.tpl"
```

使用例：関数を含んでいるテンプレートを定義することで、そのテンプレートを参照する他のテンプレートもその関数を使うことができます。

コンポーネントの参照、並べ替え、および抽出

一般的なシンタックス

```
$for [<コレクション名> | *] [alias <エイリアス名>] [sort (<最初のプロパティ名> (ASC | DESC) [, <次のプロパティ名> (ASC | DESC)]] [<抽出条件>] ...$endfor
```

\$for...\$endforでコレクションのコンポーネントを参照する

コレクションのコンポーネントを繰り返し参照する場合は、次のシンタックスを使います。

```
$for <コレクション名> ...$for <サブコレクション> ...$endfor $endfor
```

例：

```
$for Tables ...$for Fields ...$endfor $endfor
```

コレクション間の階層に注意する必要があります。例：

- 1 Fieldsコレクションは、Tablesコレクションによって変わります。
 - 2 FieldsIndexコレクションは、Indexesコレクションによって変わります。
- <コレクション名>を指定する代わりにアスタリスク (*) 記号を使うことができます。この記号を使うと、現在のインスタンスの全コレクションを呼び出します。例：

```
$for Tables ...$for * $(SqlName) ...$endfor $endfor
```

この例では、Tableインスタンスの全コレクションのSQL名 (Fields、Links、Indexes) を取得できます。

sortで最終結果を並べ替える

1つのコレクションのコンポーネントを並べ替えるには、次のシンタックスを使います。

```
$for <コレクション> sort (<最初のプロパティ名> (ASC | DESC) [, <次のプロパティ名> (ASC | DESC)]) ...$endfor
```

次のキーワードで並び替え順序を指定できます。

- 1 ASC : 昇順
- 2 DESC : 降順

例 :

```
$for Tables sort (SqlName ASC) ...$for Fields sort (UserType DESC, UserType Format ASC, SqlName ASC) ...$endfor $endfor
```

コレクションの項目またはオブジェクトのプロパティを取得する

コレクションの項目またはオブジェクトのプロパティを取得するには、次のシンタックスを使います。

```
$for <コレクション名> ...${[<コレクションの名前またはエイリアス>].[<オブジェクト名>].<プロパティ> ...$endfor
```

 **注意:**

コレクション内の\$for... \$endforループでプロパティを呼び出す場合は、<コレクションの名前またはエイリアス>を指定する必要はありません。

例 :

```
$for Tables $for Fields $(Tables.SqlName) $(SqlName) $endfor  
$for Links $(Tables.SqlName) $(SqlName) $(Reverselink.SqlName) $endfor  
$endfor
```

aliasでエイリアスを割り当てる

現在のところ、エイリアスの特別な用法はありません。

filterでコレクションのコンポーネントを抽出する

コレクションのコンポーネントを抽出するには、次のシンタックスを使います。

```
$for <コレクション名> filter <抽出条件> ...$endfor
```

抽出条件はBasicで記述します。

例：

```
$for tables filter $Left($SqlName, 1) = "p" ...$endfor
```

上の例では、SqlNameが文字「p」で始まるテーブルのみが表示されます。

\$if...\$else...\$elseif...\$endif条件

目的のコンポーネントを含んでいるプロパティの条件範囲を定義できます。

シンタックス：

```
$if <テストの条件> ...$elseif <テストの条件> ...$else <テストの条件> ...$endif
```

テストの条件は、Basic式、\$script...\$endscriptフォーマットで定義する関数、インスタンスのプロパティで定義できます。

例：

```
$for Links $if $(typed) = 0 $(Tables.SqlName) $(SqlName) $(SrcField.SqlName) $(DstTable.SqlName) $else $(Tables.SqlName) $(SqlName) $(SrcField.SqlName) $endif $endfor
```

記述テンプレートで使用可能な関数

Asset Manager Application Designerには、テンプレートで使用できるいくつかの関数が事前に定義されています。

ValueOf(<strProperty> as String) as String

Property プロパティの値を呼び出すもう1つのシンタックスです。

Propertyは必ず大文字で指定します。

例：

```
$ValueOf("PRIMARYKEY")
```

次のシンタックスと同じ結果が得られます。

```
$(PrimaryKey)
```

SetProperty(<strProperty> as String, <strValue> as String, <iValueType> as Integer) as String

テンプレート用に、**Property**という名前で**ValueType**タイプのグローバル変数を作成します。

Propertyは必ず大文字で指定します。

例：

```
I = SetProperty("NEWPROPERTY", "2", VarType(2))
```

テンプレート用に、数値**2**をとる**NEWPROPERTY**というグローバル変数を作成します。この変数が正しく作成された場合は戻りコード**I = 0**を返します。

```
I = SetProperty("NEWPROPERTY", "Test", VarType("Test"))
```

テンプレート用に、テキスト値**Test**をとる**NEWPROPERTY**というグローバル変数を作成します。この変数が正しく作成された場合は戻りコード**I = 0**を返します。

Exist(<strProperty> as String) as Integer

グローバル変数**Property**の有無をテストします。

例：

```
Exist("NEWPROPERTY")
```

プロパティが存在する場合は数値**1**を、プロパティがない場合は**0**を返します。

LogError(<strErrorCode> as String, <strMessage> as String) as String

ErrorCodeと表示するエラーメッセージ**Message**を定義します。

例：

```
LogError(1, "property not found!")
```

ASCIIのエラーメッセージを定義したフォーマット（大文字と小文字）で出力します。

SetOutput(<strFile> as String) as String

結果を出力するファイルを定義します。出力ファイルには、コマンド行で定義した優先順位があります。

例：

```
SetOutput("e:\exportdb\sortie.txt")
```

結果を「**e:\exportdb\output.txt**」というファイルに保存します。

```
SetOutput("")
```

結果を画面に表示します。

CollectionCreate(<strName> as String) as Integer

データベース項目の新しいコレクションを宣言します。作成したコレクションの名前は、**Fields**または**Tables**などの有効なデータベースコレクションでなければなりません。この関数とその後記述する関数は、通常、コレクションのコンポーネントを走査するのに使います。**\$For...\$Next**シンタックスの代わりにこれらの関数を使うことができます。

例：

```
CollectionNext() as IntegerCollectionCreate("Fields")
```

この関数は、コレクションが作成された場合に0を返します。その他の場合は、エラーコードを返します。

CollectionNext() as Integer

CollectionCreate()を使って事前に定義したコレクションの繰り返しを実行します。

例：

```
CollectionNext()
```

この関数は、繰り返しが正常に実行された場合に0を返します。その他の場合は、エラーコードを返します。コレクションの最後の要素に達した場合にもエラーを返します。

CollectionName() as String

CollectionCreate()関数を使って事前に宣言したコレクションの名前を返します。

例：

```
strName=CollectionName()
```

CollectionIsFirst() as Integer

プログラムがアクセスしているコレクションの要素がコレクションの先頭にあるかどうかをテストします。

例：

```
CollectionIsFirst()
```

この関数は、要素がコレクションの先頭にある場合に1を、それ以外の場合はずべて0を返します。

CollectionIsLast() as Integer

プログラムがアクセスしているコレクションの要素がコレクションの末尾にあるかどうかをテストします。

例：

```
CollectionIsLast()
```

この関数は、要素がコレクションの末尾にある場合に1を、それ以外の場合はずべて0を返します。

CollectionCurrentIndex() as Integer

プログラムがアクセスしているコレクションの要素のインデックス番号を返します。コレクションは、CollectionCreate()関数を使って事前に宣言しておく必要があります。

例：

```
Number=CollectionCurrentIndex()
```

CollectionCount() as Integer

現在のコレクションに属する要素の数を返します。コレクションは、CollectionCreate()関数を使って事前に宣言しておく必要があります。

例：

```
iCollec=CollectionCount()
```

テンプレートで定義した関数を使ってプロパティの値を処理する

<function>で関数を使う

関数を定義し、その関数を使ってプロパティの値を処理できます。関数を使うシンタックスは次のとおりです。

```
$(関数)<$(<プロパティ1>,...,<プロパティn>))
```

例：

```
$StrType$(Type)
```

```
$Duplicates$(Duplicates), $(NullValues)
```

\$script...\$endscriptで関数を定義する

関数は、次のように\$scriptと\$endscriptの2つのマーカーで区切られたBasicブロック内に定義します。

```
$script ...Function ...End Function ...$endscript
```

関数のシンタックスは次のとおりです。

```
Function <関数の名前>({ByVal | ByRef} [<入力する変数の名前> as <データの  
入力フォーマット>]*) as <出力フォーマット> ...End Function
```

関数は、Basicの式とインスタンスのプロパティで表すことができます。

注意:

デフォルトでは、関数の宣言で使用する変数は確認されません。特に、宣言することなく変数を使用できるため、スクリプトの実行時にエラーが発生する可能性があります。このデフォルト設定を変更するためには、関数のBasicブロックの最初に以下のラインを追加します（\$scriptマーカーのすぐ後にこのラインを追加します）。

```
Option Explicit
```

例：

```
Function ReturnYesNo(ByVal iValue as Integer) as String if iValue = 1 then  
ReturnYesNo = "Yes" else ReturnYesNo = "No" end if End Function
```

```
Function StrType(ByVal iValue as Integer) as String select case iValue case 1  
: StrType = "Integer (8 bit)" case 2: StrType = "Integer (16 bit)" case 3: StrType  
= "Integer (32 bit)" case 4: StrType = "Floating point number" case 5: StrType  
= "Double-precision number" case 6: StrType = "Text" case 7: StrType = "D  
ate and time" case 9: StrType = "Binary field" case 10: StrType = "Date" case  
11: StrType = "Time" case 12: StrType = "Long text field" case else Dim strEr  
ror as String strError = "Type " + CStr(iValue) + " undefined" strType = LogE  
rror(1, strError) End select End Function
```

改行マークを無視する

行の途中に情報を挿入しなければならない場合がありますが、情報を生成する関数は必ず行の先頭から始まります。

このような場合は、関数の前の行の末尾に次の文字列

```
$nocr
```

を追加して、改行マークを無視するようにします。

例：

```
... $for Indexes $(Tables.Sqlname) $(Sqlname) $nocr for FieldsIndex $if $(Isla  
st) = 1 $(Sqlname) $else $(Sqlname)$nocr $nocr $endif $endfor ...
```

次のように入出力されます。

```
... amProduct Prod_BrandModel Brand, Model amProduct Prod_CatalogRef C  
atalogRef amProduct Prod_lCategIdBrand lCategId, Brand, Model
```

参照したコンポーネントを数える

コレクション内で抽出条件によって参照されたコンポーネント数を数えるには、次のシンタックスを使います。

```
$for <コレクション名> filter <抽出条件> $(count) ...$endfor
```

テンプレートのグローバル変数を定義する

グローバル変数を定義するには、次のシンタックスを使います。

```
$<変数名> = <Basic式>
```


例：

```
$A = 1
```

```
$Var = "text"
```

```
$A = $(A) + 1
```

```
$Form = Left($(Var), 2)
```

特定のデータベース記述パラメータについて

このセクションでは、次の記述パラメータについて説明します。

- Databaseインスタンス
- Tableインスタンス
- Fieldインスタンス
- Linkインスタンス
- Indexインスタンス
- Scriptインスタンス
- FeatDescインスタンス
- FeatParamDescインスタンス
- FeatClassDescインスタンス
- CalcFieldDescインスタンス
- Viewインスタンス
- Pageインスタンス
- PageItemインスタンス
- SysEnumValueインスタンス
- Stringインスタンス
- ScriptFieldインスタンス
- グローバル変数

Databaseインスタンス

プロパティ

表 5.1. Databaseインスタンスのプロパティ

プロパティ名	説明	接続が必要
LoginName	データベースにアクセスするためのログイン名	はい
LoginId	データベースにアクセスするためのログインID	はい
TableCount	データベース内のテーブルの総数	いいえ
Connected	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none">1: Asset Manager Application Designerで接続を開いた。0: Asset Manager Application Designerでデータベース記述ファイルを開いた。	いいえ
接続	データベースにアクセスするときに使うAsset Managerの接続名	はい
AppInfo	Asset Managerに関する情報	いいえ
AppVersion	Asset Managerのバージョン番号	いいえ
AppBuild	Asset Managerのビルド番号	いいえ
AppLanguage	Asset Managerの言語	いいえ
DbbVersion	データベースの構造のバージョン番号	いいえ

Tableインスタンス

プロパティ

表 5.2. Tableインスタンスのプロパティ

プロパティ名	説明	接続が必要
Create	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none">1: ログインにテーブルでの作成権限がある。0: ログインにテーブルでの作成権限がない。	はい
Delete	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none">1: ログインにテーブルでの削除権限がある。0: ログインにテーブルでの削除権限がない。	はい
ComputeString	テーブルの記述文字列	いいえ
InternalName	内部名	いいえ
Label	ラベル	いいえ
Desc	説明	いいえ
SqlName	SQL名	いいえ

プロパティ名	説明	接続が必要
FieldCount	テーブル内のフィールドの総数	いいえ
LinkCount	テーブル内のリンクの総数	いいえ
IndexCount	テーブル内のインデックスの総数	いいえ
IsFirst	抽出条件と並べ替え順序を指定したときに、項目がコレクションの先頭かどうかを次のように表します。 <ul style="list-style-type: none"> ■ 0: いいえ ■ 1: はい 	いいえ
IsLast	抽出条件と並べ替え順序を指定したときに、項目がコレクションの末尾かどうかを次のように表します。 <ul style="list-style-type: none"> ■ 0: いいえ ■ 1: はい 	いいえ
Count	抽出条件を指定したときに、コレクション内で参照された項目数	いいえ
CurrentIndex	抽出条件と並べ替え順序を指定したときの、コレクション内の項目の位置	いいえ
System	オブジェクトがシステムオブジェクト（変更不可能）であるかどうかを示します。 <ul style="list-style-type: none"> ■ 0: いいえ ■ 1: はい 	いいえ
HasFeatureValueTable	テーブルに、関連付けられた任意管理項目のテーブルが含まれているかどうかを指定します。 <ul style="list-style-type: none"> ■ 0: いいえ ■ 1: はい 	いいえ
IsFeatureValueTable	テーブルが、任意管理項目のテーブルであるかどうかを指定します <ul style="list-style-type: none"> ■ 0: いいえ ■ 1: はい 	いいえ
HasMemoField	テーブルにメモ型フィールドがあるかどうかを示します。 <ul style="list-style-type: none"> ■ 0: いいえ ■ 1: はい 	いいえ

オブジェクト

表 5.3. Tableインスタンスのオブジェクト

オブジェクト名	説明
MainIndex as Index	メインインデックス
PrimaryKey as Field	主キー
FeatureValueTable as Table	任意管理項目の値を保存するテーブル
FVSourceTable as Table	ソース任意管理項目の値のテーブル
IsValidScript as Script	有効性のスクリプト
RelevantScript as Script	妥当性のスクリプト

オブジェクト名	説明
Base as Database	記述されるデータベース

Fieldインスタンス

プロパティ

表 5.4. Fieldインスタンスのプロパティ

プロパティ名	説明	接続が必要
Update	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> ■ 1: ログインにフィールドの更新権限がある。 ■ 0: ログインにフィールドの更新権限がない。 	はい
Write	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> ■ 1: ログインにフィールドの作成権限がある。 ■ 0: ログインにフィールドの作成権限がない。 	はい
Read	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> ■ 1: ログインにフィールドの読取り権限がある。 ■ 0: ログインにフィールドの読取り権限がない。 	はい
UserType	デフォルトでは、データの入力と表示フォーマットは「Type」プロパティのデータ型になります。 検証時に、「UserType」プロパティでデータの入力と表示フォーマットを指定できます。	いいえ
Type	保存値のデータ型	いいえ
UserTypeFormat	UserTypeパラメータの補足情報	いいえ
Size	フィールド値の最大サイズ (半角の場合の文字数)	いいえ

プロパティ名	説明	接続が必要
ReadOnly	フィールドを変更できるかどうかを定義します。実際に変更できるかどうかはデータベースに接続するユーザのアクセス権によります。このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> 1: 変更不可 0: ユーザに必要な権限がある場合は変更可能 	いいえ
Historized	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> 1: フィールドの履歴を記録する。 0: フィールドの履歴を記録しない。 	いいえ
ForeignKey	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> 1: フィールドは外部キーである。 0: フィールドは外部キーではない。 	いいえ
PrimaryKey	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> 1: このフィールドは主キーである。 0: このフィールドは主キーではない。 	いいえ
InternalName	内部名 この情報に特に用途はありません。	いいえ
Label	フィールドのラベル（詳細画面に表示されるフィールド名など）。	いいえ
Desc	説明	いいえ
SqlName	SQL名	いいえ
LongHelpComment	フィールド値についてのコメント	いいえ
LongHelpSample	フィールドに割り当てられる値の例	いいえ
LongHelpWarning	フィールドに関する重要な情報	いいえ
LongHelpDesc	フィールドの説明	いいえ
LongHelpCommentNoHTMLTag	フィールド使用についてのコメント（HTMLタグは除外されています）	いいえ
LongHelpSampleNoHTMLTag	フィールドに割り当てる値の例（HTMLタグは除外されています）	いいえ
LongHelpWarningNoHTMLTag	フィールドに関する重要な情報（HTMLタグは除外されています）	いいえ
LongHelpDescNoHTMLTag	フィールドの説明（HTMLタグは除外されています）	いいえ
System	オブジェクトがシステムオブジェクト（変更不可能）であるかどうかを示します。 <ul style="list-style-type: none"> 0: いいえ 1: はい 	いいえ

プロパティ名	説明	接続が必要
EmptyOnDup	複製時にデフォルト値が再適用されるかどうかを指定します。 <ul style="list-style-type: none"> 0: いいえ 1: はい 	いいえ
FieldCase	フィールドでの大文字小文字に関する操作を指定します。 <ul style="list-style-type: none"> 0: 入力値を変更しない 1: 入力値は大文字に変更される 2: 入力値は小文字に変更される 3: 入力値はsmartcase (大文字が含まれる場合は大文字小文字を区別) になる 	いいえ
Positive	数値タイプのフィールドが正の数かどうかを指定する。 <ul style="list-style-type: none"> 0: いいえ 1: はい 	いいえ

オブジェクト

表 5.5. Fieldインスタンスのオブジェクト

オブジェクト名	説明
Base as Database	記述されるデータベース
Table as Table	フィールドを含んでいるテーブル
MandatoryScript as Script	フィールド値を計算する必須スクリプト
DefaultScript as Script	フィールド値を計算するデフォルトのスクリプト
ReadOnlyScript as Script	フィールド用の読取専用の計算スクリプト
HistoryScript as Script	フィールド値の履歴を計算するスクリプト
RelevantScript as Script	フィールドの妥当性を計算するスクリプト

Typeプロパティの値

表 5.6. Typeプロパティの値

保存値	表示値	説明
1	Integer (8 bit)	-128から+127の整数
2	Integer (16 bit)	-32,768から+32,767の整数
3	Integer (32 bit)	-2,147,483,647から+2,147,483,646の整数
4	Floating point number	4バイト浮動小数点数
5	Double-precision number	8バイト浮動小数点数
6	Text	あらゆる文字を含むテキスト
7	Date and time	日付+時刻

保存値	表示値	説明
9	Binary field	画像またはその他のバイナリファイルの保存に使用します。サイズの制限はありません。
10	Date	日付のみ（時刻なし）
11	Time	時刻だけ（日付なし）
12	Long text field	可変長のテキストフィールド

UserTypeプロパティの値

表 5.7. UserTypeプロパティの値

保存値	表示値
0	Default
1	Number
2	Boolean (Yes/No)
3	Monetary
4	Date
5	Date and time
6	Time
7	System itemized list
8	Custom itemized list
9	Feature value
10	Percentage
11	Duration
12	Table or field SQL name
13	Basic script

UserTypeFormatプロパティの値

このプロパティの内容は、UserTypeプロパティ（リストデータ、期間など）に応じて以下ようになります。

- 「リストデータ」：フィールドにリンクするリストデータの名前
- 「システムリストデータ」：リストデータのエントリ項目
- 「期間」：表示形式
- 「フィールド名またはテーブル名」：プロパティは、テーブルのSQL名を保存するフィールドのSQL名を含みます。このテーブルは、記述されたフィールドが指定するフィールドを含みます。

Linkインスタンス

プロパティ

表 5.8. Linkインスタンスのプロパティ

プロパティ名	説明	接続が必要
Update	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none">■ 1: ログインにリンクの更新権限がある。■ 0: ログインにリンクの更新権限がない。	はい
Write	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none">■ 1: ログインにリンクの作成権限がある。■ 0: ログインにリンクの作成権限がない。	はい
Read	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none">■ 1: ログインにリンクの読取り権限がある。■ 0: ログインにリンクの読取り権限がない。	はい
Type	リンクのタイプ	いいえ
UserType	リンクが管理する情報のタイプ	いいえ
Typed	リンクのターゲットテーブルを事前に定義するかどうかを表します。定義しない場合は、テーブルのSQL名がレコードのいずれかのフィールドに保存されます。 <ul style="list-style-type: none">■ 1: ターゲットテーブルを事前に定義しない。■ 0: ターゲットテーブルを事前に定義しない。	いいえ
Historized	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none">■ 1: リンクの履歴を記録する。■ 0: リンクの履歴を記録しない。	いいえ
Cardinality	リンク数	いいえ
InternalName	内部名 この情報に特に用途はありません。	いいえ
Label	ラベル	いいえ
Desc	説明	いいえ
SqlName	SQL名	いいえ
System	オブジェクトがシステムオブジェクト（変更不可能）であるかどうかを示します。 <ul style="list-style-type: none">■ 0: いいえ■ 1: はい	いいえ

プロパティ名	説明	接続が必要
LongHelpComment	リンク使用についてのコメント	いいえ
LongHelpSample	リンクに割り当てられる値の例	いいえ
LongHelpWarning	リンクに関する重要な情報	いいえ
LongHelpDesc	リンクの説明	いいえ
LongHelpCommentNoHTMLTag	リンク使用についてのコメント (HTMLタグは除外されています)	いいえ
LongHelpSampleNoHTMLTag	リンクに割り当てる値の例 (HTMLタグは除外されています)	いいえ
LongHelpWarningNoHTMLTag	リンクに関する重要な情報 (HTMLタグは除外されています)	いいえ
LongHelpDescNoHTMLTag	リンクの説明 (HTMLタグは除外されています)	いいえ

オブジェクト

表 5.9. Linkインスタンスのオブジェクト

オブジェクト名	説明
Base as Database	記述されるデータベース
SrcField as Field	ソースフィールド
SrcTable as Table	ソーステーブル
DstTable as Table	ターゲットテーブル
DstField as Field	ターゲットフィールド
RelTable as Table	リレーショナルテーブル
RelSrcField as Field	リレーショナルテーブルのソースフィールド
RelDstField as Field	リレーショナルテーブルのターゲットフィールド
TypeField as Field	リンクのターゲットテーブルが事前に定義されていない場合、このプロパティはターゲットテーブルのSQL名を含んでいるフィールドを表します。
ReverseLink as Link	逆リンク
HistoryScript as Script	リンク値の履歴を計算するスクリプト
RelevantScript as Script	リンクの妥当性を計算するスクリプト

Typeプロパティの値

表 5.10. Typeプロパティの値

保存値	表示値
1	Normal
2	Own
4	Define
8	Neutral

保存値	表示値
16	Copy
18	Owncopy

UserTypeプロパティの値

表 5.11. UserTypeプロパティの値

保存値	表示値	Asset Manager Automated Process Managerに表示される値
0	Normal	普通
1	Comment	コメント
2	Image	画像
3	History	履歴
4	Feature values	任意管理項目の値

Indexインスタンス

プロパティ

表 5.12. Indexインスタンスのプロパティ

プロパティ名	説明
Duplicates	このインデックスでNULLでない同じ値をもつレコードを複数作成できるかどうかを示します。 <ul style="list-style-type: none"> ■ 1：このインデックスでNULLでない同じ値をもつレコードを複数作成できる。 ■ 0：このインデックスでNULLでない同じ値をもつレコードを複数作成できない。
NullValues	このプロパティは、Duplicatesプロパティが「いいえ」に設定されている場合にのみ有効です。 このインデックスでNULL値をもつレコードを複数作成できるかどうかを示します（インデックスを構成するフィールドの値がすべてNULLの場合は、インデックスの値がNULLになります）。 <ul style="list-style-type: none"> ■ 1：NULLのインデックスを持つレコードを1つしか作成できます。 ■ 0：NULLのインデックスを持つレコードを1つしか作成できません。
InternalName	内部名 この情報に特に用途はありません。
Label	ラベル（詳細画面に表示されるものと同じ）
Desc	説明
SqlName	SQL名

プロパティ名	説明
System	オブジェクトがシステムオブジェクト（変更不可能）であるかどうかを示します。 <ul style="list-style-type: none"> ■ 0：いいえ ■ 1：はい

オブジェクト

表 5.13. Indexインスタンスのオブジェクト

オブジェクト名	説明
Base as Database	記述されるデータベース
Table as Table	インデックスを含むテーブル

Scriptインスタンス

プロパティ

表 5.14. Scriptインスタンスのプロパティ

プロパティ名	説明
CalcMode	フィールドの値を「はい」にするか「いいえ」にするか、または2つのうちのどちらにするかをスクリプトで計算するかを示します。このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> ■ 0：いいえ ■ 1：はい ■ 2：スクリプト
ScriptType	スクリプトで管理する情報のタイプ。このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> ■ 1：フィールドへのデータ入力が必要 ■ 2：デフォルトでテーブル内の任意管理項目を表示 ■ 3：テーブル内の任意管理項目で使用可能な文字 ■ 4：フィールドの履歴を記録 ■ 5：フィールドのデフォルト値 ■ 6: テーブル内の任意管理項目の [継承目的] (SQL名: bForInheritance) フィールド

プロパティ名	説明
Source	インタフェースで表示するフィールド値を計算するスクリプト データベースに保存するフィールド値を計算するスクリプト
VbReturnType	スクリプトで計算するデータ型 <ul style="list-style-type: none"> ■ 整数：-32,768から+32,767の整数 ■ 倍長整数：-2,147,483,647から+2,147,483,646の整数 ■ 倍精度：8バイトの浮動小数点数 ■ 文字列：あらゆる文字を含むテキスト ■ 日付：日付（時刻なし）
ReferencedStorageFieldCount	スクリプト内で参照されるフィールドの数

オブジェクト

表 5.15. Scriptインスタンスのオブジェクト

オブジェクト名	説明
Table as Table	スクリプトで値を計算するフィールドが属するテーブル
Field as Field	スクリプトで計算した値が入るフィールド

FeatDescインスタンス



注意:

このインスタンスを使用するにはデータベースへ接続する必要があります。

プロパティ

表 5.16. FeatDescインスタンスのプロパティ

プロパティ名	説明
System	オブジェクトがシステムオブジェクト（変更不可能）であるかどうかを示します。 <ul style="list-style-type: none"> ■ 0：いいえ ■ 1：はい
Label	任意管理項目のラベル
Desc	説明
SQLName	SQL名

プロパティ名	説明
Unit	任意管理項目の単位
Type	任意管理項目の入力タイプ 任意管理項目に値を入力する際に使用するリストデータ。このプロパティは、任意管理項目の Type がリストデータの場合のみ有効です。
MinValue	任意管理項目の最小値 (Type が数値の場合のみ)
MaxValue	任意管理項目の最大値 (Type が数値の場合のみ)
IsConsolidated	任意管理項目が統合により計算されています。 <ul style="list-style-type: none"> ■ 0 : いいえ ■ 1 : はい
HasClass	任意管理項目が任意管理項目のクラスに属するかどうかを指定します <ul style="list-style-type: none"> ■ 0 : いいえ ■ 1 : はい

オブジェクト

表 5.17. FeatDescインスタンスのオブジェクト

オブジェクト名	説明
Class as FeatClassDesc	任意管理項目のクラス

FeatParamDescインスタンス



注意:

このインスタンスを使用するにはデータベースへ接続する必要があります。

プロパティ

表 5.18. FeatParamDescインスタンスのプロパティ

プロパティ名	説明
LongHelpComment	フィールド値についてのコメント
LongHelpSample	フィールドに割り当てられる値の例
LongHelpWarning	フィールドに関する重要な情報
LongHelpDesc	フィールドの説明
LongHelpCommentNoHTMLTag	フィールド使用についてのコメント (HTMLタグは除外されています)

プロパティ名	説明
LongHelpSampleNoHTMLTag	フィールドに割り当てる値の例 (HTMLタグは除外されています)
LongHelpWarningNoHTMLTag	フィールドに関する重要な情報 (HTMLタグは除外されています)
LongHelpDescNoHTMLTag	フィールドの説明 (HTMLタグは除外されています)
LinkFilter	フィルタのAQL条件 (リンクタイプの任意管理項目)
IsInherited	任意管理項目が継承されるかどうかを表します。 <ul style="list-style-type: none"> ■ 0: いいえ ■ 1: はい
CreationHistorized	主レコードの作成時に履歴項目が作成されます。
Write	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> ■ 0: ログインには、任意管理項目への書き込み権限がある。 ■ 1: ログインには、任意管理項目への書き込み権限がある。
Read	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> ■ 0: ログインには、任意管理項目の読取り権限がある。 ■ 1: ログインには、任意管理項目の読取り権限がある。
Update	このプロパティは、次のいずれかの値をとります。 <ul style="list-style-type: none"> ■ 0: ログインには、任意管理項目の更新権限がある。 ■ 1: ログインに任意管理項目の更新権限がある。

オブジェクト

表 5.19. FeatParamDescインスタンスのオブジェクト

オブジェクト名	説明
DefaultScript as Script	任意管理項目値のデフォルトのスクリプト
MandatoryScript as Script	任意管理項目入力の必須スクリプト
AvailableScript as Script	任意管理項目の使用可能なスクリプト
HistoryScript as Script	任意管理項目の履歴スクリプト
ForceDisplayScript as Script	任意管理項目のデフォルトの表示スクリプト
Table as Table	任意管理項目パラメータの対象となるテーブル

オブジェクト名	説明
Feature as FeatDesc	任意管理項目パラメータに関連付けられる任意管理項目
ValueField as Field	任意管理項目の格納先フィールド (fVal、ValStringまたはdtVal)

FeatClassDescインスタンス



注意:

このインスタンスを使用するにはデータベースへ接続する必要があります。

プロパティ

表 5.20. FeatClassDescインスタンスのプロパティ

プロパティ名	説明
FullName	任意管理項目クラスの完全名
Name	任意管理項目のクラス名
HasParent	任意管理項目クラスに親クラスがあるかどうかを指定します <ul style="list-style-type: none"> ■ 0: いいえ ■ 1: はい

オブジェクト

表 5.21. FeatParamDescインスタンスのオブジェクト

オブジェクト名	説明
ParentClass as FeatClassDesc	任意管理項目の親クラス

CalcFieldDescインスタンス



注意:

このインスタンスを使用するにはデータベースへ接続する必要があります。

プロパティ

表 5.22. CalcFieldDescインスタンスのプロパティ

プロパティ名	説明
Label	特殊フィールドのラベル
Desc	説明
SQLName	SQL名 特殊フィールドの計算式
UserType	特殊フィールドが返す結果のタイプ
Type	特殊フィールドのタイプ
IsVisible	特殊フィールドがデフォルトで表示されるかどうかを指定する。 <ul style="list-style-type: none">0: いいえ1: はい

オブジェクト

表 5.23. CalcFieldDescインスタンスのオブジェクト

オブジェクト名	説明
Table as Table	特殊フィールドが関連付けられているテーブル
Script as Script	特殊フィールドの計算スクリプト

Viewインスタンス

プロパティ

表 5.24. Viewインスタンスのプロパティ

プロパティ名	説明
InternalName	内部名 この情報に特に用途はありません。
System	オブジェクトがシステムオブジェクト（変更不可能）であるかどうかを示します。 <ul style="list-style-type: none">0: いいえ1: はい
Label	画面のラベル
Desc	説明
SQLName	SQL名
FuncDomain	画面に関連付けられている機能ドメイン

プロパティ名	説明
HasSystemPage	画面に常に表示される情報（タブの上の見出し）があるかを指定します。 <ul style="list-style-type: none"> ■ 0：いいえ ■ 1：はい
HasNotebook	画面にタブがあるかどうかを指定します。 <ul style="list-style-type: none"> ■ 0：いいえ ■ 1：はい
ScreenFilter	画面のシステムフィルタ
CaptionList	画面リストのタイトル
CaptionDetail	画面のタイトル

オブジェクト

表 5.25. Viewインスタンスのオブジェクト

オブジェクト名	説明
Table as Table	画面を含むテーブル
SysPage as Page	画面のタブの上の見出し

Pageインスタンス

プロパティ

表 5.26. Pageインスタンスのプロパティ

プロパティ名	説明
InternalName	内部名
System	この情報に特に用途はありません。 オブジェクトがシステムオブジェクト（変更不可能）であるかどうかを示します。 <ul style="list-style-type: none"> ■ 0：いいえ ■ 1：はい
Label	ページのラベル
SQLName	説明 SQL名

PageItemインスタンス

プロパティ

表 5.27. PageItemインスタンスのプロパティ

プロパティ名	説明
IsVerticalSplit	このオブジェクトがページを区切る縦線であるかどうかを示します。 <ul style="list-style-type: none">■ 0: いいえ■ 1: はい
IsHorizontalSplit	このオブジェクトがページを区切る横線であるかどうかを示します。 <ul style="list-style-type: none">■ 0: いいえ■ 1: はい
FrameLabel	フレームのラベル
SQLName	SQL名
IsField	オブジェクトがページのフィールドであるかどうかを示します。 <ul style="list-style-type: none">■ 0: いいえ■ 1: はい
IsLink	オブジェクトがリンクであるかどうかを示します。 <ul style="list-style-type: none">■ 0: いいえ■ 1: はい
IsFeature	オブジェクトが任意管理項目であるかどうかを示します。 <ul style="list-style-type: none">■ 0: いいえ■ 1: はい
IsCalcField	オブジェクトが特殊フィールドであるかどうかを示します。 <ul style="list-style-type: none">■ 0: いいえ■ 1: はい
BaseType	オブジェクトがフィールドの場合、フィールドのデータ型を表します。

オブジェクト

表 5.28. PageItemインスタンスのオブジェクト

オブジェクト名	説明
Field As Field	ページに含まれるフィールド

オブジェクト名	説明
Link as Link	ページに含まれるリンク
FeatParam as FeatParamDesc	ページに含まれる任意管理項目
CalcField as CalcFieldDesc	ページに含まれる特殊フィールド

SysEnumValueインスタンス

プロパティ

表 5.29. SysEnumValueインスタンスのプロパティ

プロパティ名	説明
StringValue	システムリストデータのテキスト値
NumValue	システムリストデータの数値

Stringインスタンス

プロパティ

表 5.30. Stringインスタンスのプロパティ

プロパティ名	説明
Value	文字列の値

ScriptFieldインスタンス

プロパティ

表 5.31. ScriptFieldインスタンスのプロパティ

プロパティ名	説明
Name	スクリプト内で参照されるフィールドの名前

グローバル変数

表 5.32. グローバル変数

変数名	説明
Userlogin	データベースに接続したときのログイン
Time	Asset Manager Application Designerを実行した時刻
Date	Asset Manager Application Designerを実行した日付
Dbb.Fullname	データベース記述ファイルのフルパス名
Dbb.Shortname	データベース記述ファイルの名前（拡張子なし）
Dbb.Path	データベース記述ファイルのパス名
Dbb.Name	データベース記述ファイルの名前（拡張子付き）
Dbb.Ext	データベース記述ファイルの拡張子
Template.Fullname	データベース記述テンプレートのフルパス名
Template.Shortname	データベース記述テンプレートのパス名（拡張子なし）
Template.Path	データベース記述テンプレートのパス
Template.Name	データベース記述テンプレートの名前（拡張子付き）
Template.Ext	データベース記述テンプレートの拡張子

6 データベースの診断と修復

【アクション/データベースの診断/修復】メニューでは、既存のAsset Manager データベースの整合性を検証できます。このメニューを使用するには、まず検証するデータベースに接続する必要があります。




【分析するテーブル】フィールド：データベースの全体を分析または修復するには、【(すべてのテーブル)】を選択します。

【ログファイル】フィールド：このフィールドを使うと、データベースの診断を含むログファイルを保存できるようになります。

2つのオプションがあります。

- 【解析のみ】：Asset Manager Application Designerはデータベースの基本診断テストを実行し、問題点の修復は実行しません。
- 【修復】：Asset Manager Application Designerはデータベースを診断し、必要に応じて修復します。

【実行】をクリックし、診断と修復時に実行される操作を記録するためのログファイルを選択します。診断中、各テストの結果は次のアイコンで表されます。

-  は、テストが正常に終了したことを表します。
-  は、テストでエラーが検出されたけれども、データベースは使用可能であることを表します。
-  は、テストでエラーが検出され、データベースが使用不可能であるかもしれないことを表します。



警告:

【レコードの整合性のチェック】オプションは、有効性のスクリプトを持ち、多数のレコードを含むテーブルに対しては、かなりの時間（1日以上）がかかる場合があります。このようなテーブルでは、有効性のスクリプトはテーブル内の各レコードに対して実行されます。

Asset Managerには、GUIを使用せずに、テーブルを選択的にチェックする方法も用意されています。そのため、診断が不要と思われるテーブルはスキップできます。使用方法の詳細については、「GUI以外の診断方法」セクションを参照してください。

場合によっては、チェックが終了しないこともあります。

GUI以外の診断方法

【レコードの整合性のチェック】オプションでは、データベース全体の整合性チェックが強制されるために時間がかかる場合がありますが、Asset Managerには、必要なテーブルのみを選択可能なGUI以外の診断方法も用意されています。テーブルの整合性は、次のコマンドを使ってチェックできます。

```
amdbal -dbcheck:<Database name>;ValidityScript;Table=<SQL name of the table>"
```

例：

```
amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amComputer"
```

チェックするテーブルが複数ある場合は、次のスクリプトをバッチで実行します。

```
amdbal -dbcheck:AMDemo93jp;;ValidityScript;Table="amAsset" amdbal -dbcheck:AMDemo93jp;;ValidityScript;Table="amPortfolioItem" amdbal -dbcheck:AMDemo93jp;;ValidityScript;Table="amComputer" amdbal -dbcheck:AMDemo93jp;;ValidityScript;Table="amSoftInstall"
```

7 ODBCドライバを使ったデータベースへのアクセス

この章では、Asset Manager用に開発したODBCドライバを使ってAsset Managerデータベースにアクセス（読取り専用）する方法について説明します。

 **注意:**

Asset Manager ODBCドライバでは、データベースの情報を読み取ることしかできません。

Asset Managerデータベースへのアクセス

ODBCドライバをインストールする

Asset Managerインストールプログラムは、次のいずれかの場合にODBCドライバをインストールします。

- セットアップ時にこのパッケージを選択した場合
- ドライバが必要な他のパッケージをインストールする場合

ODBCドライバ名は、「HP Asset Manager 9.30 <言語> Driver」です。
「Adbc.dll」は、Windowsの「system32」フォルダにコピーされます。

 **注意:**

ODBCドライバは、使用するDBMSには無関係です。

ODBCドライバを使う場合

SAP Crystal Reportsなどの外部ツールでデータベースのレポートを設定する場合は、このドライバを使うことをお勧めします。

注意:

必ずしもこのドライバを使う必要はありません。お使いのレポート作成プログラムが対応するDBMSを直接サポートする場合は、直接Asset Managerデータベースにアクセスできます。

ODBCドライバ経由のアクセスには以下の利点があります。

表 7.1. ODBCドライバの利点

	ODBCドライバを使う	ODBCドライバを使わない
Asset Manager データベースにアクセスする場合のセキュリティ	レポートを使う場合は、データベースにアクセスする Asset Manager ログインとパスワードが必要です。 ログインのユーザプロファイル（ユーザ権限とアクセス制限）が適用されます。	レポートを使う場合は、DBMSに要求されるデータベースアクセスパラメータを指定する必要があります。このパラメータは、 Asset Manager のユーザプロファイルにはリンクされません。
DBMS 接続パラメータ	データベースにアクセスする際、DBMSに要求される接続パラメータを指定する必要はありません。	データベースにアクセスする際、DBMSに要求される接続パラメータを指定する必要があります。
Asset Manager データベースにアクセスする接続の選択	レポートのユーザが適切な接続を選択します。	ユーザは Asset Manager 接続を使わずに直接データベースにアクセスします。
DBMS エンジンとレポートのリンク	レポートは、使用するデータベースのエンジンには無関係です。 DBMSを変更してもレポートを変更する必要はありません。	レポートは、データベースエンジンに依存します。 DBMSを変更するとレポートも変更する必要があります。

ODBCドライバからアクセスできるデータ項目

ODBCドライバによって次の情報を参照できます。

- テーブル
- 標準フィールド
- 特殊フィールド
- 任意管理項目

上記のオブジェクトは、すべてSQL名で識別されます。



注意:

リンクは参照できません。自分でリンクを再構成する必要があります。

どのODBC接続を使うか

ODBCドライバをインストールすると、標準のODBC接続が作成されます。接続名はAsset Manager Databasesです。この接続を変更したり削除したりすることはできません。

次の2種類のODBC接続を使ってAsset Managerデータベースにアクセスできます。

- 標準のAsset Manager Databases接続
- 独自に作成した接続

標準のAsset Manager Databases接続の利点

この接続を使えば、独自に接続を作成する必要がなくなります。そのため、ODBCアドミニストレータを使う必要がありません。このAsset Manager接続は、レポートの作成時と使用時に選択します。Asset Managerの標準の接続ダイアログボックスで接続を選択します。

独自のODBC接続を作成する手順

- 1 ODBCアドミニストレータを起動します。
- 2 [HP Software Asset Manager Driver] を選択し、新しい接続を作成します。
- 3 通常の方法でODBC接続を作成します。

例：ODBCドライバでSAP Crystal Reportsのレポートを作成する

- SAP Crystal Reports (Designer) を起動します。
- 新しいレポートを開きます。
- レポートが [SQL/ODBC] データであることを指定します。
- Asset Manager Databases ODBC接続を選択します。
- 標準のAsset Manager接続ダイアログボックスが表示されます。
- 適切なAsset Manager接続を選択します。レポートを作成するのに必要なログインとそのパスワードを入力します。
- 通常どおりにレポートを作成します。

8 Citrix XenAppを使用したデータベースへのアクセス

この章では、ホストされるアプリケーションとしてのAsset Managerにアクセスする手段として、Citrix XenAppを使用する方法の概要について説明します。

概要

Citrix XenAppは、ユーザがリモートサーバ上でホストされている企業アプリケーションに接続できる、アプリケーション仮想化/アプリケーション配信用の製品です。

XenAppでは、中央サーバ上のアプリケーションをホストしてユーザがリモートでそれらのアプリケーションと対話できるようにしたり、それらのアプリケーションをローカルで実行できるようにユーザデバイスにストリームおよび配信したりできます。

Citrix XenAppは、次の設定のいずれかで使用できます。

- リモートデスクトップモード：
 - Asset ManagerがCitrix XenAppサーバにインストールされている。
 - 管理者がユーザにCitrix XenAppサーバへのアクセス権限を付与している。
 - ユーザがRemote Desktop Protocol (RDP) を使用して、Citrix XenApp リモートデスクトップセッション経由でAsset Manager Windowsクライアントにアクセスしている。
- 共有アプリケーションモード：
 - Asset ManagerがCitrix XenAppサーバにインストールされている。

- 管理者がユーザにCitrix XenAppサーバ上のAsset Managerへのアクセス権限を付与している。
- ユーザがCitrix XenAppクライアントまたはURL経由でAsset Manager Windowsクライアントにアクセスしている。

Citrix XenAppをインストールする

Citrix XenAppをインストールおよび設定する方法の詳細は、<http://support.citrix.com/productdocs/>、特に『**Getting Started with Citrix XenApp**』ガイドを参照してください。

既知の制約点

- Asset Managerの作業ウィンドウは、削除またはサイズの変更を行えません。
- ユーザがデータベースへのAsset Manager接続を作成すると（リモートデスクトップセッションまたはCitrix XenAppクライアント経由で）、その接続情報は、正しい「amdb.ini」ファイルでなく、Citrix XenAppサーバの仮想化フォルダに格納されます。

回避策：管理者は、Asset Manager Windowsクライアントを使用して、Citrix XenAppサーバへのAsset Managerの直接接続を作成する必要があります。
- ユーザは、Asset Manager Windowsクライアントを使用して、Asset ManagerデータベースへのODBC接続を作成することはできません。

回避策：管理者は、Citrix XenAppサーバ上で直接Asset ManagerデータベースへのODBC接続を作成する必要があります。

9 アーカイブ

アーカイブは、その名前が示すとおり、データをアーカイブするためのものです。対象となるデータは、データベースに全体を保存しておく必要はなくなりましたが、痕跡を残して、関連するレコードを現在のデータベースに保持しておきたいものです。例えば、除却済みのコンピュータで、会計レベルではまだ参照される可能性があるものに関するレコードをアーカイブします。

アーカイブを有効にする



警告:

DB2の場合、8Kのテーブルスペースを使用します。

▶ [DB2 UDB \[p. 52\]](#)

アーカイブはデータベースの作成前にも作成後にも有効にすることができます。どちらの場合も手順は同じです。ただし、データベース作成前にアーカイブを有効にした場合、データベース作成画面の【**アーカイブ**】フィールドで、アーカイブデータ用に特定の**テーブルスペース**（論理的なデータ保管場所）を指定することができます。このオプションは既存のデータベースに対しては使用できません。

Asset Manager Application Designerでアーカイブを有効にするには、【**データベース/アーカイブを有効にする**】メニューを使用します。アーカイブを有効にすると、データベースに対して以下の操作が実行されます。

- データベースの各テーブルに対して、アーカイブテーブルが作成されます。このテーブルは標準のテーブルの構造を複製したもので、以下のSQL名を持ちます。

ar<標準テーブル名>

 **警告:**

下記のテーブルに対してはアーカイブテーブルは作成されません。したがって、これらのテーブルのレコードに対してアーカイブを作成することはできません。

SysBlob, amAccessRestr, amAction, amCFAql, amCFScript, amCalcField, amFeatMemo, amFeatParam, amFeatScript, amFeature, amFeatureClass, amFuncRight, amHelp, amHelpMemo, amImage, amLoginSlot, amOption, amProfile, amSysConfig, amUserRight, amViewDef, amFuncDomain

Asset Manager Application Designerでは、アーカイブテーブル（複製され、接頭コードのついたテーブル）にアクセスすることはできません。アーカイブテーブルの構造は標準テーブルと同一なので、編集や変更は避けてください。テーブルに何らかの変更を加えると、関連するアーカイブテーブルのレベルで対称的な変更が自動的に実行されます。

 **注意:**

データベースの作成前にアーカイブを有効にした場合、アーカイブテーブルは、データベース作成画面の [アーカイブ] フィールドで指定したテーブルスペースに保存されます。

- アーカイブしたテーブルの外部キーのそれぞれに対して、以下のフィールドが標準テーブル（および対応するアーカイブテーブル）に作成されます。

- ar<外部キーのSQL名>

このフィールドには、アーカイブ対象のレコードの識別子が保存されます。

- arstr<タイプ識別子のない外部キーのSQL名>

このフィールドは、アーカイブ対象のレコードの説明文字列を保存します。

**注意:**

フィールドによって、タイプ識別子は次のいずれかです。

- l: 整数（32ビット）タイプのフィールド
- i: 整数タイプのフィールド
- d: 倍精度タイプのフィールド
- str: テキストタイプのフィールド
- dt: 日付+時刻タイプのフィールド
- b: ブール（はい/いいえ）タイプのフィールド

- データベースに対して追加の接続（「<元の接続の名前>-アーカイブあり」）が作成されます。

アーカイブの原則

レコードに対しては以下の4種類の主要なアーカイブ操作を実行できます。

- アーカイブ：対応するアーカイブテーブルに複数のレコードを移動します。アーカイブされたレコードは標準テーブルからは削除されます。アーカイブされたデータが、最終バックアップを上書きします。
- バックアップ：対応するアーカイブテーブルに複数のレコードをコピーします。レコードは標準テーブルからは削除されません。
- 消去：アーカイブテーブルまたは標準テーブルから複数のレコードを削除します。
- 復元：アーカイブテーブルから標準テーブルに複数のレコードを復元します。

**警告:**

消去中にアーカイブテーブルから削除されたレコードは復元できません。

前記の操作は、Asset Managerクエリ言語（AQL）の拡張によって実現されています。次の表は、各操作のシンタックスと、対象となるテーブルを示します。

操作	AQLシンタックス	対象テーブル
アーカイブ	ARCHIVE FROM <FROM句> WHERE <WHERE句>)	標準テーブル
バックアップ	BACKUP FROM <FROM句> WHERE <WHERE句>)	標準テーブル
Purge	PURGE FROM <FROM句> WHERE <WHERE句>)	標準テーブルとアーカイブテーブル

操作	AQLシンタックス	対象テーブル
復元	RESTORE FROM <FROM句> WHERE <WHERE句>	アーカイブテーブル

 **注意:**

AQLと、FROM句およびWHERE句のシンタックスの詳細については、『高度な使い方』ガイドの「AQLクエリ」の章を参照してください。

アーカイブされたレコードに関連するリンク

詳細画面では、アーカイブされたレコードを参照する単純リンク（基数1-n）は灰色表示されます。

例

データアーカイブの例

以下に示すのは、過去90日間変更されていないクローズされた購入依頼をアーカイブする例です。この場合、**[amRequest]** テーブルのレコードのうち以下の条件を満たすものをアーカイブするのが適切な方法です。

- **[dtLastModif]** フィールドで参照される日付が90日より前。
- **[seStatus]** フィールドの値が **[41]** 。

 **注意:**

リストデータの値は整数に関連付けられています。ここで必要な値は、**[クローズ]** に対する値、すなわち41です。システムリストデータの値を知るには、リストが使用されているフィールドにフォーカスを置き、**[Shift]** キーと **[F1]** キーを同時に押します。フィールドのヘルプが表示され、システムリストデータの値と対応する整数のリストが見られます。

この例の問題を解決するには、**[amRequest]** テーブルに対するアクションを作成します。このアクションは、先に定義した条件（クエリのWHERE句で指定したもの）に一致するレコードに対するAQLアーカイブクエリを実行します。

- 1 **[ツール/アクション/編集]** を選択し、**[新規作成]** をクリックして新しいアクションを作成します。
- 2 アクションの**[名前]** を選択し、**[依頼]**（amRequest）テーブルを**[コンテキスト]** として選択します。アクションタイプは**[スクリプト]** です。

**注意:**

【コンテキスト】の定義は必須ではありません。ワークフローを使ってこのアクションを定期的に行う場合には、コンテキストを定義することはできません。

- 3 アクションの【スクリプト】タブに以下のスクリプトを入力します。

```
RetVal = amDbExecAql("ARCHIVE FROM amRequest WHERE dtLastModif < AddDays(getdate(), -90) AND seStatus = 41")
```

**注意:**

amDbExecAql関数は、データベースのクエリを実行します。この関数の詳細については、『**Programmer's reference**』を参照してください。

- 4 アクションの作成を承認します。
- 5 このアクションは、ワークフローから呼び出すことも、購入依頼画面から直接実行することもできます。

アーカイブを制御するには：

- 1 【資産ライフサイクル／調達ライフサイクル／依頼／購入依頼】ナビゲーションリンクを使って、購入依頼画面を開きます。
- 2 購入依頼のリストを右クリックし、以前に作成したアクションをショートカットメニューから選択します。

クローズした購入依頼に対応し、90日間より長く変更されていないレコードは、アーカイブされてリストから消えます。

ただし、他のレコードから参照されている場合は、それらのレコードの詳細に灰色で表示されます。管理者権限を持つユーザは、🔍をクリックすることによりアーカイブされたレコードを表示することができます。

データ復元の例

間違ってアーカイブした購入依頼を復元したい場合は：

- 1 アーカイブ接続を使用してデータベースに接続します。
- 2 【ツール／アクション／編集】を選択し、【新規作成】をクリックして新しいアクションを作成します。
- 3 アクションの【名前】を選択し、【arRequest】テーブル（【amRequest】テーブルのアーカイブテーブル）をアクションの【コンテキスト】として選択します。アクションのタイプは【スクリプト】です。

**注意:**

[コンテキスト] の定義は必須ではありません。ワークフローを使ってこのアクションを定期的に行う場合には、コンテキストを定義することはできません。

- 4 アクションの [スクリプト] タブに以下のスクリプトを入力します。

```
RetVal = amDbExecAql("RESTORE FROM arRequest WHERE lReqId=" & [lReqId])
```

**注意:**

amDbExecAql関数は、データベースのクエリを実行します。この関数の詳細については、『**Programmer's reference**』を参照してください。

- 5 アクションの作成を承認します。
- 6 [管理/画面一覧] メニューでアーカイブされた購入依頼 (**arRequest**) の画面を開き、アーカイブされた要求を選択します。
- 7 右クリックして、前に作成したアクションをショートカットメニューから選択します。
レコードが取得され、アーカイブから削除されて、本番データベースに復元されます。

リンクされたレコードの処理

アーカイブ操作中のリンクされたレコードの処理方法は、リンクのタイプによって異なります。次の表は、リンクタイプごとのアーカイブの動作を示します。

リンクタイプ	処理
--------	----

OWN	アーカイブ操作では、リンクされたレコードはアーカイブ対象のレコードとして扱われます。レコードをアーカイブすると、リンクされたレコードのすべてが同時にアーカイブされます。レコードを消去すると、リンクされたレコードのすべてが同時に消去されます。 例えば、資産とその従属資産は、同時にアーカイブ、コピー、復元、消去されます。
-----	--

リンクタイプ 処理

DEFINEまたはNORMAL リンクされたレコードの外部キーは0にリセットされます。アーカイブフィールド (arlc<名前>Idとarstr<名前>) には、アーカイブ対象のレコードの識別子と説明文字列がそれぞれ入力されます。

例えば、経費明細をアーカイブした場合、その経費明細にリンクされた資産のレコードの対応する外部キーは0にリセットされます。

任意管理項目が関係している場合のアーカイブ機能の動作

レコード「**A**」と、関連する任意管理項目「**C**」（[タイプ]が[リンク]以外のもの）を考えます。

- 「**A**」をアーカイブした場合、「**C**」もアーカイブされます。
- 「**C**」をアーカイブした場合、レコード「**A**」とその任意管理項目「**C**」との間のリンクは切れ、レコードはレコード詳細に表示されなくなります。このリンクは以下の2つの場合に復元されます。
 - 1 後に「**A**」をアーカイブした場合。この場合、「**A**」と「**C**」との間のリンクはアーカイブデータの中で再設定されます。
 - 2 「**C**」を復元した場合。この場合、「**A**」と「**C**」との間のリンクはデータベースの中で再設定されます。

レコード「**A**」とそれに関連する[タイプ]が[リンク]の任意管理項目「**C**」、および任意管理項目「**C**」が指している別のレコード「**D**」を考えます。

- 「**A**」をアーカイブした場合、「**C**」はアーカイブされますが、「**D**」は自動的にアーカイブされません。
- 「**C**」をアーカイブした場合、「**A**」と「**C**」との間のリンク、および「**C**」と「**D**」との間のリンクは切れます。任意管理項目はレコード「**A**」の詳細には表示されなくなります。「**A**」、「**C**」、「**D**」の間のリンクは、以下の2つの場合に復元されます。
 - 1 後に「**A**」と「**D**」をアーカイブした場合。この場合、「**A**」、「**C**」、「**D**」の間の関係はアーカイブデータの中で再設定されます。
 - 2 「**C**」を復元した場合。この場合、「**A**」、「**C**」、「**D**」の間の関係はデータベースの中で再設定されます。
- 「**D**」をアーカイブした場合、「**C**」と「**D**」との間のリンクは切れます。「**A**」、「**C**」、「**D**」の間のリンクは、以下の2つの場合に復元されます。
 - 1 後に「**A**」をアーカイブした場合。この場合、「**A**」、「**C**」、「**D**」の間の関係はアーカイブデータの中で再設定されます。
 - 2 「**D**」を復元した場合。この場合、「**A**」、「**C**」、「**D**」の間の関係はデータベースの中で再設定されます。

アーカイブの無効化

アーカイブはいつでも無効にすることができます。アーカイブを無効にするということは、以前にアーカイブが有効にされており、データベースがアーカイブ専用の追加のテーブルを持つように変更されていることを意味します。

アーカイブを無効にするには、**【データベース/アーカイブを有効にする】**オプションをクリアします。



アーカイブを無効にすると、アーカイブテーブルが削除され、アーカイブされたデータはすべて破棄されます。

10 非正規化

Asset Managerデータベースは極めて高度に正規化されます。具体的には、このデータベースのプロパティでは、有効で合理的なリレーショナルデータベースが得られるように、テーブル間の不要な冗長性や依存関係が排除されます。

非正規化はこれとは反対のプロセスと考えられます。値を複数の場所に格納して自発的な冗長性を確保することにより、データの処理と検索を高速化し、全体的なパフォーマンスを向上させることができます。

値を正規化した方がいいのはどんな時ですか？

その時期を決めるルールは特にありませんが、データベースの非正規化をすべきかどうかを判断するためのいくつかの要素があります。

- データベースのサイズ。データベースの規模が小規模または中規模の場合、非正規化を行ってもあまり効果がありません。インスタンスによっては、非正規化がパフォーマンスに悪影響を与えることがあります。例えば、非正規化によりテーブルのサイズが大幅に増加し、データ検索が高速化してもレコードの更新が遅くなってしまう場合などです。
- 使用する最重要クエリと期待する応答時間。これらのクエリで複数のテーブルに分散するデータを使用したり、クエリが答えを返す前に1つまたは複数の列で計算を実行する場合は、クエリで返される値を正規化すると有効なことがあります。
- パフォーマンス。アプリケーションのパフォーマンスが十分であると考えられる場合は、非正規化を適用する必要はありません。

Asset Managerの非正規化の原則

非正規化を行うには理論上いくつかのテクニックがありますが、Asset Managerはその中の1つ「列の非正規化」にのみ対応しています。クエリ内でテーブル間の結合が作成されないように、同じ列（すなわちフィールド）が複数のテーブルで作成されます。

例として、従業員の不在レポートのケースを挙げます。このレポートは非常に頻繁に作成されます。このレポートに入力が必要なデータの多くは「不在」テーブルに保存されています。ところがレポートで使用するデータ項目の1つである、従業員の責任者の電子メールアドレスは別のテーブル【部署と従業員】テーブルに保存されています。このような場合は、レポート作成時の2つのテーブル間の結合を回避するため、このデータの非正規化を考えてみる価値があります。

この値の非正規化を実行すると、このフィールドは「不在」テーブルに複製されます。ただし、この操作は非正規化がもたらす典型的な潜在リスク、すなわち冗長データをそのまま保持することにつながります。Asset Managerは、正規化されたデータの整合性を自動的に維持するため、このようなリスクを排除することができます。

一般的に言えば、値の非正規化はユーザに対して完全に透過的に行われる操作です。複製された値は詳細画面に表示されることはなく、AQLクエリの書き方も変わりません。Asset Managerは非正規化された値の存在を自動的に検出し、すぐにクエリを単純化します。

注意:

ただし、非正規化により作成されたフィールドはAsset Manager Application Designerで表示されます。それらのSQL名は接頭コードdfを使用し、リンクは接頭コードlnを使用します。

値の非正規化

非正規化が妥当であると確定したら、Asset Manager Application Designerを使用して実際に操作を行います。

- 1 Asset Manager Application Designerでデータベースに接続します。
- 2 非正規化フィールドまたはリンクのターゲットテーブルを選択し、次に【データベース/非正規化を追加】メニューを選択します。続いて、非正規化するフィールドを選択する画面が表示されます。

**注意:**

【非正規化フィールドにインデックスを作成】ボックスをオンにすると、この値のソート操作が高速化します。

【新規作成】をクリックします。

- 3 変更を保存し、データベース更新ウィザードに従って操作し（ログファイルの選択、オプションの設定ファイルの選択などの操作）、【完了】をクリックします。
- 4 非正規化されたフィールドまたはリンクがアクティブになり、状況に応じてクエリによって自動的に使用されます。

例

前述の、従業員の不在レポートを作成した例について考えましょう。このレポートとこのレポートに含まれるデータは頻繁に検索されるため、部署の責任者の電子メールアドレスを格納するフィールドの非正規化を行うことが適切です。

【**Manager.Email**】リモートフィールドは非正規化が必要です。これを行うには：

- 1 Asset Manager Application Designerで【不在】テーブル（**amAbsence**）を選択し、【データベース／非正規化を追加】メニューを選択します。
- 2 続いて表示される画面で、ドロップダウンリストから【**Manager.Email**】リモートフィールドを選択します。このリストには【不在】テーブルのフィールドとリンクが表示され、リンクに含まれるリモートフィールドを選択することができます。ここに示す例では、【責任者】リンクをすべて表示して、関連する【電子メール】フィールドを選択してください。
- 3 【作成】をクリックします。【不在】テーブルに、接頭コード「**df**」が付いた新規フィールドが表示されているはずです。【ファイル／データベース構造の更新】メニューを選択し、変更を確定します。データベース構造の保存ウィザードが表示されます。
- 4 【次へ】をクリックし、次にログファイルのフォルダを選択します。
- 5 【次へ】をクリックします。次のページにオプションの変換ファイルを設定する画面が表示されます。このページは非正規化には用いられません。
- 6 【完了】をクリックします。ウィザードには処理中であることを知らせる画面が表示され、データベースが正常に更新されると、それを知らせる画面になります。レポートで使用するクエリは非正規化フィールドを用いるため、【不在】テーブルと【部署と従業員】テーブルとの結合を回避することができます。

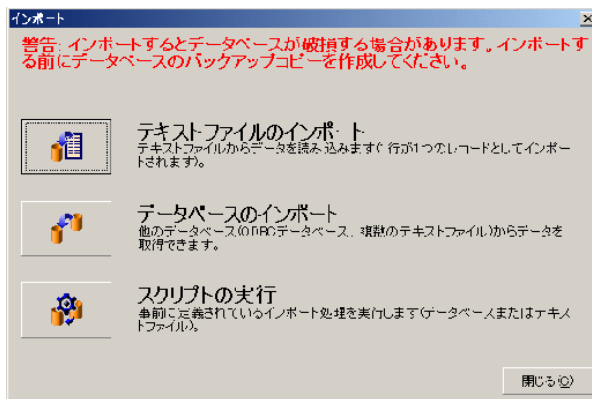
11 インポート

この章では、Asset Managerでデータをインポートする方法について説明します。

1つのテキストファイルをインポートするには、[ファイル/インポート] メニューの [テキストファイルのインポート] オプションを使います。

複数のテキストファイルをインポートするには、[ファイル/インポート] メニューの [データベースのインポート] オプションを使います。

図 11.1. インポートするデータのタイプを選択する



データのインポートの概要

Asset Manager管理者は、次のファイルやデータベースのデータをAsset Managerデータベースにインポートできます。

- 1つのテキストファイル
- 複数のテキストファイル
- 完全な1つのデータベース

1つのテキストファイルからデータをインポートする

テキストファイルは、Asset Managerデータベースのテーブルにマップされます。

管理者は、テキストファイルの各フィールドを、関連テーブルまたはリンク先のテーブル内のデータベースフィールドにマップします。



注意:

テキストファイルで使用する文字のISOコードは、Asset Managerのコードと同一でなくてはなりません。

複数のテキストファイルからデータをインポートする

各テキストファイルは、Asset Managerデータベースのテーブルにマップされます。

管理者は、テキストファイルの各フィールドを、関連テーブルまたはリンク先のテーブル内のデータベースフィールドにマップします。

完全な1つのデータベースからデータをインポートする

ODBCデータベースをインポートできます。

管理者は、ソースデータベースの各テーブルを、ターゲットデータベースのテーブルにマップします。

管理者は、ソーステーブルの各フィールドを、ターゲットデータベースのテーブルのフィールドにマップします。このフィールドは関連テーブルまたはリンク先のテーブルに属します。

Asset Managerレコードを追加または変更することはできますが、削除することはできません。

転送に関する情報はすべてスクリプトとして保存できます。スクリプトを使うと、再度データをインポートするときにインポート設定を定義する必要がありません。

インポートモジュールでは、エラーの処理方法と実行された操作の結果を、ログファイルに書き込むかどうかを選択できます。

 **注意:**

データベースの保護に関する注意：[ファイル/インポート]メニュー（「Admin」ログイン）を使用できるのは、Asset Managerの管理者または管理者権限を持つユーザだけです。このメニューは、他のユーザのインタフェース上では無効になっています。これは、データベースへのアクセスを保護するためです。

数値データをインポートする

数値型フィールドの値には、Windowsのコントロールパネルで定義されているオプションに関係なく、適切な形式を使用してください。例えば、0から9のみを使い、少数点にはピリオド（.）を使う、などです。数値型フィールドの値は数値型フィールドにインポートする必要があります。このように数値データは、コンピュータまたはインポート処理を実行するコンピュータの設定と別の設定でインポートできます。

推奨事項

Asset Managerデータベースにデータをインポートする前の推奨事項について説明します。

必須フィールドのデフォルト値

インポートモジュールは、フィールドの必須属性を認識しません。このため、インポート時に必須フィールドが空の状態になることを防ぐために、必須フィールドにデフォルト値を割り当てておくことをお勧めします。インポートされるファイルで値が指定されている場合は、ターゲットテーブルのフィールドのデフォルト値はこの値に置き換えられます。

「Id」フィールドを照合更新キーとして使わない

出力したデータを再インポートする場合は、照合更新キーとしてテーブルの「Id」フィールドを使わないようにしてください。実際には、対応するID番号は一定ではなく、変更されることがあります。資産の資産タグなど、値が「変化しない」キーを使います。

Asset Managerデータベースのバックアップ

インポートすると、Asset Managerデータベース全体が変更されるので、インポートを実行する前に、データベースのバックアップを作成してください。

Asset Managerデータベースへの同時アクセスを避ける

異なるマシンから同時にインポートを実行したり、インポート中に別のマシンからAsset Managerデータベースにアクセスしたりしないでください。

フィールドにデータをインポートする際の制約事項

インポートモジュールは、Asset Managerデータベースにデータをインポートするときに、データがデータベースの構造と互換性があるかどうかを確認します。インポートするデータの特定のフィールドまたはリンクにデータベース構造との互換性がない場合は、次のように処理されます。

- レコード自体がインポートされない。
- 値が切り捨てられる。
- フィールドがそのまま残される。

インポートエラーを防ぐには、インポートするデータとデータベースの構造に一貫性があることを確認する必要があります。

インポート時には次の制約点が考えられます。

- 構造化されたデータベースをインポートする場合
 - ソースフィールドとターゲットフィールドのタイプが同じである場合（ソースが日付でターゲットも日付など）、またはフィールドに互換性がある場合（ソースが「日付+時刻」でターゲットが「日付」など）は、制約はありません。
 - ソースフィールドが「テキスト」タイプのフィールドで、ターゲットフィールドがより特殊なタイプの場合は、ターゲットフィールドの制約を考慮する必要があります。
- テキストファイルからデータをインポートする場合
 - ターゲットフィールドの制約を考慮する必要があります。

ターゲットフィールドの [UserType]（入カタイプ）プロパティの値

値	結果
デフォルト	「データ型」（タイプ）プロパティで定義されている形式に従う必要があります。

値	結果
数値または金額	<p>ソースフィールドは数値でなければなりません。</p> <p>ソースフィールドが構造化データベースのフィールドで、フィールドのデータ型が「数値」または「金額」の場合は、制約はありません。</p> <p>ソースフィールドが「テキスト」型のフィールドの場合は、小数点としてピリオド (.) を使って数値を表します。千の位に区切り文字は使用できません。</p>
ブール (はい/いいえ)	<p>ソースフィールドは「1」（「はい」の場合）、または「0」（「いいえ」の場合）のいずれかでなければなりません。</p>
日付	<p>ソースフィールドが構造化データベースのフィールドで、フィールドのデータ型が「日付」または「日付+時刻」の場合は、制約はありません。</p> <p>ソースフィールドが「テキスト」型のフィールドの場合は、次の制約に従います。</p> <ul style="list-style-type: none"> ■ 日付形式（年、月、日の順序）をすべてのレコードで同じにする。インポート時にこの形式を指定。 ■ 日、月、および年を系統的に入力する。 ■ すべてのレコードで日、月、および年を同じ文字で区切る（任意の文字）。インポート時にこの区切り文字を指定。
システムリストデータ	<p>リストデータの値を1つだけにしておく必要があります。複数の値にすると、その行はインポートされません。リストデータの値は、その値自体またはデータベースに格納されている数値で識別できます。</p> <p>例</p> <p>リストデータが「はい 1 いいえ 0」の場合、「はい」または「1」だけをインポートしても、同じ結果になります。</p> <p>値を空にすると、インポートモジュールにより、フィールドには値「0」が割り当てられます。</p> <p>インポートするデータには格納されている数値を使うようにします。これは、Asset Managerのバージョンや言語バージョンが異なる場合も、テキスト値より安定しているためです。</p>
カスタムリストデータ	<p>フィールドの値は、ユーザーが変更できるドロップダウンリストから選択します。ユーザーは、リストデータの既存の値の1つを選択できます。さらに、リストデータが「オープン」タイプの場合は、適切な権限を持つユーザーが新しい値を入力できます。この値はリストデータに追加されます。</p>
パーセント	<p>インポートする値は、パーセント値でなければなりません。パーセント記号 (%) は付けても付けなくてもかまいません（「10」または「10%」）。</p>
期間	<p>[UserType] が「期間」の場合、[UserType] および [UserTypeFormat] プロパティで定義されている制約に従う必要があります。</p>

値	結果
Table or field SQL name	任意の英数字の値をインポートできます。ただし、この値がフィールドまたはテーブルの有効なSQL名と一致しない場合は、データベースが破損するおそれがあります。

ターゲットフィールドの [データ型] (タイプ) プロパティの値

[UserType] (入力タイプ) プロパティがデフォルトに設定されている場合、このプロパティにリンクされている制約に従う必要があります。

特殊なケース

値	結果
Date and time	ソースフィールドが構造化データベースのフィールドで、フィールドのデータ型が「日付+時刻」または「日付」の場合は、制約はありません。 ソースフィールドが「テキスト」型のフィールドで、[UserType] プロパティが「日付」または「日付+時刻」の場合は、[UserType] プロパティで定義されている制約に従う必要があります。

ターゲットフィールドの他のプロパティの値

プロパティ名	値	結果
MandatoryType	Yes	ソースフィールドが空の場合は、インポートモジュールによって追加または変更されるはずのレコードはそのまま残ります。
Size	指定値	ソースフィールドの値が長すぎる場合は、インポート時に切り捨てられます。
ReadOnly	Yes	このプロパティを持つフィールドには、値をインポートできません。

[diLastModif] フィールド (SQL名)

このフィールドは、ユーザインタフェースまたはインポートによって Asset Manager データベースのレコードを変更または作成すると更新されます。レコードの変更または作成の日付を示します。

このフィールドに値をインポートすると、実際のインポート日とその値で上書きされます。

テキストファイルまたはODBCデータベースをインポートする

このセクションでは、1つまたは複数のデータファイルまたはODBCデータベースをインポートする方法について説明します。

1つのテキストファイルをインポートするには、【ファイル/インポート】メニューの【テキストファイルのインポート】オプションを使います。

テキストファイルのセットをインポートするには、【ファイル/インポート】メニューを使用します。【データベースのインポート】オプション、【テキスト】タブを順にクリックします。

ODBCデータベースをインポートするには、【ファイル/インポート】メニューを使用します。【データベースのインポート】オプション、【ODBC】タブを順にクリックします。

テキストファイルをインポートする前に

インポートするデータを含むファイルの準備から始めます。次の条件を満たしている必要があります。

- 各列はフィールドを表す。
- 各行はレコードを表す。
- ファイルの始まりに、オプションでフィールド名を含めることができる。ファイルに含めない場合は、データのインポート時に定義できます。
- ファイルの始まりのフィールド名を含むオプション行の前に、インポートされない99行までのコメントを含むことができます。これは【最初のインポート行】フィールドに入力して行います。これらのコメント行は特殊文字で始まる必要がありません。
- OEM (DOS)、UNICODE または Latin1文字セットを使ってファイルを作成できる。
- 列は固定幅でも可変幅でもかまわない。後者の場合、区切り文字として機能する文字を選択します。
- フィールドの内容は、選択した任意の文字を使って区切ることができる。
- フィールドには以下のタイプがあります。
 - 数値
 - 文字列
 - 日付
 - 日付+時刻
- インポートする値は、Asset Managerデータベースの構造（データ入力形式、フィールドのデータ型、リンクタイプ、インデックスタイプなど）に関連する制約に従う必要がある。
- テキストの区切り文字に指定した文字を除くすべての文字を使用できる。テキスト文字列に区切り文字を含めることはできません。
- 「日付」、「日付+時刻」、および「期間」データは、プログラムに直接入力する場合と同じ制約に従う。
- ターゲットの各メインテーブル用に1つのテキストファイルを作成するようにする。

ソースデータが、インポートモジュールでサポートされていないデータベースにある場合は、データをテキストファイルに抽出し、後でこれらをインポートする必要があります。Asset Managerが認識できるデータベースの場合は、【ファイル/インポート】メニューの【データベースのインポート】オプションを使って、データベースから直接情報をインポートできます。

手順1：インポートするテキストファイルまたはODBCデータベースを選択する

1つのテキストファイルをインポートする

- 1 【ファイル/インポート】メニューを選択します。
- 2 【テキストファイルのインポート】オプションを選択します。
- 3 インポートするデータを含む構造化テキストファイルの場所を指定するダイアログボックスが表示されます。

複数のテキストファイルをインポートする


インポートするテキストファイルを選択するには：

- 1 【ファイル/インポート】メニューから【データベースのインポート】オプションを選択し、【テキスト】タブページを表示します。
- 2 【開く】をクリックします。
- 3 インポートモードになったら、【ファイル/ファイルの追加】メニューを選択します。追加するテキストファイル名を指定します。

ODBCデータベースをインポートする

【ファイル/インポート】メニューの【データベースのインポート】オプションを使って、ソースデータベースを選択します。

【ODBC】タブページを使って、ODBCデータベースをインポートします。

- 1 データソース、ユーザ、およびパスワードを指定します。 アイコンを使うと、Windowsコントロールパネルの【ODBC Data Source Administrator】を使わずに、直接ODBCデータソースを作成できます。
- 2 【開く】をクリックします。

手順2：インポートするファイルまたはテーブルの解読方法を定義する

重要項目：

ファイルに固定幅のフィールドが含まれている場合、この手順ではウィンドウを1つだけ使います。フィールド間に区切り文字が含まれているファイルの場合は、2つのウィンドウが必要です。

1つのテキストファイルをインポートする場合

手順1でテキストファイルを選択すると、その解読方法を指定するダイアログボックスが自動的に表示されます。

複数のテキストファイルをインポートする場合

手順1でテキストファイルを選択すると、その解読方法を指定するダイアログボックスが自動的に表示されます。【OK】をクリックしてこの手順を後で実行することも、一度に情報を入力することもできます。

インポートモジュールのメイン画面でこのウィンドウを呼び出す方法は2つあります。

- ソースファイルをダブルクリックする。
- ソースファイルを選択した後、【編集／プロパティ】メニューを使う。

インポートモジュールのメイン画面の【ソーステーブル】列に、テキストファイルのリストが表示されます。

ODBCデータベースをインポートする場合

手順1でデータベースを開くと、次の方法でインポートモジュールのメイン画面から各テーブルの詳細を表示できます。

- ソーステーブルをダブルクリックする。
- ソーステーブルを選択してから【編集／プロパティ】メニューを使う。

インポートモジュールのメイン画面の【ソーステーブル】列にソースファイルのリストが表示されます。

最初の画面

文字コーディング

テキストが次の文字セットのどれを使用しているかを指定します：ANSI、OEM（DOS）、UTF-8、UNICODE、またはLatin 1。

最初のインポート行

インポートする1行目のデータを含む行番号を入力します。この番号の前にある行は飛ばされます。

ドキュメントにフィールド名を含む行があり、この行が最初のデータ行のすぐ前にある場合は、フィールド名の行番号を入力します。

Asset Managerでは、ファイルの始まりの99行までを飛ばすことができます。

最初にインポートする行に列名を含める

インポートする最初の行にフィールド名（列名）が含まれている場合は、このチェックボックスをオンにします。こうすると、自分で列名を入力する必要がありません。

ファイルに列名が含まれていない場合は、次の手順で定義できます。

区切り文字

フィールド値が特定の文字で分離されている場合は、このチェックボックスをオンにします。

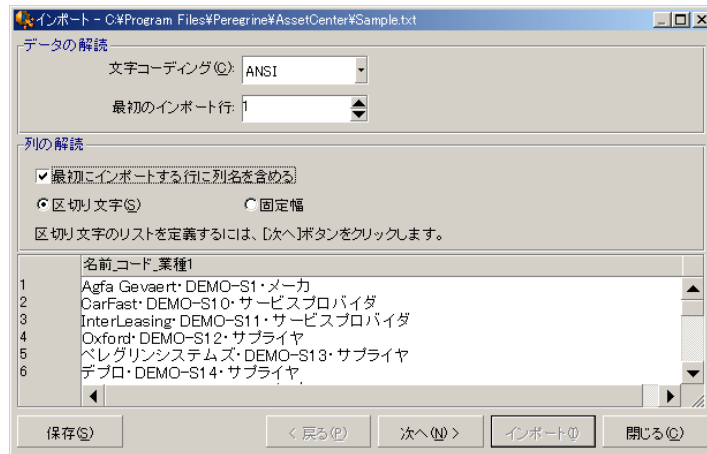
次の画面でこの文字を指定します。

固定幅

各フィールドのすべての値が同じ長さの場合は、このチェックボックスをオンにします。

Asset Managerにより、自動的に列の境界が表示されます。

- 列の境界を移動するには、マウスを使ってデータ領域の境界を選択し、希望の位置までドラッグします。タイトル領域で境界を選択することはできません。
- 列の境界を削除するには、マウスを使って境界を選択し、テーブルの外までドラッグします。
- 新しい列の境界を追加するには、データ領域の区切り文字を挿入する位置でクリックします。



注意:

画面下部に、インポートするファイルのプレビューが表示されます。最大25行まで表示されます。

2番目の画面

この2番目の画面は、最初の画面で次のことを行った場合に表示されます。

- 1 フィールド値を区切り文字で区切るように指定した場合。

2 [次へ] をクリックした場合。

列の区切り

連続する2つのフィールド値を区切るために使う文字を指定します。区切り文字にはセミicolon (;) を使うことをお勧めします。

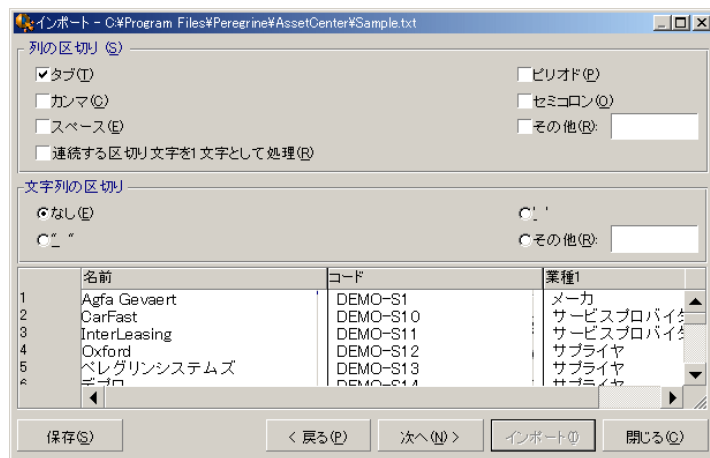
[連続する区切り文字を1文字として処理] オプションをオンにすると、連続する同じ区切り文字は1つとして扱われ、空の列は作成されません。空の列を作成する場合は、このチェックボックスをオフにして、テキストファイルで2つの区切り文字を続けて使います。

文字列の区切り

テキストを区切る文字を使う場合は、その文字を指定します。Asset Managerでは、データベースにフィールドを転送する前に、これらの余分な文字がある場合は、取り除かれます。

2つの区切り文字の間に列の区切り文字が検出された場合は、テキストとして見なされます。文字列の区切り文字は、すべての値で一貫して使わなければならないわけではありません。ただし、文字列の始まりの前に文字列の区切り文字を使った場合は、文字列の最後にも区切り文字を使ってバランスをとる必要があります。

文字列の区切り文字を値としてインポートすることはできません。



手順3：インポートするフィールドの詳細を指定する

1つのテキストファイルをインポートする場合

手順2の説明に従ってテキストファイルの解読を行ったら、[次へ] ボタンをクリックして、ファイルのフィールドの詳細を指定する画面を表示します。

複数のテキストファイルをインポートする場合

手順2の説明に従ってテキストファイルの解読を行ったら、[次へ] ボタンをクリックして、ファイルのフィールドの詳細を指定する画面を表示します。

このウィンドウは、インポートモジュールのメイン画面からも表示できます。ソースファイルをダブルクリックするか、ファイルを選択して[編集/プロパティ] メニューを選択すると、前の手順と同じファイルの解読を指定するウィンドウが表示されるので、[次へ] ボタンをクリックします。

ODBCデータベースをインポートする場合

手順1でデータベースを選択したら、インポートモジュールのメイン画面からテーブルの詳細を表示できます。ソーステーブルをダブルクリックするか、ソーステーブルを選択して[編集/プロパティ] メニューを選択します（インポートモジュールのメイン画面の「ソーステーブル」列にはソーステーブルの一覧が表示されます）。

設定する列をクリックして選択します。

番号

ここには、選択した列の番号が表示されます。

手順2で[最初にインポートする行に列名を含める] を選択していない場合は、テーブルをクリックせずに、列番号を直接選択できます。

名前

ここには列名（またはフィールド名）が表示されます。

- 1 手順2で[最初にインポートする行に列名を含める] を選択した場合、列名は変更できません。
- 2 このオプションを選択しなかった場合は、デフォルト名をそのまま残すか変更します。この名前を使うと、後の手順で列を識別しやすくなります。

タイプ

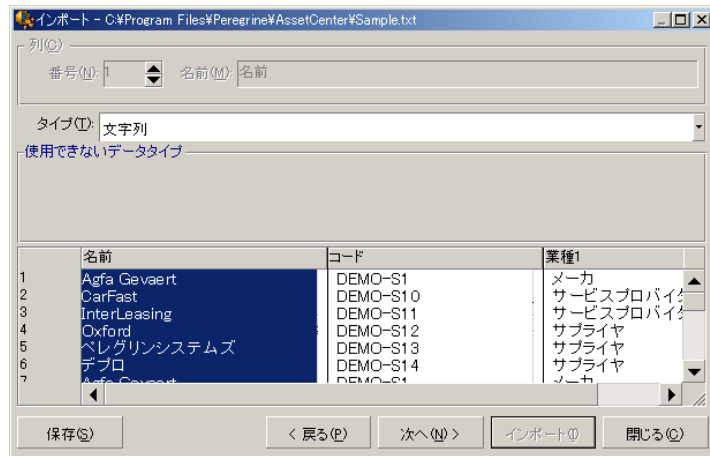
インポートするフィールドのデータ型を指定します。次の選択肢から選択できません。

- 数値：すべての文字が数字でなければなりません。他の型の文字が検出されると、フィールドの値は「0」に設定されます。
- 文字列：テキストの区切り文字として使う文字以外のすべての文字を使用できます。
- 日付：「日付形式」フレームで定義した日付形式だけが受け入れられます。インポート時に他の形式が検出されると、フィールド値はNULLに設定されます。

日付形式

フィールドのデータ型を「日付」に指定すると、日、月、年の区切り文字とその表示順を指定できるオプションが表示されます。

これら2つのオプションのほか、**Asset Manager**で日付の入力に使用できるすべてのオプションを、日付に使用できます。



注意:

画面下部のテーブルには、インポートする行が最大25行まで表示されます。

手順4：ソースフィールドをAsset Managerデータベースのターゲットフィールドにマップする

1つのテキストファイルをインポートする

手順3の説明に従ってソースフィールドを選択したら、**[次へ]** ボタンをクリックして、テキストファイルのフィールドを**Asset Manager**データベースのフィールドにマップする画面を表示します。次の操作を行います。

- 1 [ターゲットテーブル] フィールドを使って、テキストファイルを**Asset Manager**データベースのテーブルにマップします。
- 2 次に、インポートするテキストファイルのフィールド（[ソースフィールド] テーブルに表示される）を**Asset Manager**データベースのフィールドにマップします。（ターゲットテーブルおよびそのリンクテーブルのフィールドは、右側の [ターゲットテーブル] フィールドの下に表示されます）。

複数のテキストファイルをインポートする

- 1 手順3の説明に従ってソースフィールドを選択したら、**[OK]** ボタンをクリックして、インポートモジュールのメイン画面に戻ります。
- 2 各テキストファイルをターゲットテーブルにマップします。
- 3 次に、右側のテーブルの各組合せ（テキストファイル、ターゲットテーブル）ごとに、各テキストフィールドを**Asset Manager**テーブルのフィールドにマップします。組合せをダブルクリックするか、組合せを選択して**[編集/プロパティ]** メニューを使います。

ODBCデータベースをインポートする

- 1 手順3の説明に従ってソースフィールドを選択したら、**[OK]** ボタンをクリックして、インポートモジュールのメイン画面に戻ります。
- 2 個々のソーステーブルをターゲットテーブルにマップします。
- 3 次に、右側のテーブルの各組合せ（ソーステーブル、ターゲットテーブル）ごとに、ソーステーブルからのフィールドを**Asset Manager**データベースのフィールドにマップします。組合せをダブルクリックするか、組合せを選択して**[編集/プロパティ]** メニューを使います。


手順5：各テキストファイルまたはソーステーブルをターゲットテーブルにマップする


注意:

このセクションは、複数のテキストファイルまたはODBCデータベースのインポートに関する説明です。

個々のテキストファイルまたはソーステーブルをターゲットテーブルにマップする

ファイルまたはソーステーブル（[ソーステーブル] 列）および対応するターゲットテーブル（[ターゲットテーブル] 列）をクリックします。次に、以下のいずれかを実行します。

- **[編集/マッピング]** メニューを使う。
-  アイコンをクリックする。
- または、**[編集/マップ]** メニューを使用すると、**Asset Manager**が同じ名前のファイルまたはテーブルを自動的にマップします。この場合、フィールドの技術名が使われます。

ターゲットテーブルからファイルまたはソーステーブルのマッピングを取り消すには、**[編集/マッピングの解除]** メニューまたは  アイコンを使います。

Asset Managerデータベース構造を表示する

[ターゲットテーブル] 列でターゲットテーブルをダブルクリックするか、マウスで選択して [編集/プロパティ] メニューを使います。フィールド、タイプ、および長さのリストが表示されます。

手順6：インポートするフィールドをAsset Managerデータベースのフィールドにマップする

ソースフィールド

画面のこの部分には、手順3で列に割り当てた名前（テキストファイルの場合）または短いフィールドの説明（Asset Managerデータベースの場合）が表示されます。

ターゲットテーブル

1つのテキストファイルをインポートする場合




データを受け取るターゲットテーブルを選択します。テーブルの構造（テーブルまたはリンク先のテーブルのフィールド）が表示されます。

複数のテキストファイルまたはODBCデータベースをインポートする場合

ファイルまたはソーステーブルにマップされるターゲットテーブルの構造が表示されます（テーブルまたはリンク先のテーブルのフィールド）。




ソースフィールドをターゲットフィールドにマップする

いくつかの方法があります。

- マウスを使って「ソースフィールド」を「ターゲットフィールド」にドラッグしてマップします。
- 「ソースフィールド」と「ターゲットフィールド」を選択し、 アイコンをクリックしてマップします。
-  アイコンを使うと、ソースフィールドとターゲットフィールドの組合せをクリックした後で、ターゲットフィールドからソースフィールドのマップを取り消すことができます。
-  アイコンを使うと、ソースフィールドを自動的に同じ名前のターゲットフィールドにマップできます。この場合、フィールドの技術名が使われます。



ソースファイルに特殊フィールドを追加する

Asset Managerでは、ソースファイルに追加フィールドを追加できます。これらのフィールドは保存されず、メモリに格納されるだけです。

これらの追加フィールドを追加、削除、または表示するには、、 および  アイコンを使います。

キーを選択する

1つまたは複数のターゲットフィールドを選択して、レコード識別キーを作成できます。識別キーで、テーブル内のレコードを識別できます。複数のキーを選択して同時にそれらに対応する複数のレコードを識別することもできます。

キーを作成するソースフィールドとターゲットフィールドの組合せを選択し、 ボタンをクリックしてこれらを「キー」として宣言します。このボタンがアクティブなときは、押されたように淡色で表示されます。ターゲットフィールドまたはリンクの左側の小さなアイコンも  のように表示されます。

Asset Managerにより、次の手順でソースファイルから1行ずつインポートされます。




- キーがまったく同じ値を持つデータベースレコードがある場合は、テキストファイルに含まれる情報に従ってレコードが変更されます。
- 同じキーのセットを持つレコードが複数ある場合は、プログラムは最初のレコードで停止し、他のレコードは無視されます。そのため、適切なキーを選択することが重要です。
- キーに一致するレコードがない場合は、データベースに新しいレコードが作成されます。

注意:

出力したデータを再インポートする場合は、照合更新キーとしてテーブルの「Id」フィールドを使わないようにしてください。実際には、対応するID番号は一定ではなく、変更されることがあります。資産の資産タグなど、値が「変化しない」キーを使います。

リンク先レコードの作成を設定する

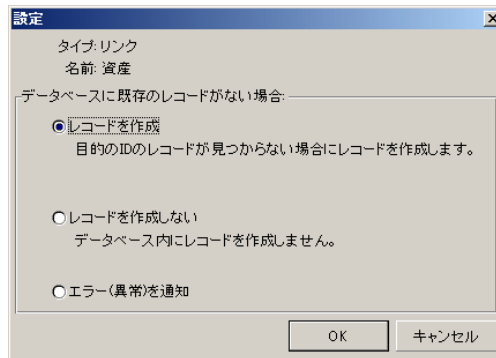
複数の異なるテーブルにインポートされるデータを含むファイルをインポートする場合（例えば、従業員と従業員のポートフォリオ品目を含むファイル）は、メインターゲットテーブル（この例では【部署と従業員】（amEmplDept）テーブル）を選択し、リンクを使って、データを他のテーブル（この例では【ポートフォリオ品目】（amPortfolio）テーブル）のどこにインポートするかを指定します。

Asset Managerでは、インポート時にレコードがない場合に、リンク先テーブルでレコードを作成するかしないかを事前に設定できます。この設定では、 アイコンを使います。このアイコンは、リンクでのみ使用できます（リンク先のテーブルのフィールドでは使用できません）。リンクは  および  アイコンで表されます。

設定画面を表示するには：

- 1 インポートするフィールドをリンク先テーブルのフィールドにマップします。
- 2 対応するリンクをクリックします。

3 📌 アイコンをクリックします。



指定した識別キーを使ってレコードが検出されると、レコードの情報が指定した設定に応じて変更されます。

レコードを作成

指定した識別キーを使ってレコードが検出されなくても、レコードを作成します。

レコードを作成しない

指定した識別キーを使ってレコードが検出されない場合は、レコードを作成しません。

エラー (異常) を通知

指定された識別キーを使ってレコードを検出できない場合、Asset Managerはエラーを発生します。

既にリンクしているレコードのみを検出 (📌)

リンクに📌アイコン (画鋏) を関連付けると、インポートモジュールは、既にメインレコードにリンクされているレコードだけを検出します。

例

従業員と従業員が使うポートフォリオ品目のリストをインポートしようとしています。[部署と従業員] テーブルがターゲットテーブルです。[ポートフォリオ品目] テーブルとのリンクに画鋏を関連付けます。ソースファイルで従業員に関連付けられている各ポートフォリオ品目について、従業員が既に使っているポートフォリオ品目だけが対象になります (従業員の詳細画面の [ポートフォリオ] タブページ)。

「Own」タイプのリンクの場合は、画鋏が自動的に関連付けられ、削除できません。「Own」タイプのリンクは、メインレコードが削除された場合に、リンクし

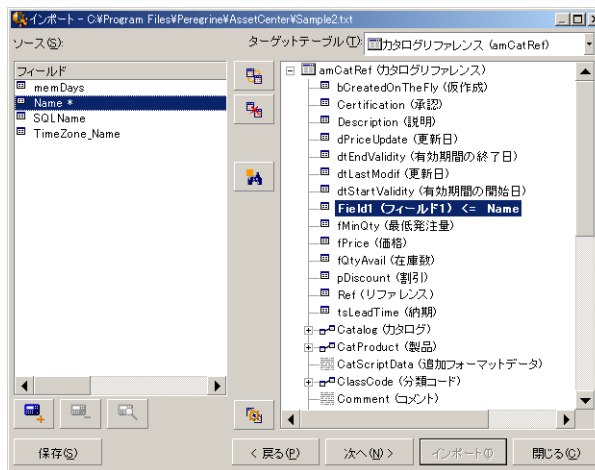
ているレコードも自動的に削除されるリンクです。このタイプのリンクの例としては、従業員と研修のリンクが挙げられます。従業員を削除すると、その従業員にリンクしている研修項目もすべて削除されます。

画鋲の影響は、リンクのタイプによって異なります。

- ターゲットテーブルがポートフォリオ品目のテーブルで、「ユーザ」リンクに画鋲を関連付けた場合、ポートフォリオ品目にリンクしているユーザだけが検索されます。特定のポートフォリオ品目のユーザは1人だけなので、キーでこのユーザを識別することなく、ポートフォリオ品目のユーザを変更または作成できます。また、特定のポートフォリオ品目の任意管理項目の値を変更する場合にも便利です。
- ターゲットテーブルが[部署と従業員]テーブルで、「ポートフォリオ品目」リンクに画鋲を関連付けた場合、従業員にリンクしているポートフォリオ品目だけが検索されます。この場合、ユーザのポートフォリオ品目を変更または作成するときに、適切な識別キーが必要ですが、画鋲を使わない場合ほど固有のキーでなくてもかまいません。

 **注意:**

Asset Managerでは、レコード間で3種類のリンクを使います。nリンク：例えば、ポートフォリオ品目は1つの設置場所だけにリンクできますが、設置場所は複数のポートフォリオ品目にリンクできます。1リンク：例えば、ポートフォリオ品目は1つのコメントだけにリンクでき、コメントも1つのポートフォリオ品目だけにリンクできます。n-nリンク：例えば、カタログは複数のサプライヤにリンクでき、サプライヤも複数のサプライヤにリンクできます。



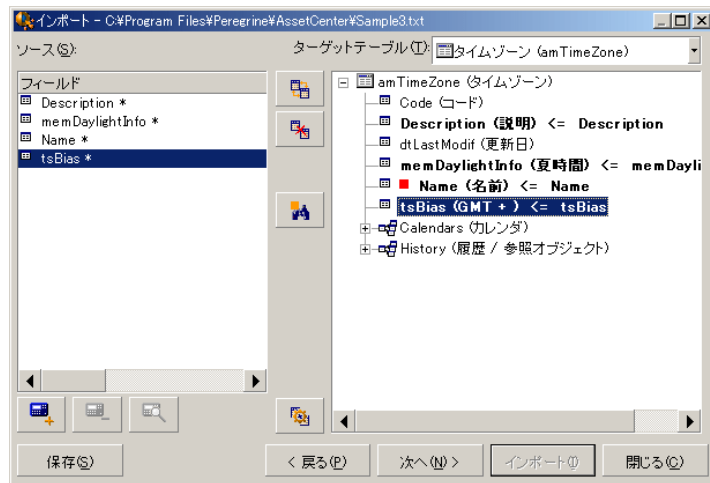
ターゲットテーブルのツリー構造で使われるシンボル

☐ オブジェクトがテーブル（大きな記号）またはフィールド（小さな記号）のどちらであるかを示します。

☐ 1個のレコードが親テーブルとリンクしているテーブルであることを示します。親テーブルのレコードの詳細画面で、リンク先のテーブルからレコードを1個だけ選択できます。このタイプのリンクは、「選択ウィンドウ」または「ドロップダウンリスト」を使って値を入力できるフィールドです。

☐ 複数のレコードが親テーブルとリンクしているテーブルであることを示します。親テーブルのレコードの詳細画面で、リンク先のテーブルから複数のレコードを選択できます。このタイプのリンクは、親テーブルの詳細画面のタブページに表示されるレコードのリストです。

☐ この記号は「画鋲」と呼ばれます。インポートするフィールドのマップ先のフィールドを含んでいるテーブルへのリンクだけに関連付けられます。画鋲を設定すると、レコードの中から「ピンで留められた」レコードにリンクするものだけが検索されます。画鋲の有無は、📌 アイコンを使って表示するオプションによって決まります。



手順7：ソースファイルに特殊フィールドを追加する

Asset Managerでは、ソースファイルに追加フィールドを追加できます。これらのフィールドは保存されず、メモリに格納されるだけです。

これらの追加フィールドを追加、削除、または表示するには、📌、🗑️ および 🔍 アイコンを使います。

名前

新しいフィールドに名前を付けます。

フィールドタイプ

新しいフィールドの計算方法を入力します。画面の内容は、選択したタイプによって変わります。

連結

このモードを使うと、ソースファイルの複数のフィールドを組み合わせることができます。連結するフィールドを1つずつ選択します。フィールド同士を二重引用符 (") で囲んだ任意の文字で連結できます。

例：フィールド1"and"フィールド2

固定

このモードを使うと、テキストフィールドの一部を抽出できます。

- 1 ソースフィールド（「メインフィールド」と呼びます）を選択します。
- 2 **【含めない文字数】**を入力します。これらの文字が飛ばされます。
- 3 **【使用する文字数】**を入力します。**【含めない文字】**を飛ばした後でこの数の文字を保持します。
- 4 **【フィールドの最後から抽出開始】** オプションは、フィールドの末尾から**【含めない文字数】**を飛ばし、そこから前方向に数えた**【使用する文字数】**を保持するときに使います。

例：

- 1 **【含めない文字数】**：3
- 2 **【使用する文字数】**：5
- 3 ソースファイルのフィールド値：「REFIMP05A18500」
- 4 データベースにインポートされる値：**【フィールドの最後から抽出開始】** チェックボックスをオフにした場合は「IMP05」、オンにした場合は「05A18」。

分割

このモードを使うと、ソースファイルのフィールドの一部を抽出できます。

- 1 ソースフィールド（「メインフィールド」と呼びます）を選択します。
- 2 メインフィールドの値内で使う**【区切り文字】**を指定します。
- 3 **【含めない区切り文字数】**を入力します。指定した数の区切り文字に続くすべてのデータが保持されます。
- 4 **【含める区切り文字数】**を入力します。保持するテキストの始まりから、最後に含める区切り文字の次の区切り文字までの情報が保持されます。

- 5 フィールドの末尾から [含めない区切り文字数] と [含める区切り文字数] オプションを適用する場合は、[フィールドの最後から抽出開始] チェックボックスをオンにします。

例：

- 1 [区切り文字] : /
- 2 [含めない区切り文字数] : 2
- 3 [含める区切り文字数] : 3
- 4 ソースファイルのフィールド値 : 「1/2/3/4/5/6/7/8/9」
- 5 データベースにインポートされる値 : [フィールドの最後から抽出開始] チェックボックスをオフにした場合は「3/4/5/6」、オンにした場合は「4/5/6/7」。

固定値

このモードを使うと、次の組合せを含めることができます。

- 二重引用符 (") で囲まれた文字列
- 変数。これらは、AmLoginName()、AmDate()、AmCounter()などデフォルトのフィールド値で使われている関数の結果発生する特定の変数です。

ツリー構造

このモードを使うと、ソースファイルの1つのフィールド値からツリー構造を構築できます。

- 1 ソースフィールド（「メインフィールド」と呼びます）を選択します。
- 2 フィールド内の値を分離する区切り文字を指定します。

ソースフィールドが複数の値に分割されます。分割後の値の数は、区切り文字で区切られた文字列の個数に等しくなります。その後、各値のレコードが作成され、階層構造で編成されます。

例：

- 1 「名前」というタイトルの列を含むテキストファイルを作成します。ファイル名の行の1つの値は、「/フランス支社/営業本部/マーケティング部」です。
- 2 「ツリー構造」タイプの計算式フィールドを作成するようにインポートモジュールを設定します（区切り文字はスラッシュ (/)）。名前は「FormulaField」です。「固定値」タイプの計算式フィールド（値 = 1）を作成し、[部署] フィールドにマップします（従業員ではなく部署を作成するため）。
- 3 「FormulaField」を従業員テーブルの[部署名/姓]（SQL名 : Name）フィールドにマップします。
- 4 ファイルのインポートを起動します。
- 5 結果：階層的にリンクされた3つの部署、「フランス支店」、「営業本部」、「マーケティング部署」が作成されます。

ファイル

このモードを使うと、ファイルをデータベースにインポートできます。画像や長いテキストのブロックのインポートに適しています。

ファイルは、次のデータ型のフィールドだけにインポートできます。

- メモフィールド
- バイナリフィールド


次のファイル形式をサポートしています。

- ANSIテキスト
- 画像（Asset Managerがサポートしているすべての画像形式をインポートできます。）

フィールド計算式では、インポートするファイルの完全パス名（パス、名前、拡張子）を含むソースフィールドを指定します。デフォルトでは、現在のフォルダがパスとして使われます。

スクリプト

このモードを使うと、Basicスクリプトを使って値を計算できます。このスクリプトはインポートされたソースフィールドを参照できます。

計算スクリプトを作成するには、コードを直接入力するか、または  ボタンをクリックして式ビルダを使います。

スクリプトでは、データベースのフィールドは参照できません。

テスト

注意:

このフィールドは、フィールドの計算式タイプが「固定」または「分割」に設定されている場合のみ表示されます

希望のフィールド値を入力します。

結果

注意:

このフィールドは、フィールドの計算式タイプが「固定」または「分割」に設定されている場合のみ表示されます

フィールドに、テストデータでシミュレートされたインポート値が表示されません。

手順8：特殊なケース

部署と従業員をインポートする

【部署と従業員】（amEmplDept）テーブルからレコードをインポートする場合は、インポートするレコードが部署か従業員かを指定しなければならないことがあります。

これは、【部署】（SQL名：bDepartment）フィールドで指定できます。部署の場合は「1」、従業員の場合は「0」に設定します。デフォルトでは、値は「0」と見なされます。

値が「1」の「固定値」タイプの計算フィールドを作成し、インポートする項目が部署の場合は、この計算式を【部署】フィールドにリンクします。

注意:

このレコードが、【部署と従業員】テーブルで子レコードを持っている場合、このレコードは部署と見なされます。従業員には子レコードは存在しません。

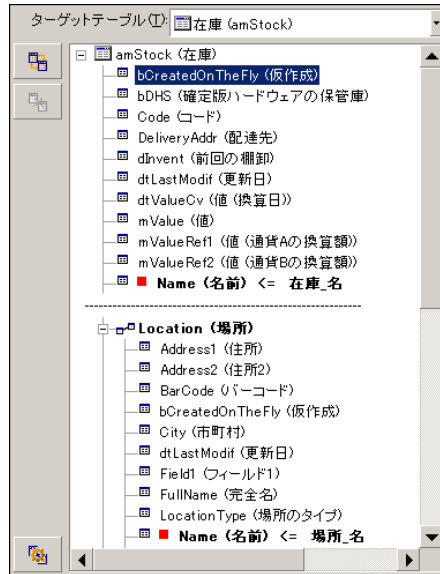
ドキュメントをインポートする

ドキュメントをインポートする場合は、インポートするフィールドを、【ドキュメント】テーブルの【テーブル】（SQL名：DocObjTable）フィールドで指定する必要があります。【テーブル】フィールドは、ドキュメントのリンク先のテーブルのSQL名を示します。

手順9：キーの使用例

Asset Managerで、選択したキーがどのように解釈されるかを説明します。

例1：リンクしているフィールドをメインテーブルのキーとして使う



この例では、在庫は2つのメインキーで識別されます。

- **Stock.Name** : 「 **Name (名前) <= 在庫_名** 」 キーは照合更新キーとして宣言されており、メインテーブルの要素であるため。
- **Location.Name** : 「 **■ Name (名前) <= 場所_名** 」 リンクフィールドは、[場所] テーブルで照合更新キーとして宣言されており、「 **Location (場所)** 」 リンクは照合更新キーとして宣言されているため。

同じ例で、場所はメインキーによって識別されます。

- **Location.Name** : 「 **■ Name (名前) <= 場所_名** 」 フィールドは [場所] テーブルで照合更新キーとして宣言されているため。

例2：メインテーブルではキーでないフィールドを、リンク先テーブルでキーとして定義する



この例では、在庫は1つのキーで識別されます。

- Stock.Name：「Name (名前) <= 在庫_名」フィールドはキーとして宣言されており、メインテーブルの要素であるため。
- 「Location (場所)」リンクは照合更新キーではない。

同じ例で、場所は1つのキーによって識別されます。


- Location.Name：「Name (名前) <= 場所_名」キーはメインキーとして宣言されているため。

結論

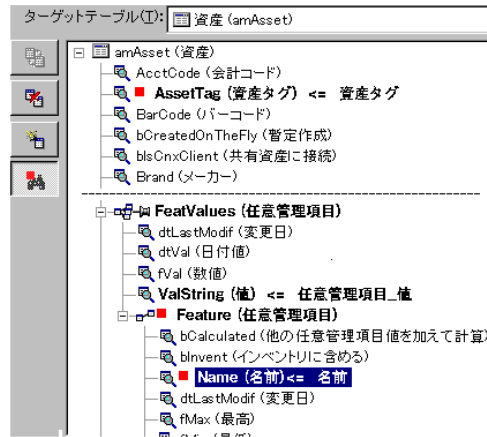
- メインテーブル用にキーを定義できるだけでなく、メインテーブルにリンクしているテーブルにも他の独立したキーを定義できます。これにより、1つのテキストファイルから複数のテーブルにデータをインポートできます。
- リンク先のテーブルのフィールドを、メインテーブルのキーの1つとして宣言するには、リンク先のテーブルとリンクを識別キーとして宣言する必要があります。リンクだけを選択すると、キーはリンク先テーブルのキーとしてのみ使われます。





例3：データベースレコードの任意管理項目の値を更新するキー

レコードの任意管理項目の値を更新するには、データベースで（レコード、任意管理項目）の組合せを見つけ、新しい値を割り当てる必要があります。

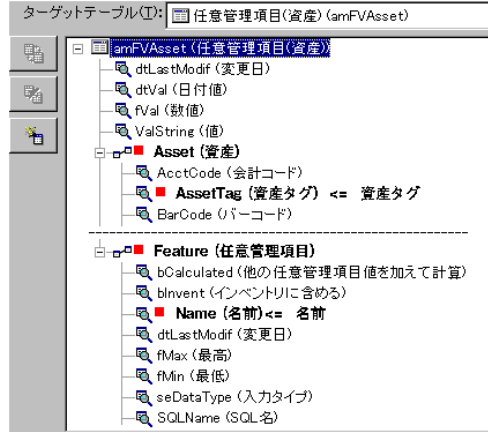
Asset Managerは、リンクの  アイコンを使ってリンクに定義したオプションを参照し、リンクするレコードを作成する必要があるかどうかを判断します。次の2つの方法のどちらかを使います。

第1の方法（資産テーブルからの例）



- 1 ■ **AssetTag (資産タグ)** のキーにより、変更する資産が識別されます。
- 2  **FeatValues (任意管理項目)** リンク上の  アイコンは、この資産の任意管理項目だけを見つけることを示します。
- 3  ■ **Feature (任意管理項目)** のキーは、（任意管理項目、値）の組合せが、任意管理項目によって識別されることを示します。
- 4  ■ **Name (名前) <= 名前** のキーは、任意管理項目が名前で識別されることを示します。
- 5 新しい値が [**ValString (値) <= 任意管理項目_値**] フィールドに表示されます。

第2の方法（資産の任意管理項目値テーブルからの例）

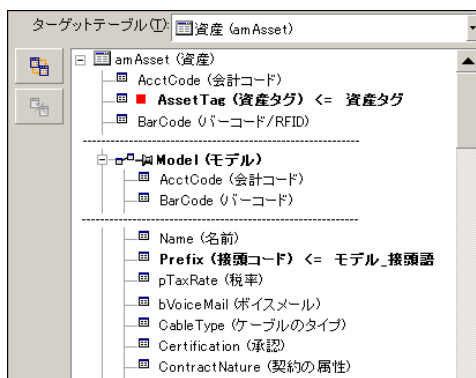


- 1 「 ■ Feature (任意管理項目) 」 と 「 ■ Asset (資産) 」 の組合せにより、資産に関連付けられている任意管理項目が識別されます。
- 2 「 ■ Name (名前) 」 のキーは、「 ■ Feature (任意管理項目) 」 リンクのキーフィールドを示します。
- 3 「 ■ AssetTag (資産タグ) 」 のキーは、「 ■ Asset (資産) 」 リンクのキーフィールドを示します。
- 4 新しい値が [ValString (値) <= 任意管理項目_値] フィールドに表示されます。

例4：照合更新キーのないリンクレコードの内容を変更する

例

特定の資産にリンクされているモデルの接頭語を変更するとします。インポートファイルには、モデルを識別するキーはありません。このためモデルがその資産にリンクされていることしか分かりません。



- 1 ■ AssetTag (資産タグ) <= 資産タグ のキーは資産を識別します。
- 2 Model (モデル) 上の Model (モデル) で、この資産に既にリンクされているモデルだけを検索するように指定します。
- 3 Prefix (接頭コード) <= モデル_接頭語 により、新しい値が、モデルの [接頭コード] (SQL名: Prefix) フィールドに割り当てられます。

注意:

最適なパフォーマンスを得るため、テーブルのインデックスを構成するキーからキーを選択するようにしてください（注意：特定のインデックスは複数のフィールドで構成されています）。

手順10：転送を設定する

1つのテキストファイルをインポートする場合

手順5でソースフィールドをデータベースのフィールドに割り当てたら、[次へ] ボタンをクリックして、転送設定ウィンドウを表示します。

複数のテキストファイルまたはODBCデータベースをインポートする場合

インポートモジュールのメイン画面から [編集/オプション] メニューを選択します。

[エラー処理] フレーム

次の中からエラー処理の方法を選択します。

エラー発生時にインポートを中止

エラーが発生すると即座にインポート処理を停止します。

インポートした行（1行）毎に適用

インポートモジュールは、インポートする行ごとにエラー発生時のデータの処理を決定します。行（メインテーブルまたはリンク先のテーブル）でエラーが検出された場合は、その1行全体のインポートが中止されます。データベースは、この行をインポートする前の状態に復元します。

複数の行単位で適用

インポートするデータは行のグループ単位で処理されます。グループの行数はユーザが指定します。グループ（メインテーブルまたはリンク先のテーブル内の任意のグループ行）でエラーが発生した場合、そのグループに属するすべての行のインポートが中止されます。データベースは、このグループ行をインポートする前の状態に復元します。

[ログファイル] フレーム

ログファイルに記録する処理を選択します。

- エラー
- 追加および更新

ログファイルの名前とパスを入力します。ログファイルが存在しない場合は **Asset Manager** が作成します。希望の拡張子を追加します。「.log」を使うことをお勧めします。



警告:

インポートモジュールでは、フォルダは作成できません。

ログファイルには、次の情報も記録されます。

- ジョブが発行された時間
- ジョブの説明
- 検出されたエラー

ログファイルは、インポートを実行するたびに上書きされます。

手順11：データ転送

前の手順に従って必要な情報が揃ったら、[インポート] ボタンをクリックしてデータ転送を開始します。

レコードの処理

- **Asset Manager**により、ソースファイルに表示された順番で1行ずつデータがインポートされます。

- 1つの行から複数の異なるテーブルにデータをインポートできます。
- インポート行の一部をインポートできない場合は、作成可能な部分だけが作成されます。
- **Asset Manager**データベースで、識別キーがソースレコードとまったく同じ値を持つレコードが検索されます。このようなレコードが見つかった場合は、テキストファイルに含まれる情報に従ってレコードが変更されます。
- 照合更新キーに一致するレコードが見つからない場合は、データベースに新しいレコードが作成されます。
- 照合更新キーを定義しない場合は、インポートする値が基本フィールドの固有属性と合っていれば、レコードがインポートされます。キーがない場合は、レコードを更新することはできません。
- インポートモジュールは、メイン項目とリンク先の項目に対して同じように動作します。

フィールドの処理

- インポートモジュールは、フィールドが必須かどうかを認識しません。自分自身でインポートするデータに必須フィールドが含まれていることを確認する必要があります。
- 「システムリストデータ」内にフィールドの不明な値が検出されると、その行はインポートされません。
- 標準のリストデータにまだ含まれていない値が検出された場合は、リストが「オープン」（変更可能）であればその行はインポートされ、新しい値がリストデータに追加されます。リストデータが「クローズド」（変更不可能）の場合は、その行はインポートされません。
- フィールド値が最大長を超えている場合は、値の末尾が切り捨てられます。
- テキストファイルに値のないフィールドがあると、対応するデータベースフィールドの既存の値が消去されます。
- 1行のデータのインポートによって新しいレコードが作成される場合、テキストファイルまたはソーステーブルで列として表れないフィールドにはデフォルト値が挿入されます。列が存在しても値が指定されていない場合は、デフォルト値ではなく空のフィールドが挿入されます。

インポートスクリプトを保存および実行する

インポートスクリプトは、特定の名前で保存されるインポート設定の集りです。インポートスクリプトを作成すると、すべてのパラメータを再定義することなく、同様のインポート条件を作成できるため、時間を節約できます。

スクリプトは次の場合に便利です。

- 希望どおりに動作するまで、同じインポート操作を繰り返し実行する場合（例えば、実行するたびにソースフィールドを変更できます）。

- 定期的にデータベースを更新する場合（例えば、人事部署からのファイルに基づいて【部署と従業員】テーブルを更新できます）。

Asset Managerでスクリプトを保存して、後で実行することができます。

スクリプトを保存する方法

1つのテキストファイルをインポートする場合

- 1 【ファイル／インポート】メニューを使ってテキストファイルを選択します。
- 2 インポート設定（データファイルの名前と場所、構造など）を定義します。
- 3 【保存】ボタンをクリックすると、これらの条件をスクリプトとしていつでも保存できます。

データベースまたは複数のテキストファイルをインポートする場合

- 1 【ファイル／インポート】メニューを使ってデータベースを選択します。
- 2 インポート設定（データベースの場所、フィールド間のマッピングなど）を定義します。
- 3 【ファイル／保存】または【ファイル／名前を付けて保存】メニューを使うと、これらの設定をスクリプトファイルとしていつでも保存できます。

スクリプトを変更する方法

1つのテキストファイルをインポートする場合

- 1 【ファイル／インポート】メニューを選択します。
- 2 テキストファイルをインポートするオプションを選択します。
- 3 【データファイルを開く】ダイアログボックスの【ファイルの種類】フィールドで「インポートスクリプト(*.scr)」を選択します。
- 4 スクリプトを開きます。
- 5 インポート設定（データファイルの名前と場所、構造など）を変更します。
- 6 【保存】ボタンをクリックすると、これらの設定をスクリプトとしていつでも保存できます。

データベースまたは複数のテキストファイルをインポートする場合

- 1 【ファイル／インポート】メニューを使ってデータベースをインポートします。
- 2 【ODBC】タブページまたは【テキスト】タブページに入力します。【開く】をクリックします。
- 3 次に、【ファイル／スクリプトを開く】メニューを使ってスクリプトを開きます。

- 4 インポート設定を変更します（データベースの名前と場所、構造など）。
- 5 **【ファイル／保存】** または **【ファイル／名前を付けて保存】** メニューを使うと、これらの設定をスクリプトファイルとしていつでも保存できます。

インポート設定の定義時に新しいスクリプトを作成する方法

1つのテキストファイルをインポートする


【閉じる】 ボタンをクリックします。新しいスクリプトの作成手順に従います。

データベースまたは複数のテキストファイルをインポートする場合

【ファイル／新規スクリプト】 メニューを使います。現在の設定を破棄する前にスクリプトとして保存することを確認するメッセージが表示されます。

スクリプトを実行する方法

既存のスクリプトを実行するには：

- 1 **【ファイル／インポート】** メニューを選択します。
- 2  アイコンをクリックします。
- 3 スクリプトファイルのフルネームを指定します。
- 4 ODBCソースに関連付けられたパスワードを指定します。
- 5 インポート処理が開始されます。

スクリプトを変更しようとする時と同じ手順でスクリプトを開いて、インポートダイアログボックスからインポート処理を実行することもできます。

コマンドプロンプトからインポートスクリプトを実行する

動作

DOSプログラムを“オンライン”で実行するには、Windowsインポートモジュールを使って事前にスクリプトを作成しておく必要があります。

インポートアプリケーション「**amimpl.exe**」（Asset Managerのインストール先フォルダの「**bin**」フォルダ内）を使って、インポートコマンドを手動、または自動（バッチファイルを介して）で実行することができます。

シンタックス

amimpl [-verbose] [-?|h|H] -src:<cnx> [-srcpass:<password>] -dst:<cnx> [-dstlogin:<login>] [-dstpass:<password>] [-log:<file>]

-verbose : インポート中にメッセージを表示します。デフォルトでは有効になっています。

-?, -hまたは-H : ヘルプメッセージを表示します。

-src : 状況に応じて、このパラメータで次の情報を指定します。

- 実行するインポートスクリプトのパスと名前
- 完全にインポートするAsset Managerデータベースの接続名（ [ファイル/接続の管理] メニュー / [名前] フィールドに指定されている名前）
- 接続のないAsset Managerデータベースの名前
[<エンジン名>;<データベースの場所>;<ユーザ>;<パスワード>]
このシンタックスの<>で囲まれたフィールドへの入力内容を下に示します。

	Oracle	MS SQL Server
エンジン名	Oracle	ODBC
データベースの場所	サーバ名	データソース名
ユーザ	アカウント名	MS SQL Serverユーザ名
パスワード	アカウントパスワード	MS SQL Serverユーザパスワード

-srcpass : インポートするソースデータベースに関連付けられているパスワード。Asset Managerデータベースの場合は、「Admin」アカウントのパスワードです。

-dst : データのインポート先のAsset Managerデータベースの接続名（ [ファイル/接続の管理] メニュー / [名前] フィールドに指定されている名前）

-dstlogin : インポートされたデータを受け取るAsset Managerアカウントのログイン名（「Admin」アカウントまたは管理者権限を持つAsset Managerユーザ）

-dstpass : 「dstlogin」に関連付けられているパスワード

-log : インポートログファイルの完全パス名



注意:

山形括弧 (<>) 内に指定する文字列にスペースを含める場合は、その文字列全体を一重引用符 (') で囲む必要があります。

例 :

```
amimpl32 -verbose -src:employee.scr -srcpass:PassWord -dst:MainDBase -dstlogin:Gerald -dstpass:PassWord -log:'My Log File.txt'
```


12 データベースへのアクセスコントロール

この章では、Asset Managerデータベースへのユーザアクセスを管理する方法について説明します。

 **注意:**

アクセス規則を管理できるのは、データベースの管理者だけです。

アクセス権限の管理の重要性と概要

Asset Managerは、複数のユーザが同時に使えるプログラムです。つまり、複数のユーザで1つのデータベースを共有します。

Asset Managerでデータベースへのアクセス権限を管理するには、次の手順に従います。

- 1 まず、アクセスできるデータとアクセス条件をユーザごとに定義する必要があります。
- 2 次に、HP AutoPass License Management Toolでインストールされた Asset Managerライセンスキーに従って、データベースへの接続数を管理する必要があります。

アクセス条件を定義する

全ユーザがデータベースで同じタスクを実行し、同じ変更を行う必要は必ずしもありません。これは、例えば、従業員Aは資産を作成する必要がある一方、従業員Bは在庫にアクセスする必要があり、従業員Cは作業指示伝票を確認する必要があるなど、会社での役割と会社の手順に依存するものです。

従業員にAsset Managerへのアクセス権限を付与する手順を次に示します。

- 従業員を、[部署と従業員] テーブルに登録します。
- 従業員がAsset Managerに接続するには、「ログイン」が必要です。
- 管理者は、従業員に「承認されたユーザ役割」および「接続に使用する役割」を割り当てるか、従業員に管理者権限を与える必要があります。

データのセキュリティと機密性

Asset Manager管理者は、データベースへのアクセスを次のように3段階で管理することができます。これにより、データのセキュリティを確実にし、情報の漏洩、変更、および破壊を防止することができます。

- ユーザのネットワークへのアクセス権限の定義
- Asset Managerユーザ役割とプロファイルの定義
- データベースバックアップの方法と時点の定義

アクセス権限の定義

このセクションでは、データベースへのアクセスを制御するアクセス権限の概念について説明します。

- ユーザ役割の定義 [P. 165]
- ユーザプロファイルの定義 [P. 165]
- ユーザ権限の定義 [P. 165]
- アクセス制限の定義 [P. 166]
- 機能権限の定義 [P. 166]

ユーザ役割は複数のユーザプロファイルに関連付けることができます。また、ユーザプロファイルを、複数のユーザ役割の一部にすることもできます。

ユーザプロファイルは、ユーザ権限、機能権限とアクセス制限からなります。ユーザ権限、機能権限、またはアクセス制限を複数のユーザプロファイルで使用できます。

ユーザ役割の定義

ユーザ役割は、Asset Managerでユーザが引き受ける役割を表します。例えば、次が挙げられます。

- 全ポートフォリオ品目の棚卸管理に関するすべての面を管理する必要がある従業員のための「棚卸マネージャ」。
- 調達手順のすべての面を管理する必要がある従業員のための「購入マネージャ」役割。

ユーザは複数のユーザ役割を担うことができます。

ユーザがAsset Managerデータベースに接続した場合、管理者は、そのユーザに関連して可能性があるすべて役割の中から1つ選択し、そのユーザに割り当てる必要があります。この役割が、「接続に使用する役割」になります。

重要項目:

管理者権限がない任意のユーザがAsset Managerデータベースに接続するには、「接続に使用する役割」が必要です。必要な場合、ユーザはその役割を変更できます。▶ [ユーザの接続に使用する役割を選択する](#) [P. 176]

ユーザプロファイルの定義

ユーザプロファイルは、次から構成されます。

- Asset Managerデータベースのテーブル、フィールド、リンクへのユーザ権限とアクセス制限。
- Asset Managerの機能ドメインへのアクセス権限（機能権限）。

例えば、次のようにアクセスを制御できます。

- 「ヘルプデスク技術者」のプロファイルでは、アクセスがヘルプデスクに関連するテーブルに制限されています。
- 「会計」のプロファイルでは、コストセンタ、予算、経費明細にしかアクセスできません。

ユーザプロファイルは、ユーザに直接割り当てることができません。プロファイルは、そのプロファイルが関連付けられたユーザ役割を介してのみユーザに割り当てられます。

ユーザ権限の定義

ユーザ権限は、ユーザプロファイルのコンポーネントの1つです。これには、Asset Managerデータベースのテーブル、フィールド、レコードに対して許可される権限があります。管理者は次の権限を割り当てることができます。

- テーブルに対して：作成、削除
- テーブルのフィールドとリンクに対して：データの読取り、挿入、変更

アクセス制限の定義

アクセス制限は、ユーザプロファイルの1部です。テーブルのレコードのフィルタに相当します。例えば、技術者は自分の部署の資産にしかアクセスできないようにできます。アクセス制限で読取り／書込み（追加または変更）に関する制限を行うことができます。

機能権限の定義

機能権限は、ユーザプロファイルの構成要素の1つです。Asset Managerで使用可能な機能ドメインに基づきます（調達、ファイナンス、ポートフォリオなど）。機能権限は、異なる機能ドメインと様々な画面へのアクセス、およびこれらのドメインで使用可能なウィザードを提供します。管理者は、ユーザに対してそのユーザの職務権限およびAsset Managerで実行する必要があるタスクに応じて権限を割り当てることができます。

アクセス条件を定義する

このセクションでは、アクセス条件を定義する方法について説明します。内容は次のとおりです。

- ユーザ役割を定義する [P. 168]
- ユーザプロファイルを定義する [P. 168]
- ユーザ権限を定義する [P. 168]
- アクセス制限を定義する [P. 171]
- 機能権限を定義する [P. 173]

アクセス条件の定義が完了すると、Asset Managerユーザに関連付けることができます。

前提条件：アクセス管理専門分野データのインポート

アクセス管理専門分野データには、独自レコードの作成に使用できる一般的なユーザ役割とプロファイルのセット（ユーザ権限、アクセス制限、機能権限とともに）を含みます。

データベース作成時の専門分野データのインポート

データベースの作成時に専門分野データを挿入するには、「▶ [Asset Manager Application Designer](#)を使ってデータベース構造を作成する [P. 57]」を参照してください。

【インポートするデータ】 ページ上で、【管理 - ユーザ役割プロファイルの権限と制限】を選択します。

既存のデータベースへの専門分野データのインポート

既存のデータベースへの専門分野データを挿入するには：

- 1 Asset Manager Application Designerを起動します。
- 2 [ファイル/開く] メニューを選択します。
- 3 [データベース記述ファイルを開く (新規データベースの作成)] オプションを選択します。
- 4 「gbbase.xml」ファイルを選択します。このファイルは、Asset Managerのインストールフォルダの「config」サブフォルダにあります。
- 5 データベース作成ウィザードを開始します（[アクション/データベースの作成] メニュー）。
- 6 ウィザードの各ページで必要な情報を入力します（[次へ] および [前へ] ボタンでウィザードページを移動します）。

[SQLスクリプトの生成/データベースの作成] ページ：

フィールド	値
データベース	データのインポート先にするデータベースへの接続を選択します。
作成	[専門分野データをインポート] オプションを選択します。
高度な作成オプションを使用	このオプションは選択しないでください。

[作成パラメータ] ページ：

フィールド	値
パスワード	Asset Manager管理者のパスワードを入力します。
	注意: Asset Managerデータベース管理者は、[部署名/姓] (Name) フィールドが「Admin」に設定されている [部署と従業員] (amEmplDept) テーブルのレコードです。

[インポートするデータ] ページ：

フィールド	値
使用可能データ	[管理 - ユーザ役割プロファイルの権限と制限] オプションを選択します。
エラー発生時にインポートを中止	問題が発生した場合にインポートを中止するには、このオプションを選択します。

フィールド	値
ログファイル	エラーや警告などすべてのインポート操作を記録するファイルの完全名。

7 ウィザードを実行します（[完了] ボタン）。


ユーザ役割を定義する

ユーザ役割の定義は、[管理/権限/ユーザ役割] ナビゲーションメニューから行います。各ユーザ役割は、それに関連付けられる様々なユーザプロファイルをグループ化します。

例えば、「発注管理」と「サプライヤ管理」のプロファイルは、「調達マネージャ」役割と関連付けることができます。

ユーザプロファイルを定義する

ユーザプロファイルは、次により定義されます。

- [管理/権限/ユーザプロファイル] ナビゲーションメニューを使う。
- または、ユーザ役割と関連付けられるユーザプロファイルのリストの  (Windowsクライアント) または [追加] (Webクライアント) をクリックする (ユーザ役割詳細の [全般] タブ)。


ユーザプロファイルは、次により構成されます。

- **ユーザ権限**。データベースのテーブル、フィールドまたはリンクの情報の読取り、書込み、作成、削除を行う権限を定義します。
- **アクセス制限**。特定のテーブル内のレコードに対する読取り/書込み条件を定義します。
- **機能権限**。Asset Manager機能ドメインに関する使用可能な画面およびウィザードへのアクセスを定義します。

ユーザプロファイルは、社内での職務とその権限に対応するものと見なされます。

ユーザ権限を定義する

次のいずれかの方法でユーザ権限を管理する画面を表示します。

- [管理/権限/ユーザ権限] メニューを使う。
- または、ユーザプロファイルと関連付けられるユーザ権限のリスト右側の  (Windowsクライアント) または [追加] (Webクライアント) をクリックする (ユーザ役割詳細の [役割、権限および制限] タブ)。

ユーザ権限は、ユーザがどのようにデータベースのテーブル、フィールドおよびリンクにアクセスするかを定義します。

設定方法としては、テーブルとそのテーブルにリンクしているフィールドに対するアクセス権限を定義したユーザ権限を、テーブルごとに作成することをお勧めします。1つのデータベーステーブルに対して複数のアクセスレベルを設定できます。

例えば、次のようなレベルでユーザ権限を設定できます。

- 基本会計
- 上級会計
- 従業員の役職に応じたアクセス
- メンテナンス
- その他

これらのユーザ権限を組み合わせることでユーザプロファイルを作成できます。

- 会計担当者
- メンテナンス技術者
- 研修生
- その他

ユーザ権限を編集する

注意事項

注意:

指定内容に応じて表示される専用のタブページを表示するには、少なくともそのタブページの表示に必要なフィールドの読取り権限が必要です。例えば、契約の詳細画面の [賃貸料] と [ローン] タブページは、同じ画面の [全般] タブページに表示される [支払属性] (SQL名: sePayType) フィールドの値に応じて表示されます。このため、このフィールドの読取り権限がない場合は、これらのタブページを表示できません。

Windowsクライアント

テーブルのツリー構造を使用してデータベース構造を示します。各テーブルのツリーは、テーブルに固有のフィールドと任意管理項目のリスト、およびテーブルにリンクしているフィールドとテーブルで構成されています。テーブル、フィールド、任意管理項目に対して、それぞれ固有のユーザ権限を割り当てます。

ユーザ権限詳細画面は、以下のように構成されます。

- ユーザ権限は、オブジェクトの隣の列に表示されます。

オブジェクト	権限 (RIU/CD)
+ 経費明細 (amExpenseLine)	rIU/CD
+ 統計 (amStatistic)	---/--
+ 値調整フィールド (amFieldAdjust)	rIU/CD
- 製品 (amCatProduct)	rIU/CD
+ フィールド	RIU
+ リンク	RIU
- (任意管理項目)	RIU
+ 乗数 (fv_OverheadFactor)	RIU
+ 時間の単位 (fv_UnitDuration)	RIU
+ (特殊フィールド)	r--
+ 調整タイプ (amFieldAdjustTemp)	rIU/CD
+ 調整項目 (amAdjustment)	rIU/CD
+ 請求明細 (amInvoiceLine)	rIU/CD
+ 請求書 (amInvoice)	rIU/CD
+ 論理ドライブ (amLogicalDrive)	RIU/CD
+ 資格 (amRelModelEntit)	RIU/CD

- **[権限のあるテーブルのみ表示]** フィルタを使うと、ユーザ権限が定義済であるテーブルのみが表示されます。
- ノード（テーブル、フィールド、リンク、任意管理項目など）を選択すると、Asset Managerは自動的にツリー内のブランチをすべて選択します。これで、ノード全体のユーザ権限の編集が可能になります。親ノードに表示される権限の略語の意味は次のとおりです。
 - 小文字の「r」：そのノード内の特定の項目に読取り権限があります。
 - 大文字の「R」：そのノード内のすべての項目に読取り権限があります。
 - 小文字の「i」：そのノード内の特定の項目に追加権限があります。
 - 大文字の「I」：そのノード内のすべての項目に追加権限があります。
 - 小文字の「u」：そのノード内の特定の項目に変更権限があります。
 - 大文字の「U」：そのノード内のすべての項目に変更権限があります。
- リストから複数の項目を同時に選択すると、一度にユーザ権限を編集できます。複数の項目を選択するには、[Shift] キーまたは [Ctrl] キーを押しながら項目を選択します。

Webクライアント

ユーザ権限詳細画面は、以下のように構成されます。


- テーブルのリスト
 - **[Show entries (エントリの表示)]** フィールド：表示ごとのテーブル数を選択します。
 - **[検索]** フィールド：このフィールドに入力した文字列が、テーブルの**テーブル名**または**SQL名**に含まれるテーブルが表示されます。
 - テーブルリスト：アクセス権を編集しようとするテーブルの**[テーブル名]**または**[SQL名]**をクリックすると、そのテーブルのコンポーネントが、この画面下部のカテゴリビューに表示されます。
- テーブルコンポーネントのカテゴリリスト

各テーブルのコンポーネントは、**【フィールド】**、**【リンク】**、**【特殊フィールド】**、および**【任意管理項目】**（存在する場合）ごとにまとめられます。テーブル、またはテーブルの特定のフィールド、リンク、および任意管理項目に権限を割り当てることができます。

- 親ノード（テーブル、フィールド、リンク、任意管理項目など）を選択すると、Asset Managerはそのすべての子ノードを自動的に選択します。これで、ノード全体のユーザ権限の編集が可能になります。親ノードの場合、小文字の"**r**"/"**i**"/"**u**"は、その親ノードの下の一部のアイテムが読取り/作成/更新権限を持つことを示します。大文字の"**R**"/"**I**"/"**U**"は、親ノード内のすべてのアイテムが読取り/作成/更新権限を持つことを示します。
- **【Cache Update (キャッシュの更新)】** ボタン：あるテーブルの権限を選択してから**【Cache Update (キャッシュの更新)】** をクリックすると、その選択内容が保存され、引き続き別のテーブルの権限を選択できます。

アクセス制限を定義する

次のいずれかの方法でアクセス制限を管理する画面を表示します。

- **【管理／権限／アクセス制限】** メニューを使う。
- または、ユーザプロファイルと関連付けられるアクセス制限のリストの  (Windowsクライアント) または **【追加】** (Webクライアント) をクリックする (ユーザ役割詳細の **【役割、権限および制限】** タブ)。

アクセス制限の詳細画面では、データベースのテーブルでユーザがどのレコードを参照できるかを定義します。

クエリと同様の基準を使って職務権限によるアクセスや変更権限を制限できます。制限の基準として、例えば次の情報を使います。

- モデルまたはブランド
- 部署または場所
- 保険契約

定義したアクセス制限とユーザ権限を組み合わせると、次のようなユーザプロファイルを作成できます。

- 大阪のメンテナンス技術者
- 購入責任者
- その他



注意:

以下のテーブルにアクセス制限を定義することはできません。これらのテーブルは、クライアントが開かれたときにキャッシュにロードされます。

- amAction
- amCalcField
- amCbkrule
- amCurRate
- amCurrency
- amDateAlarm
- amFeatMemo
- amFeatParam
- amFeature
- amFeatureClass
- amFieldAdjustTempl
- amFuncDomain
- amOption
- amReconcProposal
- amReport
- amScriptLibrary
- amSysConfig
- amTaxFormula
- amUnit
- amViewDef

アクセス制限を編集する

アクセス制限を編集するテーブルを選択してから、読取りまたは書込みのアクセス制限を定義します。これらの条件は、**Asset Manager**のクエリエディタを使って定義できます。

読取り条件

クエリエディタで定義する基準を満たすレコードのみを表示できます。他のレコードは一切表示されません。

書込み条件

既存のレコード内のフィールドの書込み条件を変更できます。



警告:

制限するテーブルに対応するユーザ権限を手動で作成する必要があります。
Asset Managerでは自動的に処理されません。

変更権限を検証する

このフィールドは、変更権限の検証の条件を指定します。【書込み】フィールドに入力されたAQLスクリプトに応じて、このフィールドを使用します。

変更権限の検証の例

【書込み】（SQL名：WriteCond）フィールドに以下のスクリプトを入力します。

```
seAssignment=1
```

【変更前】を選択する場合は、在庫に入っている品目を変更または削除できません。

【変更後】を選択する場合は、在庫に品目を入庫し、品目を変更することができます。


【自動】を選択した場合は、在庫に品目を入庫し、在庫品目を変更することが可能ですが、削除はできません。

フィールドとリンクのデフォルト値を定義する場合の注意

フィールドのデフォルト値は、アクセス制限に関係なく計算されます。そのため、デフォルト値を定義する場合は、すべてのユーザが表示できるフィールドとリンクのみを参照するように設定してください。

機能権限を定義する

次のいずれかの方法で機能権限を管理する画面を表示します。

- 【管理／権限／機能権限】メニューを使う。
- または、ユーザプロファイルと関連付けられる機能権限のリストの  (Windowsクライアント) または【追加】 (Webクライアント) をクリックする (ユーザ役割詳細の【役割、権限および制限】タブ)。

機能権限はアプリケーションの機能ドメインを定義し (【項目】列)、この機能ドメイン用のユーザの権限を指定します。

例：

例えば【調達】の機能ドメインでは、請求の責任者はサプライヤの請求書にはアクセスできますが、予約にはアクセスできません。

Windowsクライアントでは、機能権限の編集画面に、ユーザが定義するAsset Managerの機能ドメインの完全な階層リストが表示されます。

Webクライアントでは、画面が次の2つの要素で構成されます。

- 機能ドメインのリスト
 - **【Show entries (エントリの表示)】** フィールド：表示ごとの機能ドメイン数を選択します。
 - **【検索】** フィールド：このフィールドに入力した文字列が、機能ドメインの**名前**または**SQL名**に含まれる機能ドメインが表示されます。
 - テーブルリスト：機能権限を編集しようとする機能ドメインの **【名前】** または **【SQL名】** をクリックすると、そのドメインの機能権限の詳細が、この画面下部の階層ビューに表示されます。
- 選択されている機能ドメインの機能権限の階層リスト
選択されている機能ドメインの下の子アイテムごとに、機能権限を定義できます。あるドメインの機能権限の定義を完了したら、**【Cache Update (キャッシュの更新)】** をクリックして選択内容をキャッシュし、引き続き別のドメインを設定できます。

機能権限の論理

各機能ドメイン（**【項目】**列）は複数の子アイテムから構成されています。これらの子アイテムにアクセス権限（**【許可】**、**【拒否】**、**【親の値】**）を割り当てます。機能ドメインの子アイテムは、その大部分がこのドメインで使用可能な画面とウィザードです。以下で、機能権限の定義規則について詳しく説明します。

- あるアイテムに**【親の値】**が割り当てられる場合、そのアイテムは親の権限値を継承します。
- **【デフォルトで付与】** オプションは最上位のアイテムのみが対象です。権限は、**【親の値】** オプションが選択されている最上位アイテムに付与されます。
- あるドメインの子アイテムが、そのドメインの権限値と異なる権限値を持つこともできます。その場合、親の権限値とは無関係に、**【許可】** または **【拒否】** を割り当てることができます。

例えば、ある機能ドメインのアイテム全体に値**【拒否】**が割り当てられるためには、以下の条件を満たす必要があります。

- **【デフォルトで付与】** オプションが選択されていない。
- 機能ドメインの値が**「親の値」**である。
- ドメインの子アイテム全体の値が**「親の値」**である。

または

- **【デフォルトで付与】** オプションが選択されている。
- 機能ドメインの値が**「拒否」**である。
- ドメインの子アイテム全体の値が**「親の値」**である。

または

- **【デフォルトで付与】** オプションが選択されていない。
- 機能ドメインの値が**「拒否」**である。

- ドメインの子アイテム全体の値が「親の値」である。

Asset Managerユーザにアクセス条件を関連付ける

Asset Managerユーザにアクセス条件を関連付けるには：


- 1 従業員のリストを表示します（ナビゲーションバーの [組織管理/組織/従業員] リンク）。
- 2 従業員に対応するレコードがまだ存在しない場合、作成します。
- 3 従業員の詳細画面の [プロフィール] タブページに移動します。
- 4 ユーザ名とパスワードを [ユーザ名]（SQL名：UserLogin）および [パスワード]（SQL名：LoginPassword）フィールドに入力します。ユーザ名は、ユーザがデータベースに接続するときの名前です。文字で指定できます。

注意:

パスワードを指定しない場合は、自動的に [ログイン] 名と同じパスワードになります。ユーザ名を指定してデータベースに接続してからパスワードを変更することもできます。

ヒント:

パスワード管理が別のシステム（LDAPサーバ、Active Directory、Active Directory など）で処理される場合、Asset Managerデータベースに入力されるパスワードは、「Admin」を除き、処理対象外です。

- 5 [アクセスタイプ]（SQL名：seLoginClass）フィールドに入力します。様々なオプションについては、次の項を参照してください。
- 6 [承認されたユーザ役割]（SQL名：MasterProfiles）を割り当てます。ユーザ役割は従業員に関連付けることができます。
 - Windowsクライアント：[接続に使用する役割] フィールドの右にある  をクリックし、表示された画面で [新規作成] をクリックします。
 - Webクライアント：詳細画面の右下のリストで [追加] をクリックします。
- 7 [接続に使用する役割]（SQL名：DefMastProfile）で、従業員がデータベース接続時に使用するユーザ役割を選択します。このフィールドで使用可能な値は、[承認されたユーザ役割] フィールド（SQL名：MasterProfiles）に割り当てられた値です。接続に使用する役割は（承認されたユーザ役割と同様に）、そのユーザに対するアクセス条件（ユーザプロフィールとユーザ権限、アクセス制限、関連する機能権限）を網羅します。



注意:

ユーザに管理者権限を付与することもできます▶ **データベース管理者を定義する** [P. 176]。この場合、管理者権限は前回そのユーザに割り当てられた特定のアクセス条件に優先し、[接続に使用する役割] (SQL名: DefMastProfile) および [承認されたユーザ役割] (SQL名: MasterProfiles) フィールドは、従業員の詳細画面の [プロフィール] タブに表示されなくなります。

データベース管理者を定義する

従業員の詳細画面の [プロフィール] タブの [管理者権限] ボックス (SQL名: bAdminRight) をオンにします。すると、Asset Managerデータベースのすべてのテーブルに対するすべての権限と、データベースの設定に関するすべての権限がその従業員に与えられます。



注意:

[部署と従業員] テーブルにはデフォルトの管理者が存在します。これは、[ログイン] が「Admin」のレコードです。初めてAsset Managerをインストールしたときは、「Admin」だけがAsset Managerデータベースのあらゆる管理業務を実行できるログイン名です。

セキュリティ上の理由により、この「Admin」ログインのレコードは削除できません。何らかの理由で管理者権限を持つ [ログイン] 名を使ってもデータベースに接続できない場合は、この「Admin」ログインを使う必要があります。

ユーザの接続に使用する役割を選択する

ユーザは、[**Asset Managerユーザ役割を選択**] ウィザードにより自らの「接続に使用する役割」を変更することができます。ウィザードは以下の場合に表示されます。

- ユーザがAsset Managerに接続しており、管理者がそのユーザに [**Asset Managerユーザ役割を選択**] アクションを有効にする場合は、自動的に表示 (ユーザの詳細画面の [プロフィール] タブ上の [接続時のアクション] (LoginAction) フィールド)。
- ナビゲーションバーの [管理/ユーザのアクション/**Asset Managerユーザ役割を選択**] リンクから。

新しい「接続に使用する役割」は、そのユーザによる次回のデータベース接続で有効です。

データベースへの接続数を管理する

このセクションでは、データベースに接続するユーザ数を管理する方法について説明します。

データベースのアクセスタイプ

Asset Managerのデータベースには、数種類のアクセスタイプがあります。従業員のアクセスタイプは、従業員の詳細画面の【プロフィール】タブで定義します。

アクセスタイプによって、Asset Manager WebおよびWindowsインタフェース、またはAsset Manager APIによるデータベースへの接続が決まります。

注意:

アクセスタイプは、データベース接続が介するAsset Manager Export Tool、Asset Manager Automated Process Manager、またはAsset Manager Application Designerを考慮しません。

表 12.1. データベースのアクセスタイプ

アクセスタイプ	不特定	固定	一時
作成可能なログイン数	無制限	ライセンスによる制限あり	無制限
データベースへの同時接続数	ライセンスによる制限あり	無制限	ライセンスによる制限あり
同じログインで同時に接続できるか	許可されますが、推奨しません。各々の接続により、残りの接続可能数が減ります。	許可されません	許可されますが、推奨しません。各々の接続により、残りの接続可能数が減ります。

アクセスタイプ	不特定	固定	一時
テーブルへのアクセス	アクセス制限なし	アクセス制限なし	<p>アクセス制限あり（▶下の表）。</p> <p>アクセス制限を解除することはできませんが、そのユーザの役割と関連付けられた【一時ユーザ】プロファイルから別のアクセス制限を追加することはできます。このプロファイルは、アクセス管理専門分野データで利用可能：▶前提条件：アクセス管理専門分野データのインポート [P. 166]</p> <p>一時ユーザを作成する場合は、他のアクセスタイプの場合と同様に、その一時ユーザに適切なユーザ権限と機能権限のあるプロファイルが付いたユーザ役割を割り当てる必要があります。</p>

 **注意:**

「Admin」 [ログイン] のアクセスタイプは、固定です。「Admin」 [ログイン] は、登録ユーザ数には含まれません。

アクセスタイプが「一時」のユーザは、以下のテーブルへのアクセス権だけを持っています。

表 12.2. 「一時」アクセスタイプ - 読み書きモードでアクセス可能なテーブル

テーブルのSQL名	読取りアクセス	書込みアクセス
amAccessRestr	X	
amAction	X	
amActionMemo	X	
amBrand	X	
amBudgCenter	X	
amBudget	X	
amCatalog	X	
amCatProduct	X	

テーブルのSQL名	読取りアクセス	書き込みアクセス
amCatRef	X	
amClientResource	X	
amCnxType	X	X
amComment	X	X
amCompany	X	
amConnection	X	X
amContact	X	
amCostCategory	X	
amCostCenter	X	
amCounter	X	X
amCountry	X	
amCRType	X	
amCurrency	X	
amDocument	X	X
amDownTimePeriod	X	
amEmplDept	X	
amEmplGroup	X	
amEscalScheme	X	
amEscSchLevel	X	
amExpenseLine	X	X
amFuncDomain	X	
amFuncRight	X	
amFVExpenseLine	X	X
amHelp	X	
amHelpMemo	X	
amHistory	X	X
amImage	X	X
amItemizedList	X	
amItemListVal	X	
amKnowlBase	X	X
amLocation	X	
amLoginSlot	X	X
amMail	X	X
amModel	X	
amModelPort	X	
amModelSlot	X	
amNature	X	
amNews	X	
amPasswordHist	X	
amPortFolio	X	
amProblemClass	X	
amProdClassCode	X	
amProfile	X	
amProject	X	
amQuery	X	
amRelAstReplReq	X	X
amRelCalcFeat	X	X

テーブルのSQL名	読取りアクセス	書込みアクセス
amRelCatalogClients	X	X
amRelCatalogSuppliers	X	X
amRelCCatClass	X	
amRelCountLic	X	X
amRelCountSoft	X	X
amRelEmplEntitl	X	X
amRelEmplGrp	X	X
amRelEmplGrpCntr	X	X
amRelEntitlContract	X	X
amRelEntitlLoc	X	X
amRelEntitlPortf	X	X
amRelEntitlReq	X	X
amRelEscSchLoca	X	X
amRelEscSchSev	X	X
amRelFixedAsset	X	X
amRelFRProfile	X	X
amRelGrpLoca	X	X
amRelGrpPbClass	X	X
amRelItemAlarmTrig	X	X
amRelLoanAstSel	X	X
amRelLoanPayExpLin	X	X
amRelModelCompat	X	X
amRelModelEntitl	X	X
amRelNewsGrp	X	X
amRelPOrdReq	X	X
amRelPortfReql	X	X
amRelPortfUsers	X	X
amRelProfileRight	X	X
amRelRestrProfile	X	X
amRelSlotTypeModel	X	X
amRelSuppPCardType	X	X
amRelTermLoc	X	X
amRelTickAlarmTrig	X	X
amRelWfActivAct	X	X
amRelWOItems	X	X
amScriptLibrary	X	
amSeverity	X	
amSlot	X	
amSlotType	X	
amSoftLicCounter	X	
amStatistic	X	
amStock	X	
amSysConfig	X	X
amTaxFormula	X	
amTaxJuris	X	
amTaxRate	X	
amTaxType	X	

テーブルのSQL名	読取りアクセス	書き込みアクセス
amTimeZone	X	
amTraining	X	
amUnit	X	
amUserRight	X	
amViewDef	X	
amWfActivAlarm	X	X
amWfActivity	X	
amWfEvent	X	X
amWfInstance	X	X
amWfOccurEvent	X	X
amWfOrgRole	X	
amWfScheme	X	
amWfSyncPoint	X	
amWfSysActiv	X	
amWfTransition	X	
amWfUserChoice	X	
amWfWorkItem	X	X
amWkEvtScript	X	
amWorkCalendar	X	

表 12.3. 「一時」アクセスタイプ - 特定の条件下で、読み書きモードでアクセスが可能なテーブル

テーブルのSQL名	読取り条件	書き込み条件
amReqLine	(CurrentUserId() IN (Request.Requester,Request.Requester.Supervisor) AND Request.seStatus<>1) OR (Request.seStatus=1)	CurrentUserId() IN (Request.Requester,Request.Requester.Supervisor)
amRequest	(CurrentUserId() IN (Requester,Requester.Supervisor) AND seStatus<>1) OR (seStatus=1)	CurrentUserId() IN (Requester,Requester.Supervisor)
amTicket	CurrentUserId() IN (Assignee,Contact,Contact.Supervisor)	CurrentUserId() IN (Assignee,Contact,Contact.Supervisor)
amTicketHistLine	(CurrentUserId()=Ticket.Contact.Parent.Supervisor OR CurrentUserId()=Ticket.Contact)	(CurrentUserId()=Ticket.Contact.Parent.Supervisor OR CurrentUserId()=Ticket.Contact)
amWorkOrder	CurrentUserId() IN (Requester,Requester.Supervisor,Technician)	CurrentUserId() IN (Requester,Requester.Supervisor,Technician)
amPhone	無条件読取りアクセス	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)
amPort	無条件読取りアクセス	CurrentUserId() IN (User,User.Supervisor)
amSoftInstall	無条件読取りアクセス	CurrentUserId() IN (Portfolio.User,Portfolio.User.Supervisor)
amAsset	無条件読取りアクセス	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)

テーブルのSQL 名	読取り条件	書込み条件
amComputer	無条件読取りアクセス	CurrentUserId() IN (Portfolio.User,Portfolio.User.Supervisor)
amMonitor	無条件読取りアクセス	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)

接続の有効期間

指定した接続の有効期間を定義することができます。この定義は、従業員の詳細画面の【プロフィール】タブ、【時間に基づくアクセス制御】フレームで行います。

有効期間は、【有効期間の開始日】と【有効期間の終了日】フィールドで指定した日付に対応しており、自国のカレンダーを指定した日付に関連付けることができます（【カレンダー】フィールド）。

接続スロットの機能

Asset Manager WindowsクライアントまたはWebクライアントからユーザがデータベースに接続するときは、いつでも接続スロットが割り当てられます。

ユーザがデータベースに接続している間、Asset Managerは接続スロットを定期的に更新します。

- Windowsクライアント

更新の周期は、データベースオプションメニュー（【管理/データベースオプション】）の【アクセスコントロール】セクションの【スロットの更新間隔（分）】オプションで定義します。デフォルトでは5分に設定されています。

- Webクライアント

更新の周期は、「<Tomcatインストールフォルダ>\webapps\AssetManager\WEB-INF\web.xml」ファイルの **slotUpdateInterval** 設定エントリで定義します。デフォルトでは300秒に設定されています。

```
<env-entry> <description>Loginslot update interval in seconds</description>
<env-entry-name>slotUpdateInterval</env-entry-name> <env-entry-type>java.lang.Integer</env-entry-type>
<env-entry-value>300</env-entry-value> </env-entry>
```

 **注意:**

接続スロットの更新処理では、各オンラインユーザに対してデータベースへのクエリが追加で発生します。そのため、Asset Managerの同時ユーザー数が非常に多い場合は、ユーザアクティビティをリアルタイムに監視する必要がある場合は、更新周期を長くすることをお勧めします。

 **注意:**

Web接続スロット情報は、ユーザのログイン後、すぐにamLoginSlotテーブルに反映されるわけではありません。

ユーザがデータベースを切断すると、対応する接続スロットは直ちに破棄されます。

接続スロットの詳細

Asset Managerの管理者は、**【管理／接続／接続スロット】** ナビゲーションメニューを使って接続スロットを表示できます。

接続スロットには、次の情報が含まれます。

- スロットを使うアプリケーションの名前（一般にAsset Managerなど）
- ユーザの **【ログイン】**
- ユーザの **【アクセスタイプ】**（SQL名：seLoginClass）

 **ヒント:**

接続スロット（SQL名：amLoginSlot）テーブルに関するワークフローを作成して、固定ユーザと不特定ユーザの数をカウントできます。ワークフローの履歴統計により、Asset Managerがどのように使用されているか明確に把握できます。

接続スロットを破棄する

接続スロットは、次の場合に破棄されます。

- 接続スロットの詳細画面で **【削除】** ボタンをクリックした場合
- アプリケーションが異常終了した場合
- 接続した状態で長時間作業を行わない場合

接続スロットを手動で破棄する

- 1 **【管理／接続／接続スロット】** ナビゲーションメニューを選択して、接続スロットの詳細画面を表示します。
- 2 切断する接続スロットを選択します。
- 3 **【削除】** をクリックします。

アプリケーションの異常終了によるアイドルスロットを検出する

オペレーティングシステムの強制的な終了や、ネットワーク障害などにより、接続スロットを使うアプリケーションが異常終了することがあります。この場合、ユーザは正常な方法（Windowsクライアントの場合は【ファイル/データベースの接続解除】メニューの選択、Webクライアントの場合は【ログアウト】のクリック）でデータベースからログオフされたわけではありません。

接続スロットはまだ存在しますが、実際には使っていないこととなります。

その結果、データベースへのアクセスタイプが不特定ユーザの場合は、有効な同時接続数が1減ります。

Asset Managerは、このような使用されていない接続スロットを定期的に見つけ、これを新しい接続に使うことでこの問題を解決します。Asset Managerは以下の方法で、使用されていない接続スロットを検出します。

- 1 Asset Managerは、スロットが最後に変更された日付を調べます。
- 2 スロットの最後の変更日からスロット更新間隔の2倍以上の時間が経過している場合は、Asset Managerはスロットが無効であり、再利用できると見なします。

データベースに接続している間は接続スロットが定期的に更新されるので、接続スロットのテーブルから使わなくなったレコードを削除するのは容易です。

- 1 フィルタを使って、接続スロットのテーブル内で【更新日】フィールドが特定の日付より前のレコードをすべて表示します。



注意:

【更新日】（SQL名：dtLastModif）フィールドは、レコード詳細ウィンドウから見ることはできません。

- 2 これらのレコードのスロットを選択して削除します。

何もしていないユーザを切断する

Asset Managerでは、タイムアウトを設定して非アクティブなユーザを自動的に切断できます。

Windowsクライアント

タイムアウトを定義するには、【管理/データベースオプション】メニューを選択します。

- 1 接続の自動切断機能を有効にします。
 - 【アクセスコントロール】セクションの【自動切断を有効にする】オプションを使用すると、不特定ユーザの自動切断を有効にすることができます。
 - 【アクセスコントロール】セクションの【特定ユーザの自動切断を有効にする】オプションを使用すると、特定ユーザの自動切断を有効にすることができます。

- 2 自動切断機能を有効にしたら、[アクセスコントロール] セクションの [スロットの自動タイムアウト(秒)] オプションで時間を指定します。

 **警告:**

データベースオプションの変更内容を有効にするには、いったん切断してから再度接続する必要があります。

 **注意:**

バージョン3.01以前のAsset Managerでは、この自動切断機能が、全てのユーザーに適用されます。

Webクライアント

非アクティブなWebクライアントユーザは、アプリケーションサーバのセッションタイムアウト時間が経過したときに自動的に切断されます。以下に、Asset Managerでサポートするさまざまなアプリケーションサーバのセッションタイムアウト値を設定する方法を示します。

Apache Tomcat

「<Tomcatインストールフォルダ>\conf\web.xml」の<session-timeout>パラメータでタイムアウト値を設定します。

デフォルトのタイムアウト値は30分です。

```
<session-config> <session-timeout>30</session-timeout> </session-config>
```

Oracle WebLogic

「web.xml」または「weblogic.xml」のいずれかでタイムアウト値を定義できます。

- 「web.xml」ファイルでは、<session-timeout>パラメータを使って分単位で定義します。デフォルト値は0です。

```
<session-config> <session-timeout>0</session-timeout> </session-config>
```

- 「weblogic.xml」ファイルでは、TimeoutSecs要素を使って秒単位で定義します。デフォルト値は3600です。

```
<session-param> <param-name>TimeoutSecs</param-name> <param-value>3600</param-value> </session-param> </session-descriptor>
```

IBM WebSphere

「web.xml」ファイルの<session-timeout>パラメータでタイムアウト値を設定します。デフォルト値は30分です。

```
<session-config> <session-timeout>30</session-timeout> </session-config>
```

Asset Managerデータベースへのユーザ認証を管理する

このセクションでは、異なる認証システムを介したAsset Managerデータベースへのユーザアクセスを管理する方法について説明します。

これらのシステムには、すでに提供された認証情報を別のシステムへのアクセス時に使用してAsset Manager WindowsクライアントとWebクライアントにAsset Managerデータベースへのアクセスを可能にし、ユーザの再認証を必要としないものがあります。

注意:

シングルサインオン (SSO) 定義は、ディレクトリベースのソフトウェアソリューションであり、ユーザがネットワーク認証手順を1回完了すると、組織のネットワークユーザに全正規リソースへのシームレスなアクセスを提供します。

ユーザは1つのパスワードで、組織内のすべてのマルチプラットフォームアプリケーションとシステムにアクセスできます。1回目の認証中にユーザがパスワードを入力すると、ユーザに再びパスワード情報を依頼することなく、ディレクトリ (例えば、Windows下のActive Directory) がパスワードを必要とするその他すべてのアプリケーションに自動でパスワードを送信します。

警告:

シングルサインオンを導入するには、WindowsクライアントとWebクライアントをシングルサインオンで使用するための特別な設定が必要です。

Asset Managerデータベースへのユーザアクセスを制御するパスワードは、以下の方法で格納されます。

Asset ManagerのWindowsクライアントとWebクライアント : 「Admin」 ログイン - 特別な場合

ユーザのパスワードがAsset ManagerデータベースまたはLDAPディレクトリのどちらに格納されていても、「Admin」ログインのパスワードは、常時Asset Managerデータベース内に格納されます。

警告:

「Admin」ログインと、管理者権限を付与されたユーザとを混同しないでください。

「Admin」ログインの機能

「Admin」ログインのレコードは非常に重要です。

- 1 「Admin」は、最初にAsset Managerをインストールした時に、Asset Managerデータベースにアクセスしてあらゆる管理操作を実行できる唯一のログイン名です。
- 2 データベースを設定したら、[部署と従業員] テーブルの他のレコードに、管理者権限を割り当てることができます。ただし、「Admin」のログインレコードは削除できません。

何らかの理由で管理者権限を持つ[ログイン]名でデータベースに接続できない場合、「Admin」ログインが唯一の接続方法となります。

「Admin」ログインのパスワードは、Asset Managerデータベースのすべてのアクセス権限を付与するので、非常に重要です。

警告:

「Admin」ログインのパスワードを忘れないようにしてください。パスワードがないと、管理者権限を持つ他のレコードが[部署と従業員]のテーブル内で壊れた場合に、データベースを管理できなくなります。

警告:

「Admin」ログインのパスワードを知っていれば、誰でもAsset Managerデータベースですべての操作を実行でき、すべてのデータベース情報に無制限にアクセスできます。したがって、権限のないユーザにはこのパスワードを絶対に教えないでください。

注意:

上記の注意にも関わらずこのパスワードを忘れた場合は、Asset Managerによってパスワードを回復する手段はありません。HPサポートへお問い合わせください。

Adminログインのパスワードを変更する

Windowsクライアント

「Admin」ログインのレコードのパスワードは、「Admin」ログインでデータベースを開き、次の手順で変更できます。

- [ツール/パスワードの変更] メニューを使う。
- ナビゲーションツリーから[管理/ユーザのアクション/パスワードの変更]を選択する。

Webクライアント

「Admin」でWebクライアントにログインし、ナビゲーションツリーから「管理/ユーザのアクション/パスワードの変更」を選択してパスワードを変更できます。

Asset Manager WindowsクライアントとWebクライアント：Asset Managerによる全面的な認証処理

このセクションでは、ユーザ認証のすべてを管理するためのAsset Managerのセットアップ方法を説明します。

このモードで、ユーザ情報はAsset Managerデータベースにのみ保存されます。ユーザはAsset Managerデータベースに接続するたびに「ログイン」と「パスワード」を入力する必要があります。

パスワード管理の設定

パスワードに関連するデータベースオプションの入力

- 1 Asset Manager WindowsクライアントまたはWebクライアントを起動します。
- 2 管理者権限を持つユーザログインを使用してAsset Managerデータベースに接続します。
- 3 「管理/ユーザのアクション/パスワード管理...」リンクを選択します。
- 4 「パスワード管理...」（sysCorePassMgt）ウィザードに入力します。

このウィザードでは、パスワードに関する標準的なオプション（「新規パスワードのデフォルトの有効日数」、「パスワードの期限切れ通知」）の他に、スクリプトライブラリによって呼ばれるオプション（「パスワードの最小文字数」、「異なる文字による最小文字数」）も定義できます。

注意:

パスワードの有効期限が近づくと、Asset ManagerのWindowsクライアントおよびWebクライアントにログインしているユーザに、パスワードを変更するように通知が表示されます。「パスワードの期限切れ通知」は、どの程度前にパスワードを変更するかを定義します。

有効期限に達したパスワードでAsset ManagerのWindowsクライアントおよびWebクライアントにログインしようとした場合、そのユーザはパスワードを変更してからでなければログインできません。

ウィザードにはデフォルトの値セットがあります。これらの値は「アプリケーションのオプション」テーブルに保存され、変更することができます。

ウィザードはデータベースオプションで定義したスクリプトライブラリを自動的に読み込み、「オプション」テーブルに入力した値を保存します。

パスワード形式制約の定義

パスワード形式制約は、スクリプトライブラリで定義されます。デフォルトのライブラリは「**PasswordLib**」と呼ばれます。このライブラリは、ナビゲーションバーの【管理/スクリプト】リンクから変更できます。

重要項目:

ライブラリで定義したBasic関数の中でスクリプトで使用されるものはライブラリ自体と同じ名前に設定する必要があります。

デフォルトのスクリプトでユーザがパスワードを登録するには、次のルールを守らなければなりません。

- パスワードは、最小限の文字数を含むこと
- パスワードには同じ文字を続けて使用しないこと
- パスワードはログイン名を含まないこと

Asset Managerデータベースに接続可能なユーザの作成

- 1 Asset Manager WindowsクライアントまたはWebクライアントを起動します。
- 2 管理者権限を持つユーザログインを使用してAsset Managerデータベースに接続します。

注意:

これらの権限を有効にするには、従業員の詳細画面の【プロフィール】タブの【管理者権限】(bAdminRight) オプションを有効にします。

- 3 従業員を表示します (ナビゲーションバーの【組織管理/組織/従業員】リンク)。
- 4 まだ存在しない場合、その従業員を作成します。
- 5 【プロフィール】タブに情報を入力します。
- 6 【ユーザ名】(UserLogin) および【パスワード】(LoginPassword) フィールドに入力します。

これらのフィールドに使用した値は、データベース接続時に必要となります。

パスワード変更

ユーザ自身に変更する

- 1 Asset Manager WindowsクライアントまたはWebクライアントを起動します。
- 2 現在のユーザ名とパスワードでAsset Managerデータベースに接続します。
- 3 【管理/ユーザのアクション/パスワードの変更...】リンクを選択します。

- 4 ユーザーの入力画面に入力します。

管理者が変更する

ユーザーパスワードは、従業員の詳細画面の【プロフィール】タブでその管理者権限が有効になっているユーザーにより変更できます。作成した時点で古いパスワードはなくなります。

ユーザーのロックアウト

- 1 Asset Manager WindowsクライアントまたはWebクライアントを起動します。
- 2 管理者権限を持つユーザーログインを使用してAsset Managerデータベースに接続します。
- 3 従業員を表示します（ナビゲーションバーの【組織管理／組織／従業員】リンク）。
- 4 従業員を選択します。
- 5 【プロフィール】タブページを表示します。
- 6 【ユーザーアカウントのステータス】フィールド（seLoginStatus）を【ロック】に設定します。

ユーザーのアクセス有効期間を定義する

- 1 Asset Manager WindowsクライアントまたはWebクライアントを起動します。
- 2 管理者権限を持つユーザーログインを使用してAsset Managerデータベースに接続します。
- 3 従業員を表示します（ナビゲーションバーの【組織管理／組織／従業員】リンク）。
- 4 従業員を選択します。
- 5 【プロフィール】タブページを表示します。
- 6 【時間に基づくアクセス制御】フレームに情報を入力します。

パスワードを忘れた場合

Asset Managerでは、他のユーザーが読めないようにパスワードがアスタリスク(*)で表示されます。Asset Managerのユーザーが自分のパスワードを忘れた場合は、管理者が従業員の詳細画面の【プロフィール】タブページで新しいパスワードを作成する必要があります。作成した時点で古いパスワードはなくなります。

Asset Manager WindowsクライアントとWebクライアント：LDAPによる全面的な認証処理

LDAPについて

LDAP（Lightweight Directory Access Protocol）は、ディレクトリサービスにアクセスするためのプロトコルです。

Microsoft Exchangeなどの一部のメッセージサーバは、ディレクトリを公開するのにこのプロトコルを使用しています。

LDAPディレクトリ内でのパスワード格納

企業がLDAPディレクトリで統括して従業員のパスワードを管理している場合、Asset Managerデータベースへのアクセスを制御するためにLDAPを使用すると便利です。

以下のようなLDAPディレクトリのパスワード管理機能を、使用することも可能になります。

- パスワードの定期変更を強制する
- 文字数とフォーマットの制御
- その他

LDAPディレクトリの導入

- 1 Asset Managerの全ユーザがLDAPディレクトリにアクセスできるようにします。
- 2 Asset ManagerユーザをLDAPディレクトリ内に作成し、ユーザをログインとパスワードに関連付けます。
- 3 LDAPディレクトリへの匿名アクセスを無効にします。
- 4 Asset Managerを起動します。
- 5 Asset Manager Windowsクライアントを起動します。
- 6 本番データベースに接続します（[ファイル/データベースに接続]メニュー）。
- 7 データベースオプションを表示します（[管理/データベースオプション]メニュー）。
- 8 LDAP認証項目のオプションにデータを入力します。
- 9 [部署と従業員]（SQL名：amEmplDept）テーブルの[プロファイル]タブのフィールドに値を入力して、ユーザを作成/変更します。

特に次のフィールドに値を入力します。

- [ユーザ名]（SQL名：UserLogin）：Asset Managerデータベースへの接続時に使用するログインを入力します。
- LDAPディレクトリのログインを格納するフィールド（[UserLogin]フィールドでない場合）

LDAPログインの例

- Microsoft Exchange : CN=<識別子>, CN=<ドメイン名>またはDC=<ドメイン名>, CN=<識別子>
- Lotus Domino : <名> <姓>/<ドメイン名>

 **ヒント:**

HP ConnectItを使用するとLDAPディレクトリをインポートできます。

- 10 LDAP DLLがAsset Managerユーザのコンピュータにインストールされていることを確認してください。
Asset Managerクライアントのインストール時に [完全インストール] オプションを選択した場合は、LDAP DLLがインストールされます。
[カスタムインストール] オプションを選択し、[LDAP認証] を選択した場合も、LDAP DLLがインストールされます。
ただし、[標準インストール] オプションを選択した場合、LDAP DLLはインストールされません。
- 11 Asset Managerクライアントで新規オプションが有効になるよう、ユーザにデータベースへ再接続してもらいます。

Asset ManagerデータベースへのLDAP接続

Asset ManagerデータベースへLDAP認証を導入した後、ユーザは以下の手順で接続します。

- 1 Asset Manager WindowsクライアントまたはWebクライアントを起動します。
- 2 本番データベースに接続します。
 - ユーザ名：ユーザの詳細画面の [ユーザ名] (SQL名：UserLogin) フィールドにある [部署と従業員] (SQL名：amEmplDept) テーブルに保存されるユーザ名。
Asset Managerは、[管理/データベースオプション] メニューで指定したパラメータを使って、LDAPディレクトリのログインを検索します。
 - パスワード
 - Microsoft Exchange : Windows パスワード。
 - Lotus Domino : Dominoで定義されたInternetパスワード
パスワードの有効性はLDAPサーバ経由で検証されます。

LDAPサーバへのセキュア接続 (SSL)

LDAPサーバへのセキュアな接続を確保するために、Asset Managerは2つのオプションを用意しています。

- 1 LDAPCertFile : このオプションは、ただ1つのOpenLDAPサーバを必要とし、ただ1つの認証証明書を使用します。

このオプションを有効にするには、「LDAPCertFile」ファイルで、次の構文を使用して、**PEM**で証明書を持つテキストファイルを指定します。

```
-----BEGIN CERTIFICATE----- ...(CA certificate in base64 encoding) -----  
END CERTIFICATE-----
```

- 2 LDAPCertDir : このオプションはより複雑であり、複数のOpenLDAPサーバを必要とし、複数の認証証明書を使用します。

 **注意:**

このオプションは、HP Connect-Itで頻繁に使用されます。

このオプションを使用するには、LDAPCertDirファイルで、ハッシュコードに基づく名前の証明書を含むディレクトリを指定します。フォルダの各ファイルは**PEM**形式で保存し、各証明書のsubjectを含めます。

現在のディレクトリでハッシュ関数を有効にし、Unixでリンクを作成するには、c_rehashコマンドを使用します。

 **警告:**

LDAPサーバへのセキュア接続を有効にするには、接続前にすべてのコンピュータを設定しておく必要があります。

Asset Manager Windowsクライアント : IIS上のActive Directoryを使用するシングルサインオンを導入する

Active Directoryを使用してAsset Manager Windowsクライアントユーザを定義する方法を説明します。

Windowsの統合型認証により、Windows起動時に認証したユーザは、Asset Manager Windowsクライアントの起動時に認証する必要がありません。

Asset Managerでユーザが持つ権限が、Asset Managerデータベースで定義されます（ユーザ役割）。

シングルサインオンを正しく動作させるには、Asset ManagerとActive Directoryとの間の認証情報を同期する必要があります。

同期を実行すると、以下のことが可能になります。

- Active Directoryユーザのリストは、自動的にAsset Managerデータベースにインポートされます。
- Windowsユーザは、[ログイン] フィールドおよび[パスワード] フィールドに入力することなく、データベースに接続できます。

概要

Asset ManagerとActive Directoryとの間の同期は、次のいずれかの方法で実行されます。

すべてのActive Directoryユーザを同期する

- 同期はHP Connect-Itのシナリオで実行されます。
- Asset Manager Automated Process ManagerがHP Connect-Itのシナリオをスケジュール設定し、トリガします。
- HP Connect-Itは以下のフィールドを使ってAsset Managerデータベースの人員を識別します。
 - [部署名/姓] (Name)
 - [名前] (FirstName)
 - [ユーザ名] (UserLogin)
- 以下の項目の値が同一である場合、ユーザは指定のログインによるWindows統合セキュリティを使用して、Asset Managerデータベースに接続することを許可されます。
 - Asset Managerユーザの [SID] (識別子) フィールド
 - Active Directory [SID] フィールド

Asset Managerデータベースでユーザを同期する

ユーザの詳細画面で、[NTユーザ] ボタンをクリックしてウィザードを起動します。

完全同期を導入するための主要手順

導入手順は以下のとおりです。

- 手順1：Active Directoryでユーザとグループを作成する [P. 194]
目的：Asset Managerに転送する情報を準備する。
- 手順2：Asset Manager Automated Process Manager とHP Connect-Itを設定する [P. 195]
目的：ユーザ情報のやりとりをどのNTドメインとの間でするかを定義します。
- 手順3：更新プロセスを初めてトリガする [P. 195]
目的：ユーザ情報を転送してみて、ログインをテストする。
- 手順4：Asset Managerデータベースのユーザ情報を補足する [P. 196]
- 手順5：Asset Manager Automated Process Managerモジュールの自動トリガを有効にする [P. 196]

手順1：Active Directoryでユーザとグループを作成する

以下のフィールドに入力します。

- ユーザ情報
 - [フルネーム] フィールド:

 ヒント:

Asset Managerデータベースで従業員（ユーザ）を作成する際に、Asset Manager Automated Process Managerは [完全名] フィールド（Active Directory）の値を取り、左から数えて最初の空白文字を検索します。この最初の空白文字の左にある文字は、[名前] フィールドを作成するために使用されます。空白文字の右にある文字は、[部署名/姓] フィールドを作成するために使用されます。空白文字がない場合は、[部署名/姓] フィールドのみに値が入力されます。

従って、複数の単語で構成される名前を入力する場合、空白文字の代わりにハイフンを使用してください。

- [説明] フィールド
このフィールドは、Asset Managerデータベースの [コメント]（SQL名：Comment）に入力するために使用されます。
- グループ情報： [グループ名] フィールド
このフィールドは、Asset Managerデータベースの [部署名/姓]（Name）に入力するために使用されます。

手順2 : Asset Manager Automated Process Manager とHP ConnectItを設定する

- ▶ 『管理』ガイドの「**Asset Manager Automated Process Manager**」の章、「**Asset Manager Automated Process Managerでモニタするモジュールを設定する / [データベースにNTユーザを追加] モジュール (AddUser)**」のセクション。

手順3 : 更新プロセスを初めてトリガする

- 1 [アクション/起動] メニュー選択します。
- 2 [データベースにNTユーザを追加] チェックボックスをオンにします。
- 3 [OK] をクリックします。

Asset Managerは更新を開始します。

 注意:

このタスクに関与するユーザ数が、データベース内でアクティブにされたライセンスキーで許可された固定ユーザ数を超える場合、Asset Manager Automated Process Managerはユーザを不特定ユーザとして指定します。

手順4 : Asset Managerデータベースのユーザ情報を補足する

データベースは、選択されたドメインで検出されるNTユーザ1人に対して従業員1人を含むこととなります。ログインは次のとおりです。

[ドメイン]\[ユーザ]

作成されたユーザは、有効なパスワードを持ちません。

注意:

この操作の終了後に、[部署と従業員] テーブルで作成されたレコードが、実際にAsset Managerユーザに対応するかどうかを確認することをお勧めします。特に、[パスワード] フィールドの値を再入力する必要があります。

手順5 : Asset Manager Automated Process Managerモジュールの自動トリガを有効にする

- 1 [ツール/モジュールの設定] メニューを選択します。
- 2 [データベースにNTユーザを追加] モジュールを選択します。
- 3 [検証スケジュール] フレーム内のフィールドに入力します。
- 4 [有効] チェックボックスをオンにします。
- 5 [変更] をクリックします。

Asset Managerへの接続

前述の操作が終了すると、NTユーザは、Asset Managerに直接アクセスできるようになります。初めて接続する際に、[統合NTセキュリティ使用] チェックボックスをオンにしてから、[開く] をクリックします。

次の接続からは、認証情報は一切必要ありません。

別のログインで接続する場合は、[ファイル/データベースに接続] メニューの接続ボックスを選択していつでも接続することができます。

ユーザの作成/変更時にAsset Managerが適用する規則

- Asset Managerログインは、Windowsの [ドメイン] 名と [ユーザ名] を連結して<ドメイン名>\<ユーザ名>という形式で作成されます。
- 以下のフィールドがAsset Managerデータベースのユーザの照合更新キーとして使用されます。
 - [部署名/姓] (Name)
 - [名前] (FirstName)
 - [ユーザ名] (UserLogin)
- あるユーザをインポートすると、照合更新キーが既に存在しない場合のみにこのユーザが作成されます。作成されると、HP Connect-Itのシナリオに含まれている全フィールドが入力されます。

照合更新キーに対応するレコードが既に存在する場合は、HP Connect-Itのシナリオにあるフィールドのうち識別キーを構成するフィールドを除いたもの全てが更新されます。

 **注意:**

【パスワード】フィールドは特別な文字が入力されていて、ユーザが自動ログイン以外の方法でAsset Managerデータベースにアクセスできないようになっています。ユーザが手動ログインでデータベースにアクセスすることを許可するには、【パスワード】フィールドに手動で値を入力する必要があります（特に、ユーザが別のコンピュータでログインする場合）。

- Asset Managerでは、【バーコード】フィールドの値がインデックスとして使用されます。このインデックスは固有の値である必要があります。各ユーザに対して異なる【バーコード】が指定されていることを確認してください。この結果、同じ姓名を持つ複数のユーザを区別して作成できます。このためには、Asset Managerによりインストールされたデフォルト値の計算スクリプトを使用します。

Asset Manager Webクライアント：IIS（バージョン7より前）上のActive Directoryを使用するシングルサインオンを導入する

シングルサインオンを使用したAsset Manager Webへのアクセスは、使用する環境に応じていくつかの方法で処理されます。

ここでは、すべての可能性を説明せず、Active DirectoryとIIS 7以前のバージョンのIISを使用したシングルサインオンの認証のみを取り上げます。

IIS 7での手順については、Asset Manager Web クライアント：IIS 7 32ビット上のActive Directoryを使用するシングルサインオン（SSO）を導入する [P. 201] を参照してください。

 **注意:**

Active Directory以外のシングルサインオンシステムを使用することもできます。この例はガイドラインとして使用し、ニーズに合わせて調整してください。

IIS（Internet Information Services）は、Windowsでのシングルサインオンを正しくサポートする唯一のWebサーバです。しかし、IISと通信するように正しく設定されたものであれば、異なるアプリケーションサーバ（Tomcat、WebSphereなど）を使用することもできます。

アプリケーションサーバとしてTomcatとのシングルサインオンのインストールおよび設定が、このセクションで説明されます。

Asset Manager Web ServiceおよびAsset Manager Web Tierをインストールしているマシンで実行する手順を、以下にリストします（異なるマシンにインストールされている場合、各手順が各マシン上で実行される必要があります）。

前提条件

起動前に、以下があることを確認します。

- ネットワーク上で動作可能なActive Directoryの導入
このセクションの手順では、Windows Server 2003 R2で提供されるActive Directoryバージョンを使用しての導入を説明しますが、前バージョンのActive Directoryにも有効です。
- 動作可能なDNS (Domain Name Service) サーバこれをチェックするには、コマンドライン (コマンドプロンプトウィンドウ) で次を入力します。

```
ipconfig
```

次に実行するコマンド :

```
nslookup <DNSサーバのIPアドレス>
```

- アプリケーションサーバとしてTomcatを使用するAsset Manager Web
- IIS (Windowsで提供されるWebサーバ)

Jakarta ISAPI Redirectorプラグインのインストール

まず、IISとTomcatとの間の接続を確立する必要があります。Apacheには、実装する必要があるISAPI ISAPI (Internet Server Application Programming Interface) Redirectorプラグインが用意されています。

公式ドキュメントの「**Apache Tomcat Connector - Webserver HowTo**」 (http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html) を参照してください。

ヒント:

Apache Tomcatコネクタの配布は、次のサイトから利用可能です。
<http://tomcat.apache.org/download-connectors.cgi>

IISの設定

- 1 デスクトップ上で、[マイコンピュータ] を右クリックします。
- 2 [管理] をクリックします。
- 3 ブランチ [サービスとアプリケーション/インターネットインフォメーション サービス/Web サイト/規定の Web サイト] を展開します。
- 4 [jakarta] を右クリックし、次に [プロパティ] をクリックします。
- 5 [仮想ディレクトリ] タブをクリックし、パラメータをチェックします。

以下の項目を有効にする必要があります。

- このリソースに接続する場合、内容はこのコンピュータにあるディレクトリから来る必要があります。
- [ローカルパス] セクション：
 - スクリプトソースアクセス

- 読み取り
 - ログ訪問
 - **【アプリケーション設定】** セクション：
 - アプリケーション名： **[jakarta]**
 - 実行許可： **[スクリプトおよび実行可能ファイル]**
 - (フィールドが使用可能な場合) アプリケーション保護： **[中 (プール)]**
- 6 次に、**【既定の Web サイト】**に戻って右クリックし、**【プロパティ】**をクリックします。
- 【ISAPIフィルタ】** タブ上で、**[jakarta]** フィルタに正しくフィルタが機能していることを示す緑色の上矢印を確認します。
- 確認できない場合：
- a **【追加】** をクリックします。
 - b **【フィルタ名】** フィールド：タイプ **[Jakarta]**
 - c **【実行ファイル】** フィールド：Jakarta ISAPI Redirectorをインストールしたフォルダのサブフォルダ「**bin**」にある「**isapi_redirect.dll**」ファイルを参照します。

例：「C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\bin\isapi_redirect.dll」

Jakarta ISAPI Redirectorのインストールフォルダがユーザマシン上の異なる場所にある場合、「C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\」を正しいインストールフォルダのパスに置き換えます。

このガイドでは、このインストールフォルダを例として使用します。
 - d **【OK】** をクリックし、**【規定のWebサイト】** プロパティページで再度 **【OK】** をクリックします。
 - e プロパティページを開き直します。

Jakarta フィルタの前に緑色の上矢印が確認できるはずです。
- 7 **【規定の Web サイトのプロパティ】** ウィンドウのままで、**【ディレクトリセキュリティ】** タブをクリックします。
- 8 **【認証とアクセス制御】** セクションで **【編集...】** をクリックします。
- 9 **【匿名アクセス】** オプションをクリアします。
- 10 **【Windows ドメイン サーバーでダイジェスト認証を使用する】** と **【統合 Windows 認証】** オプションを有効にします。
- 11 ユーザのドメイン名が **【既定のドメイン】** フィールドに表示されていることを確認します。

ポート転送の設定

- 1 この目的は、**IIS**（デフォルトでポート80を受信）と**Tomcat**（ポート8080を受信）との間のポート転送、および**Tomcat**に認証を依頼するログインウインドウを表示しないように指示することです。

そのため、「C:\Tomcat\conf\jk2.properties」（このファイルの場所は、ユーザのTomcatインストールフォルダによって異なります。以下では、あくまでも例として用います）をテキストエディタで開き、次の2行をファイルの最後に追加します。

```
request.tomcatAuthentication=false request.registerRequests=false
```

- 2 ポート転送が正しく動作することを確認するには、次のテストを実行します。デフォルトのTomcatアプリケーションを開き、次のURLを入力します。
<http://localhost:8080/jsp-examples/>

注意:

動作しない場合、Tomcatが起動されていることを確認します。

ブラウザで次のURLを開きます。

<http://localhost/jsp-examples/>

前と同じページが確認できるはずですが、これにより、ポート転送が正しく設定されていることを確認できました。

- 3 次に、TomcatデフォルトアプリケーションとAsset Managerを管理するようJakartaに指示する必要があります。

そのため、テキストエディタで「C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\conf\uriworkermap.properties」を開き、次の行を適切なセクションに追加します。

- Asset Manager Web Serviceがインストールされているマシンでは、このファイルに次の行を追加します。

```
/AssetManagerWebService/*=wlb
```

- Asset Manager Web Tierがインストールされているマシンでは、このファイルに次の行を追加します。

```
/AssetManager/*=wlb
```

注意:

Asset Manager Web ServiceとAsset Manager Web Tierが同じマシンにインストールされている場合、「C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\conf\uriworkermap.properties」ファイルの適切なセクションに上記の行を追加します。

Tomcatを再起動します。

- 4 Windowsユーザとして接続している場合、再び認証することなく Asset Manager Webクライアントにアクセスできるはずですが。
- 5 ただし、ログイン画面が表示されたままの場合、システムを再起動します。問題が改善しない場合、書込みモードで「C:\Tomcat50\conf\server.xml」ファイルを開き、次の行のコメント文字（<!--および-->）を削除します。

```
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
```

これにより、Tomcatは強制的に追加ログファイル「catalina_log.YYYY-MM-DD.txt」を生成します。

ログファイルを開き、次のパラメータの入力をチェックします。

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: queryString=null  
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (IPアドレスをここに入力)  
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (IPアドレスをここに入力)  
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteUser=<ドメイン名>\<ユーザ名>  
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: requestedSessionId=null  
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: scheme=http
```

この例では、シングルサインオンが正しく動作していることを示します。

IISは、ユーザのWindows認証証明書をすべて転送します。そして、Tomcatはそのユーザを認証します。

[remoteUser] フィールドが空の場合、前の手順がすべて正しく完了していることを確認してください。

Asset Manager Web クライアント : IIS 7 32ビット上のActive Directoryを使用するシングルサインオン (SSO) を導入する

このセクションは、特にIIS7 32ビットに適用されます。

前提条件

起動前に、以下があることを確認します。

- ネットワーク上で動作可能なActive Directoryの導入
このセクションの手順では、Windows Server 2003 R2で提供されるActive Directoryバージョンを使用するの導入を説明しますが、前バージョンのActive Directoryにも有効です。
- 動作可能なDNS (Domain Name Service) サーバ
これをチェックするには、コマンドライン (コマンドプロンプトウィンドウ) で次を入力します。

```
ipconfig
```

次に実行するコマンド :

nslookup <DNSサーバのIPアドレス>

- アプリケーションサーバとしてTomcatを使用するAsset Manager
- IIS 7 (Windowsで提供されるWebサーバ、Windows Server 2008とWindows Vistaでのみサポートされています)

Windows Server 2008の場合、以下の設定には、Webサービスのデフォルトのコンポーネントセクションで十分です。

Windows Vistaの場合は、コンポーネントの [ISAPI フィルタ]、[ISAPI 拡張]、および [Windows 認証] の各オプションを手動で選択します。



注意:

IIS 7のインストールの詳細については、次のリンクを参照してください。

<http://learn.iis.net/page.aspx/29/installing-iis7-on-windows-server-2008/>

<http://learn.iis.net/page.aspx/28/installing-iis-70-on-windows-vista/>



注意:

バージョン7以前のIISでの手順については、Asset Manager Webクライアント : IIS (バージョン7より前) 上のActive Directoryを使用するシングルサインオンを導入する [P. 197]を参照してください。

Jakarta ISAPI Redirectorプラグインのインストール

まず、IISとTomcatとの間の接続を確立する必要があります。Apacheには、実装する必要があるISAPI ISAPI (Internet Server Application Programming Interface) Redirectorプラグインが用意されています。

公式ドキュメントの「**Apache Tomcat Connector - Webserver HowTo**」

(http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html) を参照してください。



ヒント:

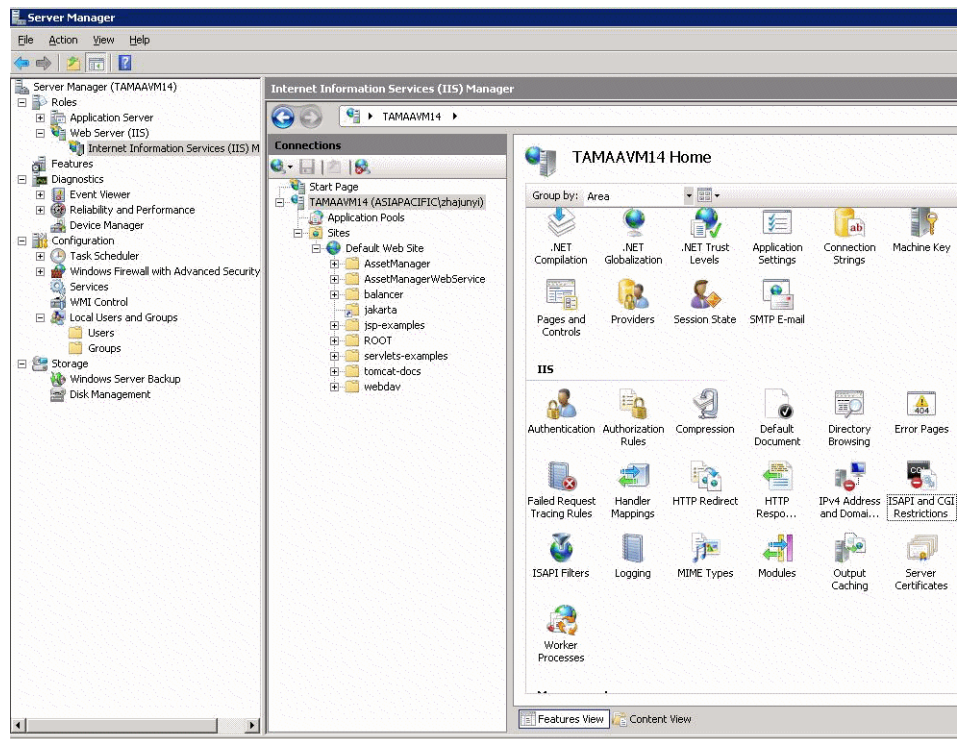
Apache Tomcatコネクタの配布は、次のサイトから利用可能です。

<http://tomcat.apache.org/download-connectors.cgi>

IISの設定

- 1 IISがインストールされているサーバのデスクトップで、[マイコンピュータ] を右クリックします。
- 2 [管理] をクリックします。

- 3 以下に示す例のように、[役割/ Web サーバー (IIS)/ インターネット インフォメーション サービス (IIS) マネージャ] を展開します。



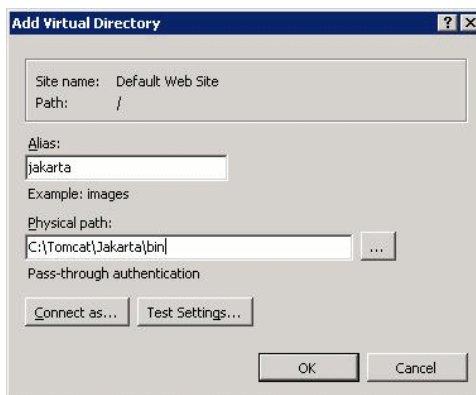
- 4 [インターネットインフォメーションサービス (IIS) マネージャ] パネルで、ホストを選択し、[ISAPI および CGI の制限] をダブルクリックします。
- 5 [アクション] リストの [追加] をクリックします。
- 6 [ISAPI または CGI パス] を「isapi_redirect.dll」ファイル（例：「C:\tomcat\Jakarta\bin\isapi_redirect.dll」）をポイントするように設定し、[説明] を [jakarta] に設定します。
- [拡張パスの実行を許可する] をオンにします。

以下の例を参照してください。



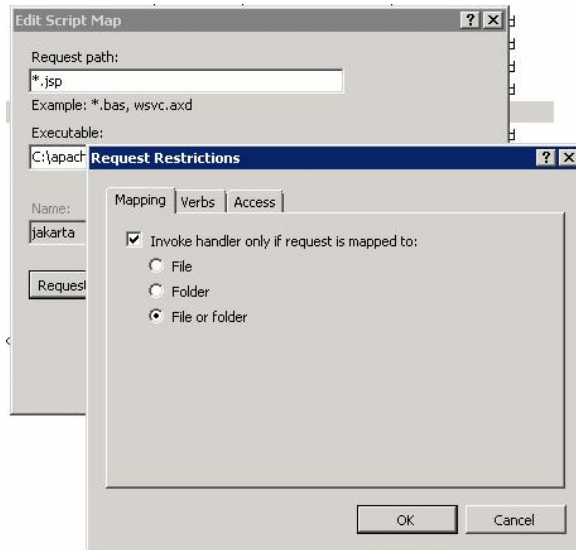
- 7 **[OK]** ボタンをクリックします。
- 8 **[既定の Web サイト]** をクリックし、**[ISAPI フィルタ]** に移動します。
- 9 **[アクション]** リストの**[追加]** をクリックします。
- 10 **[実行可能ファイル]** が「isapi_redirect.dll」（例：
「C:\tomcat\Jakarta\bin\isapi_redirect.dll」）をポイントし、**[フィルタ名]** が **[jakarta]** になっている新しいフィルタを追加します。
- 11 **[OK]** をクリックします。
- 12 **[既定の Web サイト]** を再度クリックします。
- 13 仮想ディレクトリを追加し、**[エイリアス]** を **[jakarta]** に設定します。
[物理パス] を「isapi_redirect.dll」を含むディレクトリをポイントするように設定します。

以下の例を参照してください。



- 14 **[OK]** をクリックします。

- 15 新たに作成された仮想ディレクトリの **[jakarta]** をクリックします。
- 16 **[ハンドラ マッピング]** をダブルクリックします。
- 17 **[アクション]** リストに、**[要求パス]** が **[*.jsp]** で、**[実行可能ファイル]** が **[isapi_redirect.dll]** をポイントし、**[名前]** が **[jakarta]** のスクリプトマップを追加します。
- 18 **[要求の制限]** をクリックします。
次のオプションをオンにします。
 - **[マップ]** タブを開き、**[要求のマップ先が次の場合のみハンドラを呼び出す:]** をオンにして、**[ファイルまたはフォルダ]** を選択します。
 - **[アクセス]** タブを開き、**[実行]** を選択します。
以下の例を参照してください。

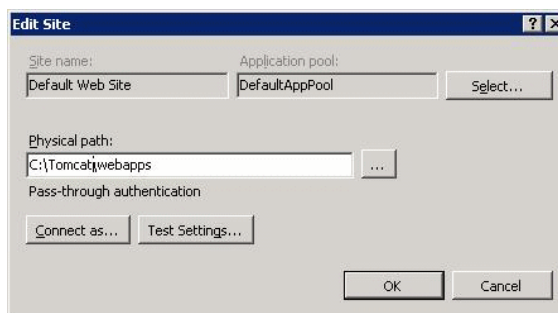


- **[OK]** ボタンをクリックして、ウィンドウを閉じます。
- 19 **[既定の Web サイトのプロパティ]** ウィンドウのままで、**[認証]** タブをクリックします。
 - 20 **[匿名認証]** を無効にし、**[ダイジェスト認証]** および **[Windows 認証]** を有効にします。

以下の例を参照してください。

Name	Status	Response Type
Active Directory Client Certificate Aut...	Disabled	HTTP 401 Challenge
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

- 21 [既定の Web サイトのプロパティ] に移動します。
- 22 [アクション] リストの [基本設定] を選択します。
- 23 [物理パス] を<Tomcatインストールフォルダ>\webappsに設定します。
以下の例を参照してください。



- 24 [OK] ボタンをクリックします
- 25 再度ホストに移動し、[アクション] リストの [再起動] をクリックします。

ポート転送の設定

- 1 テキストエディタを使用して「<Tomcatインストールフォルダ>\conf\server.xml」を開き、次の例のようにTomcat認証を無効にします。

```
<Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" tomcatAuthentication="false" />
```

- 2 ポート転送が正しく動作することを確認するには、Tomcatを再起動して、次のテストを実行します。
- 3 ブラウザを開き、次のURLに移動します。
<http://localhost:8080/AssetManager/>

ヒント:

動作しない場合、Tomcatが起動されていることを確認します。

- 4 TomcatデフォルトアプリケーションとAsset Managerを管理するようJakartaに指示します。

これを行うには、テキストエディタで「<Tomcatインストールフォルダ>\conf\uriworkermmap.properties」を開き、次の行をファイルに追加します。

```
/AssetManager/*=wlb
```

- 5 TomcatとIISを再起動します。
- 6 Windowsユーザとして接続している場合、再び認証することなくAsset Manager Webクライアントにアクセスできるはずです。

トラブルシューティング

次に、いくつかの最も一般的な問題とその考えられる解決策を示します。

- 8080ポートでAsset Managerを開けますが、ポート80で開くことができず、「ページが見つかりません」エラーが発生します。

回答：まず、IISサービスが稼動していることを確認し、Jarkataファイル（「<Tomcatインストールフォルダ

>\Jarkata\conf\uriworkermmap.properties」）を開いて、Asset Managerアプリケーションが追加されているかどうかを確認してください。

- SSOを有効にした後でも、Asset Managerのログインが求められます。

回答：前のすべての手順が正しく完了していることを確認します。書込みモードで「<Tomcatインストールフォルダ>\conf\server.xml」ファイルを開き、次の行のコメント文字（<!--および-->）を削除します。

```
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
```

これにより、Tomcatは強制的に追加ログファイル

「Catalina_log.YYYY-MM-DD.txt」を生成します。

このログファイルを開き、次のパラメータの入力をチェックします。

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: queryString=null
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (IPアドレスをここに入力)
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (IPアドレスをここに入力)
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteUser=<ドメイン名>\<ユーザ名>
```



```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: requestedSessionId=null
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: scheme=http
```

[remoteUser] フィールドが空の場合は、再度Tomcat認証を確認します。「<Tomcatインストールフォルダ>\conf\server.xml」ファイルを開き、Tomcat認証が無効になっていることを確認します。

- 次のエラーが表示されます。

```
Login failed with error "Error: com.peregrine.ac.AmException: Error (12,008): Login 'xxx' unknown."
```

回答：ユーザがデータベース内に存在し、ユーザにAsset Managerにアクセスする承認された役割があるかどうかを確認します。ユーザのログイン名に注意してください。ログイン名にドメイン名が含まれているかどうかを確認するには、「<Tomcatインストールフォルダ>\webapps\AssetManager\WEB-INF\classes\application-context.html」ファイルを開き、[keepDomain] の値をチェックします。ドメイン名を保持する場合、この値を“true”に更新します。それ以外の場合、“false”として更新します。Tomcatを再起動して、もう一度やり直します。

Asset Manager Web クライアント : IIS 7 64ビット上のActive Directoryを使用するシングルサインオン (SSO) を導入する

このセクションは、特にIIS7 64ビットに適用されます。

前提条件

起動前に、以下があることを確認します。

- ネットワーク上で動作可能なActive Directoryの導入
このセクションの手順では、Windows Server 2003 R2で提供されるActive Directoryバージョンを使用しての導入を説明しますが、前バージョンのActive Directoryにも有効です。
- 動作可能なDNS (Domain Name Service) サーバこれをチェックするには、コマンドライン (コマンドプロンプトウィンドウ) で次を入力します。

```
ipconfig
```

次に実行するコマンド：

```
nslookup <DNSサーバのIPアドレス>
```

- アプリケーションサーバとしてTomcatを使用するAsset Manager
- IIS 7 (Windowsで提供されるWebサーバ、Windows Server 2008 64ビットとWindows Vistaでのみサポートされています)

次のコンポーネントオプションがインストールされない場合は、手動で選択します。 [ISAPI フィルタ]、 [ISAPI 拡張]、 および [Windows 認証]。

 **注意:**

IIS 7のインストールの詳細については、次のリンクを参照してください。
<http://learn.iis.net/page.aspx/29/installing-iis7-on-windows-server-2008/>
<http://learn.iis.net/page.aspx/28/installing-iis-70-on-windows-vista/>

Microsoft Windows 64ビット対応のJavaのインストール

64ビット版のJavaのインストール方法とダウンロードリンクの詳細については、<http://java.sun.com/j2se/1.4.2/install-windows-64.html>を参照してください。

インストールしたら、設定する必要があります。

- 1 Windowsデスクトップ上で、 [マイコンピュータ] を右クリックします。
- 2 [プロパティ] を選択し、 [詳細設定] タブ / [環境変数] に移動します。
- 3 [システム環境変数] リストで、 [新規] をクリックします。
- 4 次のプロパティを持つ新規変数を作成します。

変数名 : [JAVA_HOME]

変数値 : [<Javaインストールフォルダ>\jdk1.5.0_19]

 **注意:**

上記のパスには、Java 64ビットをインストールしたフォルダを適応させてください。

- 5 開いている設定ウィンドウのそれぞれで [OK] をクリックして、変更を保存します。

Jakarta ISAPI Redirectorプラグインのインストール

まず、IISとTomcatとの間の接続を確立する必要があります。Apacheには、実装する必要があるISAPI ISAPI (Internet Server Application Programming Interface) Redirectorプラグインが用意されています。

- 1 次のサイトから64ビットのRedirectorをダウンロードします。

http://apache.oregonstate.edu/tomcat/tomcat-connectors/jk/binaries/win64/jk-1.2.28/amd64/isapi_redirect-1.2.28.dll

- 2 「isapi_redirect-1.2.28.dll」 ファイルを「<Tomcatインストールフォルダ>Jakarta\bin」フォルダに配置します。名前を「isapi_redirect.dll」に変更します。

公式ドキュメントの「**Apache Tomcat Connector - Webserver HowTo**」 (http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html) を参照してください。

- 3 テキストエディタを使用して、次の内容を持つ「<Tomcatインストールフォルダ>\Jakarta\conf\workers.properties.minimal」というファイルを作成します。

```
# workers.properties.minimal - # # This file provides minimal jk configuration
properties needed to # connect to Tomcat.# # The workers that jk should
create and work with #

worker.list=wlb,jkstatus

# # Defining a worker named ajp13w and of type ajp13 # Note that the name
and the type do not have to match.# worker.ajp13w.type=ajp13 worker.ajp13w.host=localhost worker.ajp13w.port=8009

# # Defining a load balancer #

worker.wlb.type=lb worker.wlb.balance_workers=ajp13w

# # Define status worker #

worker.jkstatus.type=status
```

- 4 テキストエディタを使用して、次の内容を持つ「<Tomcatインストールフォルダ>\Jakarta\conf\uriworkermap.properties」というファイルを作成します。

```
# uriworkermap.properties - IIS # # This file provides sample mappings for
example wlb # worker defined in workermap.properties.minimal # The general
syntax for this file is: # [URL]=[Worker name]

/AssetManager/*=wlb

# # Mount jkstatus to /jkmanager # For production servers you will need to
secure the access to the /jkmanager url # /jkmanager=jkstatus
```

- 5 RedirectorをWindowsレジストリに追加します。
- a テキストエディタを使用して、拡張子が「.reg」で、次の内容を持つファイルを作成します。

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi Redirector\1.0] @="" "extension_uri"="/jakarta/isapi_redirect.dll" "log_file"="C:\\Tomcat\\Jakarta\\log\\isapi_redirect.log" "log_level"="info" "worker_file"="C:\\Tomcat\\Jakarta\\conf\\workers.properties.minimal" "worker_mount_file"="C:\\Tomcat\\Jakarta\\conf\\uriworkermap.properties"
```



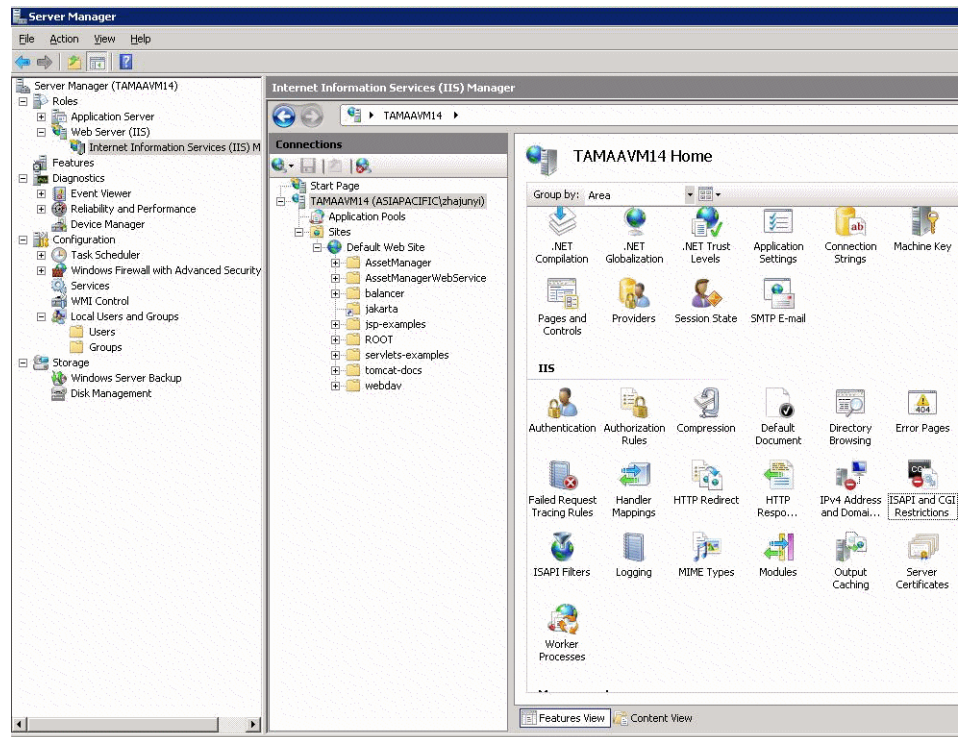
注意:

上記のパスには実際のTomcatインストールフォルダを適応させてください。

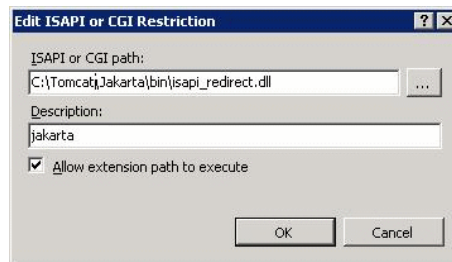
- b ファイルをダブルクリックして、レジストリにインポートします。
- c レジストリエディタを開き（コマンドラインで**regedit**を実行）、**「HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi Redirector\1.0」**を展開して、**Jakarta**が正常に登録されていることを確認します。

IISの設定

- 1 IISがインストールされているサーバのデスクトップで、**「マイコンピュータ」**を右クリックします。
- 2 **「管理」**をクリックします。
- 3 以下に示す例のように、**「役割／Web サーバー (IIS)／インターネット インフォメーション サービス (IIS) マネージャ」**を展開します。

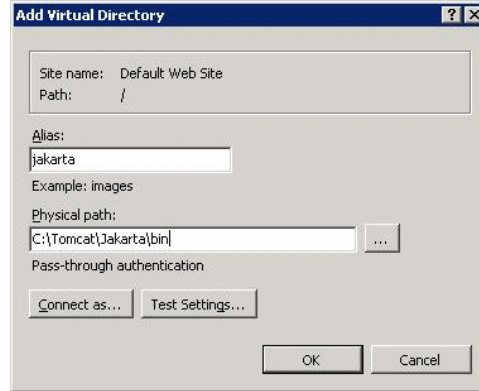


- 4 [インターネットインフォメーションサービス (IIS) マネージャ] パネルで、ホストを選択し、[ISAPI および CGI の制限] をダブルクリックします。
- 5 [アクション] リストの [追加] をクリックします。
- 6 [ISAPI または CGI パス] を「isapi_redirect.dll」ファイル（例：「C:\tomcat\Jakarta\bin\isapi_redirect.dll」）をポイントするように設定し、[説明] を [jakarta] に設定します。
[拡張パスの実行を許可する] をオンにします。
以下の例を参照してください。



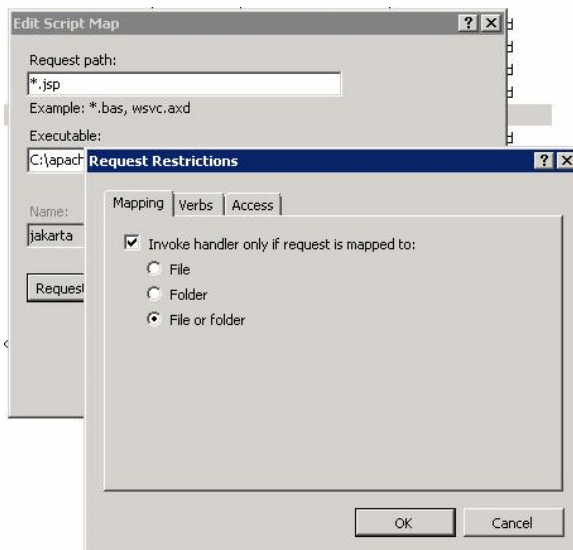
- 7 [OK] ボタンをクリックします。
- 8 [既定の Web サイト] をクリックし、[ISAPI フィルタ] に移動します。
- 9 [アクション] リストの [追加] をクリックします。
- 10 [実行可能ファイル] が「isapi_redirect.dll」（例：「C:\tomcat\Jakarta\bin\isapi_redirect.dll」）をポイントし、[フィルタ名] が [jakarta] になっている新しいフィルタを追加します。
- 11 [OK] をクリックします。
- 12 [既定の Web サイト] を再度クリックします。
- 13 仮想ディレクトリを追加し、[エイリアス] を [jakarta] に設定します。
[物理パス] を「isapi_redirect.dll」を含むディレクトリをポイントするように設定します。

以下の例を参照してください。

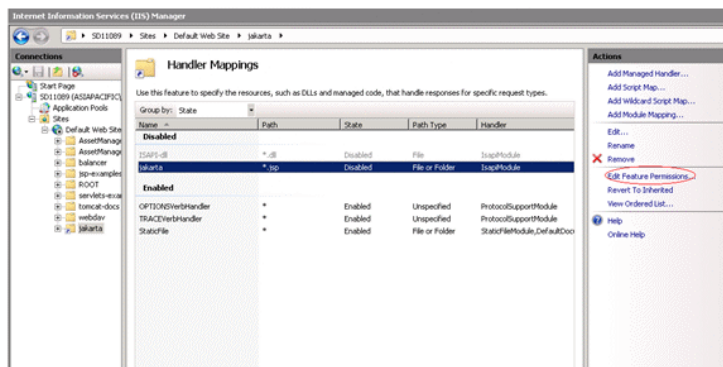


- 14 **[OK]** をクリックします。
- 15 新たに作成された仮想ディレクトリの **[jakarta]** をクリックします。
- 16 **[ハンドラ マッピング]** をダブルクリックします。
- 17 **[アクション]** リストに、**[要求パス]** が **[*.jsp]** で、**[実行可能ファイル]** が **[isapi_redirect.dll]** をポイントし、**[名前]** が **[jakarta]** のスクリプトマップを追加します。
- 18 **[要求の制限]** をクリックします。
次のオプションをオンにします。
 - **[マップ]** タブを開き、**[要求のマップ先が次の場合のみハンドラを呼び出す:]** をオンにして、**[ファイルまたはフォルダ]** を選択します。
 - **[アクセス]** タブを開き、**[実行]** を選択します。

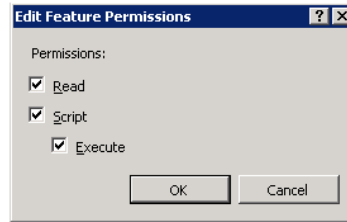
以下の例を参照してください。



- **[OK]** ボタンをクリックして、ウィンドウを閉じます。
- 19 新しいJakartaハンドラマッピングを選択します。右側のパネルで、以下の例に示すように、**【機能のアクセス許可を編集】**を選択します。



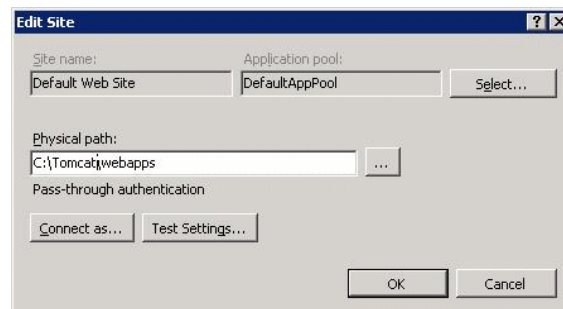
- 20 **【実行】** を選択し、**【OK】** をクリックします。



- 21 **【既定の Web サイトのプロパティ】** ウィンドウのまま、**【認証】** タブをクリックします。
- 22 **【匿名認証】** を無効にし、**【ダイジェスト認証】** および **【Windows 認証】** を有効にします。
以下の例を参照してください。

Name	Status	Response Type
Anonymous Authentication	Disabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Enabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Enabled	HTTP 401 Challenge

- 23 **【既定の Web サイトのプロパティ】** に移動します。
- 24 **【アクション】** リストの **【基本設定】** を選択します。
- 25 **【物理パス】** を **<Tomcatインストールフォルダ>\webapps** に設定します。
以下の例を参照してください。



- 26 **[OK]** ボタンをクリックします
- 27 再度ホストに移動し、**[アクション]** リストの**[再起動]** をクリックします。

ポート転送の設定

- 1 テキストエディタを使用して「<Tomcatインストールフォルダ>\conf\server.xml」を開き、次の例のようにTomcat認証を無効にします。

```
<Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" tomcatAuthentication="false" />
```

- 2 ポート転送が正しく動作することを確認するには、Tomcatを再起動して、次のテストを実行します。
- 3 ブラウザを開き、次のURLに移動します。

<http://localhost:8080/AssetManager/>



ヒント:

動作しない場合、Tomcatが起動されていることを確認します。

- 4 TomcatデフォルトアプリケーションとAsset Managerを管理するようJakartaに指示します。

これを行うには、テキストエディタで「<Tomcatインストールフォルダ>\conf\uriworkermmap.properties」を開き、次の行をファイルに追加します。

```
/AssetManager/*=wlb
```

- 5 TomcatとIISを再起動します。
- 6 Windowsユーザとして接続している場合、再び認証することなくAsset Manager Webクライアントにアクセスできるはずです。

トラブルシューティング

次に、いくつかの最も一般的な問題とその考えられる解決策を示します。

- 8080ポートでAsset Managerを開けますが、ポート80で開くことができず、「ページが見つかりません」エラーが発生します。

回答：まず、IISサービスが稼働していることを確認し、Jakartaファイル（「<Tomcatインストールフォルダ

>\Jakarta\conf\uriworkermmap.properties」）を開いて、Asset Managerアプリケーションが追加されているかどうかを確認してください。

- SSOを有効にした後でも、Asset Managerのログインが求められます。

回答：前のすべての手順が正しく完了していることを確認します。書込みモードで「<Tomcatインストールフォルダ>\conf\server.xml」ファイルを開き、次の行のコメント文字（<!--および-->）を削除します。

```
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
```


これにより、Tomcatは強制的に追加ログファイル
「Catalina_log.YYYY-MM-DD.txt」を生成します。

このログファイルを開き、次のパラメータの入力をチェックします。

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: queryString=null
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (IPアドレスをここに入力)
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (IPアドレスをここに入力)
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteUser=<ドメイン名>\<ユーザ名>
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: requestedSessionId=null
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: scheme=http
```

[remoteUser] フィールドが空の場合は、再度Tomcat認証を確認します。
「<Tomcatインストールフォルダ>\conf\server.xml」ファイルを開き、
Tomcat認証が無効になっていることを確認します。

- 次のエラーが表示されます。

```
Login failed with error "Error: com.peregrine.ac.AmException: Error (12,008): Login 'xxx' unknown."
```

回答：ユーザがデータベース内に存在し、ユーザにAsset Managerにアクセスする承認された役割があるかどうかを確認します。ユーザのログイン名に注意してください。ログイン名にドメイン名が含まれているかどうかを確認するには、「<Tomcatインストールフォルダ

>\webapps\AssetManager\WEB-INF\classes\application-context.html」ファイルを開き、**[keepDomain]** の値をチェックします。ドメイン名を保持する場合、この値を“true”に更新します。それ以外の場合、“false”として更新します。Tomcatを再起動して、もう一度やり直します。

Asset Manager Webクライアント：SiteMinderやWebSEALなどのID管理ツールを使用するシングルサインオン（SSO）を導入する

- ▶ Asset Manager Webクライアント：Lightweightシングルサインオン（LW-SSO）を使用するユーザ認証を共有する [P. 218]

Asset Manager Webクライアント : Lightweightシングルサインオン (LW-SSO) を使用するユーザ認証を共有する

概要

Lightweightシングルサインオン (LW-SSO) フレームワークが有効な場合、ユーザはセッション中 (1つのHP Software BTOアプリケーション) に一度だけログインする必要があります。ユーザのログイン情報は、同じドメインに導入されている他のHP Software BTOアプリケーションに伝達されます。

LW-SSOでは、Lightweightテクノロジーを使用して、HP Software BTO製品に関するWebおよびWebサービスの両方のSSOの問題を解決します。この際、追加のインフラストラクチャをインストールする必要はありません。

LW-SSOの使用例

LW-SSOを使用すると、次の使用例に対応できます。

- 1 Webのシングルサインオン** : ユーザが資格情報を使用して1つのBTOアプリケーションにログインし、同じブラウザで2番目のBTOアプリケーションにアクセスします。ユーザは既にログインしているため、LW-SSOによって、資格情報をインタラクティブに再認証せずに、2番目のアプリケーションに参入できます。
- 2 Webサービスのシングルサインオン** : 1つのBTOアプリケーションに対するユーザ要求中に、2番目のBTOアプリケーションに対する呼び出しを実行する必要があります。LW-SSOによって、ユーザデータが2番目のBTOアプリケーションに転送されます。
- 3 Webサービスの呼び出し** : Webサービスの呼び出しは、非インタラクティブユーザの情報を使用して、最初のBTOアプリケーションから2番目のBTOアプリケーションに対して行われます。LW-SSOによって、ユーザデータが2番目のBTOアプリケーションに転送されます。
- 4 LW-SSOはアブストラクションレイヤとして使用されます** (BTOアプリケーションと外部ID管理 (IdM) /SSOツール間)。これで、IdM/SSOツールおよび開発コスト全体との製品統合が容易になります。

主な機能

- LW-SSOはそれ自体ではユーザ認証を提供または調停しません。各アプリケーションがユーザの認証を担当します。
- LW-SSOは、ユーザ情報をApplication1からApplication2に渡すだけです (共有cookies経由で)。
- このため、LW-SSOは他のIdMソリューションの置換ではありません。
- BTO Application1およびBTO Application2は、同じユーザストレージまたは別のユーザストレージを使用できます。ただし、別のユーザストレージを使用する場合、ユーザ名を同期できる必要があります。この同期は、LW-SSOフレームワークでは処理されないためです。

- LW-SSOは対称ソリューションです。つまり、ユーザは、最初にApplication1にログインしてから透過的にApplication2へのリンクを開いたり、逆に、最初にApplication2にログインしてからApplication1へのリンクを開いたりできます。

LW-SSOの設定

仮設定

Asset Manager Web Serviceの展開には、基本的なLW-SSOパラメータを設定する管理画面が含まれています。

次のURLに接続します。

`http://${AMWebServiceServer}:${AMWebServicePort}/AssetManagerWebServiceContext/ssoConfig.jsp`

例：

`http://localhost:8081/AssetManagerWebService/ssoConfig.jsp`

注意:

Tomcatなどのアプリケーションサーバが正しく設定され実行されている必要があります。

次の例のように、サインオン画面が表示されます。



以下のテーブルの説明に従って、フィールドに入力します。

注意:

この画面では、管理者権限が必要になります。

フィールド/コントロール名	用途
Enable LW-SSO framework (LW-SSO フレームワークを有効にする)	このオプションをオンにすると、次のフィールドで定義されるターゲットドメインで、LW-SSOフレームワークが有効になります。

フィールド/コントローラ名 用途

ドメイン名	互換バージョンのLW-SSOを購読しているホストされているHP Software BTOアプリケーション間で、ユーザのログイン情報が伝達されるドメインです。これはアプリケーションサーバの完全なDNS名である必要があります。例：「asiapacific.mycompany.net」
Shared encryption key (共有暗号化キー)	LW-SSO cookiesの暗号化キーとして使用されるシークレットテキスト文字列です。これは initstring パラメータの伝達に使用されます。LW-SSOを購読している他のHP Software BTOアプリケーションは、同じ initstring を使用して共通のcookiesを解読および共有できるように設定する必要があります。LW-SSOセキュリティの考慮事項 [P. 220]も参照してください。
Request header key containing the user name (ユーザ名を含むヘッダーキーを要求する)	HTTPヘッダー応答では、指定されたユーザ名（これはSiteMinderなどのSSOソリューションのサポートに使用できます）を参照する必要があります。ラベルが示すとおり、ヘッダーキーの名前のみ指定する必要があります。

追加の設定を行うには、LW-SSO APIを手動で変更します。

共有cookiesを無効にする

LW-SSOが有効になっている場合、デフォルトでは、LW-SSOを購読している1つのアプリケーションからログアウトすると、同じドメインでホストされている他のすべての購読中のHP Software BTOアプリケーションからログアウトされます。これを防止し、アプリケーションごとに明示的なログアウトを要求するには、次のエントリを

「**AssetManager.ear/AssetManager.war/WEB-INF/classes**」にある「**lwssofmconf.xml**」ファイルに追加します。

```
<logoutURLs> <url> .*cwc/logoutcleanup.jsp.* </url> </logoutURLs>
```

LW-SSOセキュリティの考慮事項



注意:

特に求められない限り、LW-SSOは無効にする必要があります。

LW-SSOセキュリティでの機密の**initString**パラメータ

LW-SSOでは、対称暗号化を使用して、LW-SSOトークンを検証および作成します。**initString**またはシークレットフレーズ設定パラメータは、暗号化キーの初期化に使用されます。あるアプリケーションがトークンを作成し、同じ**initString**パラメータを使用する各アプリケーションがトークンを検証します。



注意:

initStringに関する注意:

- **initString**パラメータを設定せずにLW-SSOを使用することはできません。
- **initString**パラメータは機密情報であるため、公開、送受信、永続性の観点からそれに応じた取り扱いをする必要があります。
- **initString**は、LW-SSOを使用してお互いに統合しているアプリケーション間でのみ共有する必要があります。
- **initString**の最小の長さは12文字です。

認証セキュリティレベルの決定方法

最も弱い認証フレームワークを使用し、他の統合アプリケーションが信頼できるLW-SSOトークンを発行するアプリケーションによって、すべてのアプリケーションの認証セキュリティのレベルが決定します。

強力で安全な認証フレームワークを使用しているアプリケーションにのみ、LW-SSOトークンを発行させることをお勧めします。

対称暗号化の意味

LW-SSOでは、LW-SSOトークンを発行および検証するために、対称暗号法を使用しています。

このため、LW-SSOを使用しているアプリケーションは、同じ**initString**を共有している他のすべてのアプリケーションによって信頼されるトークンを発行できます。

この潜在的なリスクは、**initString**を共有しているアプリケーションが信頼できない場所に存在するか、そのような場所からアクセスできる場合に該当します。

ユーザマッピング (同期)

LW-SSOでは、統合アプリケーション間のユーザマッピングを確認しません。このため、統合アプリケーションでは、ユーザマッピングをモニタする必要があります。すべての統合アプリケーション間で同じユーザレジストリ (LDAP/ADとして) を共有することをお勧めします。

ユーザのマップに失敗すると、セキュリティ違反およびネガティブなアプリケーション動作が発生する可能性があります。例えば、同じユーザ名がさまざまなアプリケーションの異なる実際のユーザに割り当てられることがあります。

さらに、ユーザがあるアプリケーション (AppA) にログオンしてから、コンテナまたはアプリケーション認証を使用する2番目のアプリケーション (AppB) にアクセスする場合、ユーザのマップに失敗すると、ユーザは手動でAppBにログオンしてユーザ名を入力する必要があります。ユーザがAppAへのログオンに使用したものと異なるユーザ名を入力すると、次の予期しない動作が発生する可能性があります。ユーザがその後AppAまたはAppBから3番目のアプリケーション

(AppC) にアクセスする場合、それぞれAppAまたはAppBへのログオンに使用したユーザ名を使用してアクセスを行います。

認証に使用されるIdentity Manager

Identity Manager内のすべての保護されていないリソースは、LW-SSO設定では **nonsecureURLs** としてリスクしておく必要があります。

LW-SSOの制限事項

ドメイン名

LW-SSOを活用するには、ユーザはログインURLの完全修飾ドメイン名 (FQDN) (例: <http://fhedaraly.emea.hpqcorp.net:8080/AssetManager>) を使用して、Asset ManagerなどのBTOアプリケーションにアクセスする必要があります。

LW-SSOでは、IPアドレス付きのURL (例: <http://16.59.45.143:8080/WebApp>) をサポートしていません。

LW-SSOでは、ドメインなしのURL (例: <http://flood:8080/WebApp>) をサポートしていません。

LW-SSOフレームワークの統合

アプリケーションは、事前にLW-SSOフレームワークと統合されている場合のみ、LW-SSO機能を活用できます。

複数ドメインのサポートの制限

- 複数ドメイン機能は、HTTP参照元に基づいています。このため、LW-SSOでは、あるアプリケーションから別のアプリケーションへのリンクがサポートされ、両方のアプリケーションが同じドメインにある場合を除き、ブラウザウィンドウへのURLの入力はサポートされません。
- HTTP POSTを使用したクロスドメインリンクはサポートされていません。
複数ドメイン機能では、2番目のアプリケーションに対する最初のHTTP POST要求はサポートされません (HTTP GET要求のみサポートされます)。例えば、アプリケーションに2番目のアプリケーションに対するHTTPリンクがある場合、HTTP GET要求はサポートされますが、HTTP FORM要求はサポートされません。最初の要求以降のすべての要求は、HTTP POSTまたはHTTP GETのいずれかとなります。
- LW-SSOトークンサイズの制限
LW-SSOがあるドメイン内のあるアプリケーションから別のドメイン内の別のアプリケーションに転送できる情報のサイズは、15個のグループ/役割/属性 (各項目の平均の長さは15文字) に制限されます。
- 複数ドメインでの保護 (HTTPS) から非保護 (HTTP) ページへのリンクシナリオ:
複数ドメイン機能は、保護 (HTTPS) から非保護 (HTTP) ページへのリンクでは、動作しません。これは、保護から非保護リソースへのリンク時に参

照元ヘッダーが送信されないというブラウザの制限です。例については、<http://support.microsoft.com/support/kb/articles/Q178/0/66.ASP> を参照してください。

- Internet Explorerでの「サードパーティ」製cookies

IE6で、Microsoftは“Platform for Privacy Preferences (P3P) Project”をサポートするモジュールを追加しました。これは、「サードパーティ」ドメインからのcookiesが、デフォルトで「インターネット」セキュリティゾーンでブロックされることを意味します。つまり、セッションcookiesは、IEでも「サードパーティ」製cookiesと見なされるため、ブロックされ、LW-SSOが動作を停止します。

詳細については、<http://support.microsoft.com/kb/323752/en-us> を参照してください。

 **注意:**

考えられる解決策は、起動されるアプリケーション（またはDNSドメインサブセットを*.mydomain.comとして）をお使いのコンピュータの「イントラネット」/「信頼済み」ゾーンに追加することです（Internet Explorerメニューオプションの「ツール/インターネット オプション/セキュリティ/ローカルイントラネット/サイト/詳細設定」を使用）。これでcookiesが受け入れられます。

 **警告:**

重要: LW-SSOセッションcookieは、このパラメータでブロックされる「サードパーティ」製アプリケーションcookiesの唯一のものです。

SAML2トークン関連の制限

LW-SSOと統合されたJavaベースのアプリケーションを使用して、これもLW-SSOと統合された別のJavaベースのアプリケーションにアクセスする場合、SAML2トークンを使用しないでください。この場合、SAML2トークンを使用すると、予期しない動作が発生する可能性があります。代わりに、LW-SSOトークンを使用してください。

いずれかのアプリケーションがLW-SSOと統合されていない場合にのみ、SAML2トークンを使用する必要があります。

SAML2トークンの使用時には、ログアウト機能はサポートされません。このため、SAML2トークンを使用して2番目のアプリケーションにアクセスする場合、最初のアプリケーションからログアウトするユーザは、2番目のアプリケーションからログアウトされません。

SAML2トークンの有効期限は、アプリケーションのセッション管理には反映されません。

このため、SAML2トークンを使用して2番目のアプリケーションにアクセスする場合、各アプリケーションのセッション管理は、相互に独立して処理されます。アプリケーションがTomcatサーバまたはJBossサーバのContainer Managed Securityを使用している場合、**AutoCookieCreation**機能を有効にして正しくSAML2をサポートすることをお勧めします。**AutoCookieCreation**が無効になっている場合、次の規則に準拠する必要があります。それ以外の場合、LW-SSOでは、SAML2で要求を正しく認証できません。

- 最初のリソース（JSPページなど）でHTTPセッションを作成する必要があります。
- 最初のリソースに保護リソース（JPEG画像など）を含めない。
- HTTPセッションは、ユーザがブラウザを閉じるまで消去しない。

Tomcat関連の制限

- TomcatのJAAS Realmはサポートされません。
- Tomcatディレクトリでのスペースの使用はサポートされません。
Tomcatインストールパス（フォルダ）にスペース（Program Filesなど）が含まれていて、LW-SSO設定ファイルが「common\classes」というTomcatフォルダにある場合、LW-SSOを使用できません。

ロードバランサの設定

LW-SSOで導入されるロードバランサは、スティッキーセッションを使用するように設定する必要があります。

LW-SSOに関する注意

LW-SSOトークンの有効期限

LW-SSOトークンの有効期限の値によって、アプリケーションのセッションの有効期間が決まります。このため、有効期限の値は、少なくともアプリケーションセッションの有効期限の値と同じである必要があります。

LW-SSOトークンの有効期限の推奨される設定

LW-SSOを使用する各アプリケーションでは、トークンの有効期限を設定する必要があります。推奨される値は60分です。高度なセキュリティを必要としないアプリケーションでは、300分の値を設定できます。

GMT時間の同期

LW-SSO統合に参加するすべてのアプリケーションは、最大15分の差で同じGMT時間を使用する必要があります。

protectedDomains設定

複数ドメイン機能では、LW-SSO統合に参加するすべてのアプリケーションで、異なるDNSドメイン内のアプリケーションと統合する必要がある場合、

protectedDomains設定を行う必要があります。また、設定のlwssso要素に正しいドメインを追加する必要もあります。

URL機能用のSecurityTokenの取得

他のアプリケーションからURL用のSecurityTokenとして送信される情報を受信するには、ホストアプリケーションで設定のlwssso要素に正しいドメインを設定する必要があります。

nonsecureURLs設定

アプリケーションのログインおよびログアウトURLをnonsecureURLs設定として設定しないことをお勧めします。このように設定すると、予期しない動作が発生する可能性があります。

LW-SSOの既知の問題

IdMセキュリティと発信Webサービス

LW-SSOと統合されていて、IdMセキュリティを発信Webサービスのセキュリティトークンとして使用しているアプリケーションは、UI用のLW-SSOセキュリティトークンを作成または発行できません（自動的に作成されたか、それ以外の場合）。

この場合、LW-SSO設定には次の値が含まれている必要があります。

```
enableAutoCreation = "false"
```

```
enableCookieCreation = "false"
```

複数の属性値の未サポート

LW-SSOセキュリティコンテキストでは、属性名ごとに1つの属性値しかサポートされません。このため、SAML2トークンによって同じ属性名に複数の値が送信された場合、ただ1つの値しかLW-SSOフレームワークによって受け入れられません。

同様に、IdMトークンが同じ属性名に複数の値を送信できるように設定されている場合、ただ1つの値しかLW-SSOフレームワークによって受け入れられません。

Internet Explorer 7での複数ドメインのログアウト機能

複数ドメインのログアウト機能は、Internet Explorer 7の使用時、およびアプリケーションがログアウト手順で4つ以上の連続したHTTP 302リダイレクト動詞を呼び出す場合、失敗する可能性があります。このような場合、Internet Explorer 7では、HTTP 302リダイレクト応答の処理を誤り、代わりに「Internet Explorer 7ではこのページは表示できません」というエラーページが表示されます。

回避策としては、可能な限り、ログアウトシーケンスでのアプリケーションリダイレクトコマンドの数を減らすことをお勧めします。

LW-SSOのシステム要件

LW-SSOバージョン2.1.1では、ブラウザと関連テクノロジーの次のバージョンと設定がサポートされています。

名前	バージョン	コメント
Java	1.5以上	
HTTP Servlets API	2.1以上	
Internet Explorer	6.0以上	ブラウザでは、HTTPセッションcookieとHTTP 302リダイレクト機能を有効にする必要があります。
FireFox	2.0以上	ブラウザでは、HTTPセッションcookieとHTTP 302リダイレクト機能を有効にする必要があります。
JBoss認証	JBoss 4.0.3 JBoss 4.3.0	
Tomcat認証	Standalone Tomcat 5.0.28 Standalone Tomcat 5.5.20	
Acegi認証	Acegi 0.9.0 Acegi 1.0.4	
Webサービスエンジン	Axis 1 - 1.4 Axis 2 - 1.2 JAX-WS-RI 2.1.1	

トラブルシューティング

次の表に、LW-SSOに関して考えられる問題、および考えられる原因と推奨される解決策を示します。

LW-SSO関連の使用例

問題	考えられる原因	考えられる解決策
BTOアプリケーションのWebクライアントを開くことができない。例えば、追加のWebアプリケーションをTomcatに導入後など。	Tomcatレパートリーが正しく更新されていない。	例： 「C:\Tomcat5\webapps\AssetManager」を削除します。このフォルダは、次回Webクライアントにアクセスを試みると、動的に再作成されます。 「Tomcat<xx>\webapps」下のフォルダで、開かないアプリケーションに対応するフォルダを手動で削除します。

問題	考えられる原因	考えられる解決策
BTO Webアプリケーションへのログイン後に、LW-SSO cookieが作成されない。	ドメインが設定のLW-SSO要素で正しく定義されていない。	設定のLW-SSO要素で定義されているドメインがアプリケーションのドメインと等しいことを確認します。
	パラメータとして enableSSO 関数に渡されるドメインが正しくない。	パラメータとして enableSSO 関数に渡されるドメインがアプリケーションのドメインと等しいことを確認します。
	ログインURLの完全修飾ドメイン名 (FQDN) (例: http://localhost:8080/WebApp) を使用して、アプリケーションにアクセスできなかった。	ログインURLの完全修飾ドメイン名 (FQDN) (例: http://foodmercuryglobal8080/WebApp) を使用して、アプリケーションにアクセスできることを確認します。
LW-SSOでAutoCookieCreation機能用のcookieを作成できない。	ドメインが設定のLW-SSO要素で正しく定義されていない。	設定のLW-SSO要素で定義されているドメインがアプリケーションのドメインと等しいことを確認します。
LW-SSOトークンが検証されていない。	2つのアプリケーションで、設定の crypto 要素に異なる initString パラメータ (または他の crypto パラメータ) がある。	両方のアプリケーションで同じ initString を使用します (LW-SSO creation 要素の他のすべての crypto に加えて)。
	2つのアプリケーション間のGMT時間の差が15分を超えている。	LW-SSO統合に参加しているすべてのアプリケーションが、最大15分の差で同じGMT時間に設定されていることを確認します。

問題	考えられる原因	考えられる解決策
LW-SSOで、複数ドメイン環境のLW-SSOトークンを検証できない。	いずれかのアプリケーションの設定で、ドメインがLW-SSO要素で正しく定義されていない。	アプリケーションの設定のLW-SSO要素で定義されているドメインは、使用中の実際のドメインに基づいてアプリケーションのドメインと同じである必要があります。
	いずれかのアプリケーションの設定で、ドメインが <code>protectedDomains</code> リストで正しく定義されていない。	アプリケーションのすべての設定の <code>protectedDomains</code> リストにあるドメインが正しく定義されていることを確認します。
	LW-SSOセッションcookieが、Internet Explorer 6.0、7.0、または8.0のブラウザ使用時にブロック/拒否される。	すべてのLW-SSOサーバを、コンピュータ上のInternet Explorerセキュリティゾーン内の「イントラネット」/「信頼済み」ゾーン（[ツール/インターネットオプション/セキュリティ/ローカルイントラネット/サイト/詳細設定]）に追加します。これですべてのcookiesが受け入れられるようになります。
	一部のアプリケーションで、設定のcrypto要素に異なる <code>initString</code> パラメータ（または他のcryptoパラメータ）がある。	すべてのアプリケーションで同じ <code>initString</code> を使用します（LW-SSO creation要素の他のすべてのcryptoに加えて）。
	一部のアプリケーションに15分を超えるGMT時間の差がある。	LW-SSO統合に参加しているすべてのアプリケーションが、最大15分の差で同じGMT時間に設定されていることを確認します。
	複数ドメインリンクが、保護（HTTPS）から非保護（HTTP）リソースに移動している。	あるドメインから別のドメインにリンク/クロスしている場合、最初のリンク/クロス要求がある保護リソース（HTTPS）から別の保護（HTTPS）リソースに移動していることを確認します。

SAML2関連の使用例

問題	考えられる原因	考えられる解決策
LW-SSOでSAML2トークンを発行できない。	SAML2作成用のkeystore設定が無効で、有効なキーストアをポイントしていない可能性が高い。	SAML2作成用のkeystore設定が有効で、有効なキーストアをポイントしていることを確認します。

問題	考えられる原因	考えられる解決策
LW-SSOでSAML2トークンを検証できない。	<p>SAML2作成用の <code>privateKeyAlias</code> または <code>privateKeyPassword</code> 設定が有効でない。有効なプライベートキーをポイントしていない可能性がある。</p> <p>SAML2検証用の <code>keystore</code> 設定が有効でない。有効なキーストアをポイントしていない可能性が高い。</p> <p>署名に使用される証明書が、必要なパブリックキーエイリアスで設定されたキーストアにインポートされていない。</p> <p>注意：パブリックキーエイリアスは、SAML2検証設定で設定するか、発行者がSAML2トークンで渡すことができます。</p>	<p>SAML2作成用の <code>privateKeyAlias</code> または <code>privateKeyPassword</code> 設定が有効で、有効なプライベートキーをポイントしていることを確認します。</p> <p>SAML2検証用の <code>keystore</code> 設定が有効で、有効なキーストアをポイントしていることを確認します。</p> <p>署名に使用される証明書が、必要なパブリックキーエイリアスで設定されたキーストアにインポートされていることを確認します。</p>
LW-SSOですべての役割またはグループを受信できない。	<p>SAML2設定で定義されている要素の <code>roleAttributeName</code> または <code>groupAttributeName</code> が、アプリケーションの <code>configurations</code> 要素で設定されているものと異なる。</p>	<p>要素の <code>roleAttributeName</code> または <code>groupAttributeName</code> が、アプリケーションのすべての <code>configurations</code> と SAML2設定で同様に定義されていることを確認します。</p>

13 Asset Manager Automated Process Manager

この章では、期限（アラーム、購入依頼の承認、在庫の再発注など）とアクションの自動的なトリガ（催促メッセージの自動発信など）を管理する方法について説明します。

管理者は、Asset ManagerではなくAsset Manager Automated Process Managerプログラムを使って期限のモニタとアクションの自動的なトリガを管理します。

Asset Manager Automated Process Managerの概要

Asset Managerパッケージには、期限のモニタとアクションを自動的にトリガするシステムが含まれています。このプログラムはAsset Manager Automated Process Managerと呼ばれ、Asset Managerから独立して機能します。

Asset Manager Automated Process Managerは、指定されたデータベースについて、次のすべての期日を自動的にモニタします。

- アラーム（契約の満了日など）
- ヘルプデスク特有の期限（チケットの割り当てのタイムアウト、解決期限など）
- 購入依頼の承認
- 在庫明細の再発注時の在庫数（発注点）
- 資産賃貸料と契約賃貸料の計算
- リース契約の損失額の計算
- コストセンタに関連する経費明細の分割操作

- 履歴項目の検証
- ワークフローの期限
- 新しいワークフロー実行グループの検索
- ワークフロー規則の実行
- タイムゾーンの検証

期限になると、Asset Manager Automated Process Managerは内部メッセージシステムを介して、Asset Managerデータベース内で催促メッセージを発信するなどのアクションを実行します。必要に応じて、契約賃貸料、リース契約の損失額などを計算します。

 **警告:**

Asset Manager Automated Process Managerを終了すると、すべての自動モニタ機能が一時停止します。

Asset Manager Automated Process Managerを複数のマシンで実行することも可能です。複数のマシンで実行すると、モニタする期限や実行するタスクをそれぞれのAsset Manager Automated Process Managerで分担できます。これにより、Asset Manager Automated Process Managerの性能が向上します。

 **注意:**

1つのタスクは単独のAsset Manager Automated Process Managerインスタンスでのみ実行されます。

データベースへの接続と同じログインを使って接続できます。このログインには管理者権限が必要です。

Asset Manager Automated Process Managerを実行する

推奨する画像の詳細

Asset Manager Automated Process Managerは、頻繁にデータベースにアクセスする必要があります。データベースはネットワーク上にあります。

- データベースに高速接続できるワークステーションがある場合は、そのワークステーションからAsset Manager Automated Process Managerを実行します。すべてのユーザに関するモニタを実行します。
- 低速リンクでしかデータベースにアクセスできない場合で、しかもWindowsでサーバを稼動している場合は、サーバ上で直接Asset Manager Automated Process Managerを実行できます。

データベースの構造を変更する場合

Asset Manager Application Designer、または [オブジェクトの設定] ショートカットメニューを使ってデータベース構造を変更する場合は、一旦Asset Manager Automated Process Managerをデータベースから切断し、作業が終わってから再接続する必要があります。

Asset Manager Automated Process Managerを起動する

Asset Manager Automated Process Managerを手動で起動する

Windowsの [スタート] メニューのプログラムから、またはAsset ManagerプログラムグループからAsset Manager Automated Process Managerプログラムを起動します。



警告:

Asset Manager Automated Process Managerをデータベースから切断すると、すべてのモニタ機能とアクションの自動的なトリガ機能が一時停止します。Asset Managerからデータベースに接続すると、Asset ManagerにはAsset Manager Automated Process Managerが過去1時間以上データベースにアクセスしていないことを表す警告メッセージが表示されます。

Asset Manager Automated Process Managerをサービスモードで自動的に起動する

Asset Manager Automated Process Managerをサービスモードで起動するには：

- 1 Asset Manager Automated Process Managerを手動で起動する
- 2 データベースへの接続を選択し、[この接続をサービスモードで使用] チェックボックスをオンにします。
- 3 Asset Manager Automated Process Managerを終了します。
- 4 Windowsのコントロールパネルで、Asset Manager Automated Process Managerを選択し、スタートアップの種類を「自動」に設定します。

DOSコマンドプロンプトからAsset Manager Serverを起動する

次のコマンドを使って、Asset Manager Automated Process Managerの起動を自動化できます。

```
amsrv -cnx:<接続名> -login:<ログイン> -password:<ログインのパスワード>
```



警告:

ログインには、Asset Manager管理者のログイン（「Admin」または管理者権限を有するユーザのログイン）を入力します。

<~>で表されている文字列にスペースを含めることは出来ません。

例：

```
amsrv32 -cnx:Base -login:Gerald -password:Password
```

このコマンドは、バッチファイルに挿入できます。


Windows上でAsset Manager Automated Process Managerを手動で実行する

Asset Manager Automated Process Manager をデータベースに接続する

Asset Manager Automated Process Managerからデータベースに接続できるのは管理者だけです。これは、"Admin"ユーザまたは管理者権限のあるユーザです。


適切なログインとパスワードを入力する必要があります。Asset Manager Automated Process Managerをサービスモード（NTのサービスとして）で実行している場合に、選択した接続をデフォルトの接続として使うには、**【この接続をサービスモードで使用】** チェックボックスをオンにします。

Asset Manager Automated Process Managerをデータベースに接続するには、次のいずれかの方法を行います。

- **【ファイル/データベースに接続】** メニュー
-  アイコンをクリックする。

Asset Manager Automated Process Managerをデータベースから切断する

Asset Manager Automated Process Managerをデータベースから切断するには、次のいずれかの方法を行います。

- **【ファイル/データベースの接続解除】** メニュー
-  アイコンをクリックする。

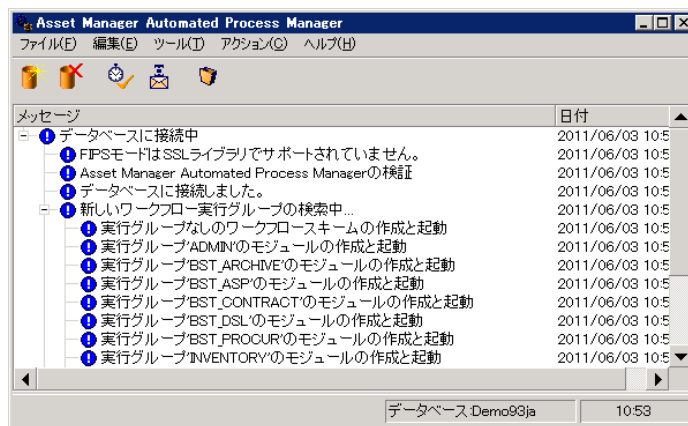
Asset Manager Automated Process Managerを終了する場合

Asset Manager Automated Process Managerでデータベースに接続した後に接続エラーが発生した場合は、**【全般】** タブページで定義した周期でデータベースへの再接続を試みます。この再接続が実行されるのは、最低1つの接続に成功した場合のみです。

Asset Manager Automated Process Managerとメッセージシステム

Asset Manager Automated Process Managerをインストールするコンピュータには、有効なメッセージシステムをインストールしておく必要があります。また、各ログインユーザがメッセージの送信に使うメッセージシステムをデータベースで正しく設定しておく必要があります。






Asset Manager Automated Process Managerのメイン画面



メイン画面には、Asset Manager Automated Process Managerが処理したすべてのイベントが表示されます。

プログラムメニューとそのアイコンの各機能を次に示します。

表 13.1. Asset Manager Automated Process Managerプログラムのメニューとアイコン

アイコン	メニュー	関数
	[ファイル] / [データベースに接続]	データベースに接続します。
	[ファイル] / [データベースの接続解除]	データベースとの接続を解除します。
	[アクション] / [起動]	起動したい期限の監視モジュールを選択します。
	[アクション] / [リストを空にする]	メイン画面からメッセージを消去します。
	[アクション] / [メッセージシステムに接続]	外部のメッセージシステムに接続を試みます。 このボタンは、問題なく接続している場合は淡色で表示されます。 接続できなかった場合にこのボタンは有効になります。このボタンをクリックして接続を試みます。

イベントのリストが大きくなりすぎた場合は、いつでも【アクション/リストを空にする】コマンドを使ってリストを空にすることができます。

Asset Manager Automated Process Managerの全般オプション

主なモニタオプションにアクセスするには、【ツール/オプション】メニューを選択します。

これらのオプションで、Asset Manager Automated Process Managerの一般的な操作を管理します。

【再接続の間隔】フレーム、【データベース】フレーム

このオプションは、Asset Manager Automated Process Managerが1回で問題なくデータベースに接続した後にエラーメッセージを受信したときに使われます。

エラーが発生すると、Asset Manager Automated Process Managerはデータベース接続が無効になったと見なし、期限のモニタを停止します。エラーが発生する前に1回でもデータベースに問題なく接続したことがある場合は、Asset Manager Automated Process Managerが、【再接続の間隔】フレーム内の【データベース】フィールドで指定した周期でデータベースへの再接続を試みます。

Asset Manager Automated Process Managerがデータベースに再接続すると、モニタ機能が再開されます。

このフィールドのデータ入力フォーマットには、「期間」を指定します。

【再接続の間隔】フレーム、【メッセージ】フレーム

外部メッセージシステムに問題が発生した場合、Asset Manager Automated Process Managerはそのメッセージシステムへのメッセージの送信を停止します。

Asset Manager Automated Process Managerは、このフィールドで定義した周期で外部メッセージシステムへの再接続を試みます。

このフィールドのデータ入力フォーマットには、「期間」を指定します。

ログファイル

ファイル

このオプションで指定するファイルに、Asset Manager Automated Process Managerのメイン画面に表示されるメッセージが保存されます。

最大サイズ

このフィールドで、Asset Manager Automated Process Managerのメイン画面に表示されるメッセージを記録するファイルのサイズを制限します。

保存容量がこの最大サイズに達すると、最も古いメッセージが削除され、代わりに新しいメッセージが保存されます。

タイムゾーン

設定画面の [タイムゾーン] フレームで、次のオプションからタイムゾーンのテストの実行方法を選択します。

- データベースサーバのタイムゾーンの検証
- サーバと比較したローカルマシン時間の検証

これらの2つのテストはいずれも、データベースサーバの時刻とAsset Manager Automated Process Managerをインストールしたマシンの時刻を比較します。時間差は $(n * 30分) + m$ で表されます。ただし、mは-15から+15分です。

2つのテストに共通する特徴

時間差が5分を超えると、Asset Manager Automated Process Managerがインストールされているマシンのローカル時刻を更新するメッセージが表示されます。

この更新を拒否すると（例えば、サーバの時刻を更新する場合）、接続が拒否されます。データベースサーバの時刻とAsset Manager Automated Process Managerをインストールしたマシンの時刻のいずれか、または両方を変更して、2つの時刻の差を5分以内にすると、再接続できるようになります。

[データベースサーバのタイムゾーンの検証] オプションの特徴

必要に応じて、Asset Managerのオプションテーブルで指定されているサーバのタイムゾーンに関する情報が更新されます（ $(n * 30分)$ の数値がサーバのタイムゾーンに対応しない場合）。

この機能を正しく実行するには、Asset Manager Automated Process Managerが稼動しているマシンで、時刻と夏時間の変更に関する情報を正しく設定しておく必要があります。

[サーバと比較したローカルマシン時間の検証] オプションの特徴

Asset Managerの内部操作に必要なサーバのタイムゾーンを取得します。

注意:

どちらのオプションを選択しても、Asset Manager Automated Process Managerがデータベースに接続する時、Asset Manager Automated Process Managerの [モジュールの設定] ウィンドウ（ [ツール/モジュールの設定...] メニュー）で定義された間隔で定期的に、テストが行われます。

Asset Manager Automated Process Managerでモニタするモジュールを設定する

はじめに

Asset Manager Automated Process Managerでモニタするモジュールを設定するには、**【ツール/モジュールの設定】**モジュールメニューを選択します。

モジュールの設定画面には、モジュールのリストが表示されます。各モジュールに対して次の情報を定義します。

- モジュールを有効にするかどうか
- モジュールが実行するタスク
- 各モジュールのモニタの周期

ヒント:

複数のマシンのそれぞれでAsset Manager Automated Process Managerのセッションを実行することにより、モニタするモジュールを複数のマシンで分担することができます。これによりパフォーマンスの向上を図れます。ただし、1つのモジュールは同時に1台のマシンでしか実行することは出来ません。

以下では、Asset Manager Automated Process Managerが管理する各モジュールについて説明します。これらのモジュールはAsset Manager Automated Process Managerでリストになっている「説明」と「名前」で識別されます。

Asset Manager Automated Process Manager検証スケジュール

モジュールの検証スケジュールを定義するには：

- 1 **【ツール/モジュールの設定】**メニューを選択します。
- 2 設定するモジュールを選択します。
- 3 **【検証スケジュール】** 枠の左タブページで情報を指定します。
- 4 必要に応じて、タブページのラベルをマウスの右ボタンでクリックして**【規則の追加】**メニューを選択し、新しいタブページを作成します。
タブページでは、モニタする日付と時刻を定義できます。

表 13.2. 実行日

【実行日】フィールドの値	モニタの実行間隔
毎日	年間を通じて毎日、例外なく実行します。

[実行日] フィールドの値	モニタの実行間隔
曜日指定	<p>[実行日] フィールドの右のフィールドに表示されるリストデータから曜日を選択します。</p> <p>例</p> <p>毎週月曜日</p>
日付指定	<p>日付のみ、または年と月を組み合わせで指定できます。指定するには、[日]、[月]、[年] チェックボックスをオンにします。</p> <p>例</p> <p>7月20日</p>
第1 第2 第3 第4 最後から2番目 最後	<p>[日]チェックボックスを使って定義した曜日に、[月] および [年] チェックボックスを使って定義した月と年を組み合わせます。</p> <p>例</p> <p>毎月「第1」金曜日</p> <p>9月の「第2」月曜日</p> <p>11月の「最後から2番目」の水曜日</p> <p>2000年の毎月「最終」火曜日</p>

モニタする時刻

定期的

検証モジュールの実行間隔は1日の時刻に応じて2つ定義することができます。1つめの実行間隔は、設定した期間の間、すなわち「**期間内**」に適用されます。次の2つの方法で期間を設定できます。

- スケジュールバーをクリックしてドラッグし、期間を設定します。
- スケジュールバーの右のフィールドに直接リストデータの値を入力します。次のシンタックスを使います。

<期間の開始時刻> - <期間の終了時刻>

Asset Managerで定義した時刻形式に従う必要があります。

複数の期間を定義するには、セミコロン (;) で区切ります。

2つ目の実行間隔は、前の手順で定義した期間の外、すなわち「**期間外**」に適用されます。

例：

リストデータ

検証を実行する時刻を右のフィールドに複数入力できます。

- Asset Managerで定義した時刻形式に従う必要があります。
- 複数の時刻は、セミコロン (;) で区切ります。

例：

実行時間	リストデータ	08:00:00;10:30:00;12:00:00;14:00:00;16:30:00;18:00:00
------	--------	---

プレビュー

次のプレビューを実行できます。

- モジュールの検証スケジュールで定義した規則を【プレビュー】フィールドで確認できます。
- モジュールを選択した場合は、モジュールに設定したすべての規則を、モジュールタブの【プレビュー】タブページで確認できます。

[データベースにNTドメインのコンピュータを追加] モジュール (AddCpu)

Asset Manager Automated Process Managerでは、NTドメインで宣言されているコンピュータの取得をプログラムできます。

分析するドメインは、HP Connect-Itの「addecpu.scn」シナリオで指定されています。

【データベースにNTドメインのコンピュータを追加】モジュールを有効にする前に、以下のパラメータが適切であることを確認してください。

- Asset Manager Automated Process Managerの【データベースにNTドメインのコンピュータを追加】モジュールのパラメータ
- HP Connect-It「addecpu.scn」シナリオのパラメータ（HP Connect-Itインストールフォルダの「scenario\ntsec\ntac93」サブフォルダに格納されています）。

[データベースにNTドメインのコンピュータを追加] モジュールのパラメータ [ユーザーデータ] フィールド)

デフォルトのパラメータは次のとおりです。

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/ntsec/ntac$version$/addcpu.scn' -dc:AssetCenter.SERVER=$cnx$ -dc:AssetCenter.LOGIN=$login$ -dc:AssetCenter.TEXTPASSWORD=$pwd
```

上記のパラメータに関する説明

- \$connectit_exedir\$は、Windowsのレジストリにある「conitsvc.exe」プログラムのパスを格納します。

このパラメータを変更する必要はありません。

- **-once**は、「conitsvc.exe」を1度のみ実行することを意味します（HP Connect-Itの「**Once**」スケジューラを使用）。
このパラメータは、変更しないでください。Asset Manager Automated Process Managerアプリケーションでスケジュールを定義します。
- **-wpplog**は、Asset Manager Automated Process Managerによる読み取りが可能なフォーマットでシナリオの実行ログを生成するのに使用します。
- **\$connectit_exedir\$../scenario/ntsec/ntac93/addcpu.scn**は、使用するHP Connect-Itシナリオへのパスです。
別のHP Connect-Itシナリオを使用する場合は、このパラメータを変更します。
- **-d:Asset Manager.SERVER=\$cnx\$ -d:Asset Manager.LOGIN=\$login\$ -d:Asset Manager.TEXTPASSWORD=\$pwd\$**は、Asset Manager Automated Process Managerが開くAsset Managerの接続名、ログイン、およびパスワードを含みます。
上記のパラメータは、「**addcpu.scn**」シナリオのAsset Managerコネクタで定義された値を上書きします。
これらのパラメータを変更する必要はありません。

HP Connect-Itの「addcpu.scn」シナリオのパラメータ

「addcpu.scn」シナリオを変更するには：

- 1 HP Connect-Itシナリオエディタを実行します。
- 2 「addcpu.scn」シナリオを開きます（HP Connect-Itインストールフォルダの「scenario\ntsec\ntac93」サブフォルダ内に格納されています）。
- 3 シナリオ図の画面で**NT security**ボックスのタイトルバーをクリックして、**NT security**コネクタを選択します。
- 4 [ツール/設定] メニューを選択します。
- 5 [次へ] をクリックします。
- 6 [ドメイン] フィールドにコンピュータをインポートするドメインの名前を入力します。

警告:

次のコンピュータが取得されます。

- Asset Manager Automated Process Managerのユーザが接続するドメインと同じドメインに属するユーザ
- Asset Manager Automated Process Managerのユーザが接続するドメインの「信頼される側」のドメインに属するユーザ

ヒント:

ある特定のコンピュータが取得されるかどうかを知るには：

- 1 Asset Manager Automated Process ManagerのマシンでWindowsエクスペローラを実行します。
- 2 近隣ネットワークを表示します。
- 3 Asset Manager Automated Process Managerは、ここに表示されるコンピュータからデータを取得することができます。

Asset Managerデータベースにどのデータが入力されるかを知るには、「addecpu.scn」シナリオの詳細を確認します。

モジュールは、[コンピュータ] テーブルの [次回のスキャン] フィールドにモジュールの実行日「00:00」を入力します。

NTセキュリティコネクタの詳細については、HP Connect-Itに付属している『コネクタ』ガイドの「アプリケーションコネクタ」の章、「NTセキュリティコネクタ」のセクションを参照してください。

Asset Managerコネクタの詳細については、HP Connect-Itに付属している『コネクタ』ガイドの「Hewlett-Packardコネクタ」の章、「Asset Managerコネクタ」のセクションを参照してください。

[データベースにNTユーザを追加] モジュール (AddUser)

Asset Manager Automated Process Managerでは、NTドメインで宣言されているユーザの取得をプログラムできます。

このモジュールは、[部署と従業員] テーブルに、Asset Managerデータベース（統合NTセキュリティを使用）の接続に関する情報を入力するために役立ちます。

分析するドメインは、HP Connect-Itの「adduser.scn」シナリオで指定されています。

[データベースにNTユーザを追加] モジュールを有効にする前に、以下のパラメータが適切であることを確認してください。

- Asset Manager Automated Process Managerの [データベースにNTユーザを追加] モジュールのパラメータ
- HP Connect-It 「adduser.scn」シナリオのパラメータ（HP Connect-Itインストールフォルダの「scenario\ntsec\ntac93」サブフォルダに格納されています）。

[データベースにNTユーザを追加] モジュールのパラメータ [ユーザデータ] フィールド)

デフォルトのパラメータは次のとおりです。

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$./scenario/ntsec/ntac$version$/adduser.scn' -dc:AssetCenter.SERVER=$cnx$ -dc:AssetCenter.LOGIN=$login$ -dc:AssetCenter.TEXTPASSWORD=$pwd$
```

上記のパラメータに関する説明

- `$connectit_exedir$`は、Windowsのレジストリにある「conitsvc.exe」プログラムのパスを格納します。
このパラメータを変更する必要はありません。
- `-once`は、「conitsvc.exe」を1度のみ実行することを意味します（HP Connect-Itの「Once」スケジューラを使用）。
このパラメータは、変更しないでください。Asset Manager Automated Process Managerアプリケーションでスケジュールを定義します。
- `-wpplog`は、Asset Manager Automated Process Managerによる読み取りが可能なフォーマットでシナリオの実行ログを生成するのに使用します。
- `$connectit_exedir$./scenario/ntsec/ntac93/adduser.scn`は、使用するHP Connect-Itシナリオへのパスです。
別のHP Connect-Itシナリオを使用する場合は、このパラメータを変更します。
- `-d:Asset Manager.SERVER=cnx -d:Asset Manager.LOGIN=$login$ -d:Asset Manager.TEXTPASSWORD=pwd`は、Asset Manager Automated Process Managerが開くAsset Managerの接続名、ログイン、およびパスワードを含みます。
上記のパラメータは、「adduser.scn」シナリオのAsset Managerコネクタで定義された値を上書きします。
これらのパラメータを変更する必要はありません。

HP Connect-Itの「adduser.scn」シナリオのパラメータ

「adduser.scn」シナリオを変更するには：

- 1 HP Connect-Itシナリオエディタを実行します。
- 2 「adduser.scn」シナリオを開きます（HP Connect-Itインストールフォルダの「scenario\ntsec\ntac93」サブフォルダ内に格納されています）。
- 3 シナリオ図の画面で**NT security**ボックスのタイトルバーをクリックして、**NT security**コネクタを選択します。
- 4 [ツール/設定] メニューを選択します。
- 5 [次へ] をクリックします。
- 6 [ドメイン] フィールドにコンピュータをインポートするドメインの名前を入力します。

ヒント:

複数のドメインを探索する場合は、ドメインごとにHP Connect-ItシナリオとAsset Manager Automated Process Managerモジュールを作成することをお勧めします。

警告:

次のユーザが取得されます。

- Asset Manager Automated Process Managerのユーザが接続するドメインと同じドメインに属するユーザ
- Asset Manager Automated Process Managerのユーザが接続するドメインの「信頼される側」のドメインに属するユーザ

ヒント:

ある特定のコンピュータが取得されるかどうかを知るには：

- 1 Asset Manager Automated Process ManagerのコンピュータでWindowsエクスプローラを実行します。
- 2 共有フォルダを作成します。
- 3 この共有フォルダのアクセス権を指定します。
- 4 アクセス権にユーザを追加します。
- 5 Asset Manager Automated Process Managerは、ここに表示されるユーザからデータを取得することができます。

Asset Managerデータベースにどのデータが入力されるかを知るには、「adduser.scn」シナリオの詳細を確認します。

NTセキュリティコネクタの詳細については、HP Connect-Itに付属している『コネクタ』ガイドの「アプリケーションコネクタ」の章、「NTセキュリティコネクタ」のセクションを参照してください。

Asset Managerコネクタの詳細については、HP Connect-Itに付属している『コネクタ』ガイドの「Hewlett-Packardコネクタ」の章、「Asset Managerコネクタ」のセクションを参照してください。

【賃貸料とローンの計算】モジュール (Rent)

【賃貸料とローンの計算】モジュールは、次の手順で経費明細を生成または再計算します。

- 契約賃貸料の支払
- ローンの支払

【賃貸料とローンの計算】モジュールの設定では次を定義します。

- 賃貸料の支払に対して作成された経費明細の日数
- モジュールの実行頻度

概要

賃貸料

Asset Manager Automated Process Managerは、経費明細の作成または再計算が必要かどうかをあらかじめ設定した間隔で検証します。必要である場合はこれらの操作を行います。

定期的な賃貸料に関連する経費明細のチェックと作成が終了すると、Asset Manager Automated Process Managerは最新の経費明細（過去または現在）の日付を【計算開始日】（SQL名：dRecalcul）フィールドに入力します。

- 契約賃貸料を資産に配分（比例配分または割り当て）する場合、Asset Manager Automated Process Managerは、資産の詳細画面の【取得】タブの【賃貸料】サブタブにある【計算開始日】フィールドを変更します。
- 契約賃貸料を資産に配分（比例配分または割り当て）しない場合、Asset Manager Automated Process Managerは、契約の詳細画面の【賃貸料】タブの【賃貸料】サブタブにある【計算開始日】フィールドを変更します。

ただし、1つ1つの経費明細を作成するたびに再計算を行うわけではありません。

- 定期的な賃貸料に関連する経費明細の見積は、必ず再計算されます。
- 各賃貸料に固有の【計算開始日】フィールドは、定期的に支払う賃貸料の経費明細が最後に再計算された日付を表します。

借主は、【計算開始日】フィールドを直接編集し、見積られた経費明細以外の再計算日を直接変更できます。これにより、税率が変更された場合などに不適切な経費明細を再計算することができます。

ローンの支払

Asset Manager Automated Process Managerは経費明細ですべてのローン返済明細の作成または再計算が必要かどうかを定期的にチェックします。

作成パラメータ

 **注意:**

このセクションは賃貸料の支払にのみ適用され、ローンの支払には適用されません。

【ユーザデータ】フィールドでは、賃貸料の作成パラメータを設定します。このフィールドのシンタックスは次のとおりです。

<Duration>d

Durationには、賃貸料項目の見積を計算する日数を設定します。例えば、90日間の賃貸料を計算するには、次のように入力します。

90d

 **注意:**

トランザクションごとの賃貸料の最大計算回数は、「amsrv.cfg」設定ファイルのUserDataエントリで指定します。

このファイルの場所：▶ 『**Asset Manager - インストールとアップグレード**』ガイドの「.iniおよび.cfgファイル」の章。

Asset Manager Automated Process Managerは、指定した期間内で賃貸料を見積り、経費明細を作成します。作成しない場合は、単にこのフィールドを「0」に設定してください。

例

次の内容を設定することにします。

- 契約は、2004年7月1日から2007年7月1日まで有効です。
- 賃貸料は、月1回、月初めに支払います。
- Asset Manager Automated Process Managerでは、賃貸料の支払を2ヶ月ごとに検証し、向こう12ヶ月に支払う賃貸料の見積を作成します。

Asset Manager Automated Process Managerを2005年7月1日に初めて起動した場合、次のデータが作成されます。

- 2004年7月1日から2005年7月1日までの過去の賃貸料
- 2005年7月1日現在の賃貸料
- 01.08.05から01.07.06までの賃貸料の見積

計算が終ると、[計算開始日] フィールドには見積った経費明細の日付ではなく最新の経費明細の日付、つまり2005年7月1日が表示されます。

Asset Manager Automated Process Managerはバックグラウンドで動作し、2ヶ月後の2005年9月1日に次のデータを作成します。

- 2005年10月1日から2006年9月1日までの賃貸料の見積
- 支払日が[計算開始日] フィールドの日付より前または現在の賃貸料、つまり、2005年8月1日から2005年10月1日までの賃貸料

[規定の損失額の計算] モジュール (LostVal)

Asset Manager Automated Process Managerは、計算方法が[全期間で計算]

(リース明細の詳細画面の[リース] タブページの[計算] (SQL名: seLossValCalcMode) フィールド) に設定されているリース明細の損失額を定期的に再計算します。このように、Asset Manager Automated Process Manager

がデータベースにアクセスした後に損失額の計算規則を変更すると、損失額が更新されます。

[受領品に対応する資産、消耗品などの作成 (Delivery)] モジュール

前提条件

このモジュールを起動する前に、以下の操作を実行する必要があります。

- Asset Managerを実行します。
- [管理/データベースオプション] メニューを選択します。
- [調達/Asset Manager Automated Process Managerで受領品をポートフォリオに自動的に作成] オプションを選択します。
- このオプションの値を「はい」に設定します。

このモジュールが実行するタスク

このモジュールは、[受領品] (SQL名 : amItemReceived) テーブルのレコードを処理し、受領品に対応するレコード (資産、消耗品など) を適切なテーブルに作成します。

このモジュールの利点

Asset ManagerではなくAsset Manager Automated Process Managerを使ってこのタスクを実行すると、注文品の受領を行うユーザのパフォーマンスを向上を図ることができます。

実行の頻度

ユーザが受領品に対応するレコードに迅速にアクセスできるようにするには、このモジュールを毎日、数回実行することをお勧めします。

[ヘルプデスクアラームのトリガ] モジュール (HDAlarms)

Asset Manager Automated Process Managerは、ヘルプデスクに特有の期限をモニタします。

注意:

ヘルプデスクに特有の期限は、エスカレーション処理の詳細画面の [全般] タブの業務用カレンダーを考慮に入れて計算されます。又、期限の計算時には期限は時間に変換されます。

エスカレーション処理のアラーム

ヘルプデスクチケットはそれぞれ、チケットの詳細画面の【トラッキング】タブページの【エスカレーション】（SQL名：EscalScheme）フィールドに明記されたエスカレーション処理に、関連づけられています。

Asset Manager Automated Process Managerでモニタする期限は、エスカレーション処理の詳細画面の【アラーム】タブページで指定します。期限は、それぞれのチケットの特定の処理段階に対応しています。

- チケットをオープンしてから経過した時間：チケットのオープン日は、チケットの詳細画面の【オープン日】（SQL名：dtOpened）フィールドで指定されています。
- 解決時間比（%）：チケットに関連付けられたエスカレーション処理の、【期限】（SQL名：tsResolDelay）フィールドに指定された期間のパーセンテージを示しています。
- 解決期限前の時間：チケットの期限は、チケットの詳細画面の【トラッキング】タブページの【解決予定期限】（SQL名：dtResolLimit）フィールドで指定されています。

注意:

Asset Manager Automated Process Managerはエスカレーション処理の詳細画面の【遷移】タブと【割り当て】タブで定義されたアクションをトリガしません。このアクションをトリガするのは、Asset Managerのエージェントです。

保留の作業履歴のアラーム

Asset Manager Automated Process Managerは保留のヘルプデスクチケットをモニタします。

保留の作業履歴の【保留】タブページの期限期間が過ぎると、Asset Manager Automated Process Managerは、保留作業履歴の【保留】タブページで定義されたアクションを自動的にトリガします。

[実行グループにワークフロー規則を適用]（WkGroup*）モジュール

Asset Manager Automated Process Managerは、ワークフローの実行グループ（例：ADMIN）を検出すると、ワークフロー規則を実行します。

ワークフロー実行グループをモニタする

Asset Manager Automated Process Managerは、実行グループに関連付けられているワークフローに固有の期限をモニタします。

アクティビティがトリガされたときにAsset Manager Automated Process Managerでモニタする期限は、ワークフローのアクティビティテンプレートの詳細画面の【アラーム】タブページで定義します。

これらの期限は、実行する一連のタスクに設定した期限によって定義されます。

 **注意:**

ワークフローに固有の期限は、ワークフローのアクティビティテンプレートの詳細画面／[時間] タブページで指定した業務用カレンダーに基づいて計算されます。計算された期限の制限時間は、業務時間に変換されます。

定期的なイベントを処理する

定期的なイベントの場合、Asset Manager Automated Process Managerは、イベントの詳細画面の[パラメータ] タブで定義した周期で起動条件をテストし、起動条件が満たされる場合にイベントをトリガします。

Asset Manager Automated Process Managerの処理内容は、次のようにイベントの詳細画面の[全般] タブページで指定したイベントの処理モードによって決まります。

- **イベントを保存してからサーバで処理**：イベントが発生すると同時に、Asset Manager Automated Process ManagerはこれをamWfOccurEvent (SQL名) テーブルに保存します。
その後、Asset Manager Automated Process Managerの設定画面で定義した検証周期に従って、トランジションを起動します。
- **イベントを保存してすぐに処理**：イベントが発生すると同時に、Asset Manager Automated Process ManagerはこれをamWfOccurEvent (SQL名) テーブルに保存し、すぐにトランジションを起動します。
- **イベントを保存しないですぐに処理**：イベントが発生すると同時に、Asset Manager Automated Process Managerはトランジションを起動します。

遷移の起動

Asset Manager Automated Process Managerは、設定画面で定義した周期に従ってイベントの遷移を起動します。対象となるイベントは次のとおりです。

- [システム] イベント
- [データベース] タイプと[定期的] タイプのイベントで処理モードが[イベントを保存してからサーバで処理] に設定されているもの

タスクの実行

Asset Manager Automated Process Managerでは、[自動アクション] タイプまたは[テスト/スクリプト] タイプのアクティビティによって発生するタスクを実行します。ただし、[アクションを即実行] (SQL名: bExecImmediately) チェックボックスが選択されているアクティビティから発生するタスクは行いません。

Asset Manager Automated Process Managerがタスクを検証して実行する周期は、Asset Manager Automated Process Managerの設定画面で指定します。

【アクションを即実行】（SQL名：bExecImmediately）チェックボックスがオンになっている【自動アクション】タイプまたは【テスト/スクリプト】タイプのアクティビティから発生するタスクの場合は、次のようになります。

- Asset Manager Automated Process Managerが起動したトランジションによって作成されたタスクの場合は、Asset Manager Automated Process Managerがそのタスクを実行します。この場合、Asset Manager Automated Process Managerは、Asset Manager Automated Process Managerが作成したトランジションを起動した直後にタスクを実行します。
- それ以外の場合は、Asset Managerクライアントマシンでタスクを実行します。

【テーブル用統計情報の更新 (Stats)】モジュール

このモジュールは、データベースに関する統計を更新します。

Asset ManagerがサポートするDBMSは、SQLクエリプランを最適化するためにデータベース統計を使用します。

データベース統計が更新されていない場合、DBMSは最も効率的なインデックスを検出できません。

1週間に1度はこのモジュールを起動するようをお勧めします。データベースが大きく変更される場合は、毎晩起動します。

【入カイベントテーブルの消去 (PurgeEventInTable)】モジュール

このモジュールは、【入カイベント】（SQL名：amInputEvent）テーブルのレコードを、以下のフィールドの値に応じて削除します。

- 【入カイベント】（SQL名：amInputEvent）テーブルの【ステータス】（SQL名：seStatus）フィールド
- 【入カイベント】（SQL名：amInputEvent）テーブルの【削除】（SQL名：seRemoveFlag）フィールド
- 【入カイベントの有効時間（時間）】オプションで指定された期限（Asset Managerの【管理/データベースオプション】メニュー）

【出カイベントテーブルの消去 (PurgeEventOutTable)】モジュール

このモジュールは、【入カイベント】（SQL名：amInputEvent）テーブルのレコードを、以下のフィールドの値に応じて削除します。

- 【出カイベント】（SQL名：amOutputEvent）テーブルの【ステータス】（SQL名：seStatus）フィールド
- 【出カイベント】（SQL名：amOutputEvent）テーブルの【削除】（SQL名：seRemoveFlag）フィールド

- [出カイベントの有効時間 (時間)] オプションで指定された期限 (Asset Managerの [管理/データベースオプション] メニュー)

[新しいワークフロー実行グループの検索 (WorkflowFinder)] モジュール

Asset Manager Automated Process Managerは、新しいワークフロー実行グループの作成をモニタします。

例えば、Asset Manager Automated Process Managerが新しいワークフロー実行グループ「G」を検出すると、[実行グループ'G'にワークフロー規則を適用]という新しいモニタモジュールを作成します。

この機能によって、次のことが可能になります。

- ワークフローの実行グループごとに検証スケジュールを定義できます。
- 複数の異なるワークフロー実行グループを複数のAsset Manager Automated Process Managerで分担してモニタできます。

[データベースサーバに信号送信 (UpdateToken)] モジュール

Asset Manager Automated Process Managerは、データベースサーバに機能していることを示す信号を定期的に送信します。

データベースサーバがAsset Manager Automated Process Managerから1時間以上信号を受信しない場合は、ユーザがAsset Managerでデータベースに接続した時に、メッセージが表示されます。

メッセージには、Asset Manager Automated Process Managerがこのデータベース上で稼動していないため、モニタ機能が中断することが示されます。

データベースサーバがAsset Manager Automated Process Managerから1週間以上信号を受信しない場合は、データベースに接続できなくなります。

[コストセンタの経費を分割する (CostCenter)] モジュール

Asset Manager Automated Process Managerで、経費明細を分割することができます。

概要

Asset Manager Automated Process Managerは、まず [分割処理のステータス] (SQL名: seSplitStatus) フィールドが [未分割] に設定されている経費明細を検索して分割する経費明細を特定します。

デフォルトでは、そのステータス (経費明細の [ステータス] (SQL名: seStatus) フィールドの値) に関わらず、すべての経費明細を分割します。

Asset Manager Automated Process Managerが特定した経費明細を分割すると、次のようになります。

- 親コストセンタ内に、分割した経費明細と同等の借方の経費明細が作成されます。
- 分割先のコストセンタに配分比（パーセント）分の経費明細が作成されます。この経費明細の分割処理のステータスはデフォルトで【未分割】になります。

特殊な例：コストセンタを削除する

Asset Managerでは、コストセンタを削除しようとしても、そのコストセンタに経費明細がある場合は、【編集/オプション】メニューの【編集】項目の【拡張削除の許可】オプションがオンになっていない限り削除できません。

この場合は、次の3つの方法でコストセンタを削除できます。

- リンクしているすべてのレコードを削除する。
- リンクしているレコードを切り離します。
- リンクしているレコードを別のレコードにリンクします。

それぞれの方法でコストセンタを削除した場合の結果を次に示します。

リンクしているすべてのレコードを削除する

コストセンタが削除されると、Asset Managerは次の明細を削除します。

- 削除されたコストセンタの経費明細
- 削除されたコストセンタの分割処理で発生した経費明細

削除されたコストセンタで分割処理が行われ、別の経費明細が発生した場合は、Asset Managerが分割前の元の経費明細の【分割処理のステータス】（SQL名：seSplitStatus）フィールドを「未分割」に変更します。

このようにステータスは未分割に戻っても、分割により発生した経費明細がまだリンクされている経費明細をAsset Manager Automated Process Managerが検出した場合は、Asset Manager Automated Process Managerがそれらのリンクしている経費明細をすべて削除します。それと同時にAsset Manager Automated Process Managerは分割前の元の経費明細も削除します。

次に、Asset Manager Automated Process Managerは未分割のステータスに戻った経費明細を分割し、新しいパラメータに基づいて再計算します。

リンクしているすべてのレコードを切り離す

この場合は次のようになります。

- 削除されたコストセンタの経費明細は、切り離したコストセンタとは関連がなくなります。
- 経費明細が、削除されたコストセンタの分割前の元の経費明細の場合は、その経費明細が再分割されます。
- 経費明細が、削除されたコストセンタの分割操作により発生した経費明細の場合は、変更されません。

リンクしているレコードを別のレコードにリンクする

この場合は、削除されたコストセンタに代わるコストセンタXを選択します。

- 削除されたコストセンタの経費明細をコストセンタXに関連付けます。
- 経費明細が、削除されたコストセンタの分割前の元の経費明細の場合は、コストセンタXが新しいコストセンタと見なされ、その経費明細が再分割されます。
- 削除されたコストセンタの分割操作によって発生した経費明細は削除され、コストセンタXの経費明細が分割されます。

[データベースサーバのタイムゾーンの検証 (TimeZone)] モジュール

このモジュールは、サーバの現地時間とクライアントコンピュータ間の時間差を確認します。このモジュールは、クライアントマシンのタイムゾーンを指定した場合（ナビゲーションメニューの [管理/タイムゾーン] ）に役立ちます。

[アラームの検証 (Alarms)] モジュール

モニタするアラーム

資産のアラーム

以下の重要な日付をモニタします。

- 資産の予約終了日：この日付は、資産の詳細画面の [ポートフォリオ/予約] タブページの [終了日] (SQL名: dtEnd) フィールドに表示されます。
- 資産の保証期限：資産の詳細画面の [メンテナンス] タブページ / [失効日] (SQL名: dWarrEnd) フィールド
- 資産のリース、レンタル、ローンの期間満了日：このアラームは、資産の取得方法（資産の詳細画面の [取得] タブページ / [取得方法] (SQL名: seAcquMethod) フィールド) が [リース]、[レンタル]、[ローン] のいずれかに設定されている場合にのみ定義できます。期間満了日は、[取得] タブページ / [価格と条件] サブタブページ / [終了日] (SQL名: dEndAcqu) フィールドに表示されます。
- 資産の賃貸料の最終支払日：アラームは賃貸期間の終了日（ [取得] タブページ / 賃貸料の詳細サブタブページ / [スケジュール] フレーム）にリンクできます。

消耗品のアラーム

Asset Manager Automated Process Managerは、消耗品の予約終了日をモニタします。この日付は、消耗品の予約の詳細画面の [予約終了日] (SQL名: dReservEnd) フィールドに表示されます。消耗品の予約の詳細画面にアクセスするには：

- 1 Asset Managerを起動します。
- 2 [資産ライフサイクル/調達ライフサイクル/依頼/購入依頼] を選択します。
- 3 消耗品を予約する購入依頼を選択します。
- 4 この購入依頼の明細を表示します。
- 5 消耗品に対応する依頼明細の詳細画面を表示します。
- 6 依頼明細の[予約] タブページを表示します。このタブページには、予約している消耗品のリストが表示されます。
- 7 予約の詳細を表示します。
[終了日] (SQL名: dtEnd) フィールドがモニタされます。

プロジェクトのアラーム

Asset Manager Automated Process Managerは、プロジェクトの終了日（プロジェクトの詳細画面の[全般] タブページ/ [終了] (SQL名: dEnd) フィールド) をモニタします。

契約のアラーム

以下の重要な日付をモニタします。

- 期間満了日: 契約の詳細画面の[全般] タブページ/ [終了] (SQL名: dEnd) フィールド
- 契約の[タイプ] (SQL名: seType) が[リース明細] または[マスターリース (リース)] の場合、アラームは期間満了時オプションの通知日にリンクできます。これらの日は、可能性のある期間満了時オプション、[更新]、[買い取り]、[返品] が記述されるサブタブのある[買い取り通知期限]、[更新通知期限] または[返却通知期限] フィールドの右側に表示されます。
- 契約の[タイプ] (SQL名: seType) が[リース明細] の場合、アラームは[賃貸料] タブの[スケジュール] フレームの[契約賃貸料終了日] にリンクできます。

購入依頼のアラーム

購入依頼の取得方法（購入依頼の詳細画面の[ファイナンス] タブページ/ [取得方法] (SQL名: seAcquMethod) フィールド) が[リース]、[レンタル]、[ローン] のいずれかに設定されている場合は、リース、レンタル、ローンの終了日（購入依頼の詳細画面の[ファイナンス] タブページ/ [取得方法] フィールド) に関連するアラームを定義できます。

見積と発注についても同様です。

第1レベルのアクションがトリガされると、第2レベルのアラームではどうなるか。

2つのレベルをアラームに設定する場合は、第1レベルのアラームで実行したアクションによって、第2レベルのアラームがトリガされるかどうかが決まります。

- 第1レベルのアラームが、Asset Managerの内部メッセージシステム以外のメッセージシステム（サードパーティのメッセージシステムなど）でメッセージを送信するアクションを起動した場合は、第2レベルのアラームは常に定義された時間にトリガされます。
- 第1レベルのアラームにより、内部メッセージシステムでAsset Managerユーザのグループにメッセージが送信され、受信者の誰かがそのメッセージを開封した場合は、第2レベルに定義されているアクションはトリガされません。

[定期的経費付替え規則をトリガ] モジュール (Cbktimer)

このモジュールは経費付替えシステムの規則をトリガするのに使用されます。



注意:

経費付替え規則は経費付替え明細の作成を自動化するのに使用されます。

Asset Manager Automated Process Managerは、[定期的経費付替え規則をトリガ] モジュールで定義されているとおりに、経費付替えシステムスケジューラを検証すべきかどうかを定期的にチェックします。

この判断は、経費付替えシステムの[規則の前の実行] フィールド (dtLastExec) の値とスケジューラの実行頻度を比較して行います。

経費付替え規則を実行すると、Asset Manager Automated Process Managerは[規則の前の実行] フィールド (dtLastExec) に実行した日と時間を保存します。

経費明細を作成できない場合、Asset Manager Automated Process Managerは経費付替えイベントを作成します。経費付替えシステムの[イベント] タブにある経費付替えイベントリストを参考にして、原因を特定することができます。

[定期的経費付替え規則をトリガ] モジュールを設定するには、[有効] オプションを選択し、モジュールのトリガ用カレンダーに入力します。[ユーザーデータ] フィールドには入力する必要はありません。

[null識別子レコードの検証 (NullRecords)] モジュール

このモジュールは、主キーがnullのレコードの整合性を確認します。

null識別子レコードは、データベースの作成時にすべてのテーブルに作成されます。

Asset Managerは、null識別子レコードを使って特定の管理タスクを実行します。このモジュールは、null識別子レコードが存在することを確認し、必要に応じて再作成します。

データベースの整合性を保つために、このモジュールを毎日最低1度は起動するようお勧めします。

【履歴項目の検証 (History)】モジュール

データベース内でレコードを破棄しても対応する履歴項目が残っている場合があります。Asset Manager Automated Process Managerは、このような履歴項目があるかどうかを検証し、あればその項目を破棄します。

【在庫の検証 (Stock)】モジュール

Asset Manager Automated Process Managerは、発注点（再発注時の在庫数）をモニタします。

Asset Manager Automated Process Managerは、在庫の詳細画面の【管理】タブページで定義した在庫規則を在庫ごとに参照します。

モデルの在庫規則に従って次の処理が実行されます。

- Asset Manager Automated Process Managerは、ポートフォリオ品目の詳細画面の【割り当て】フィールドの値を基に実際の数量を計算します。
- 数量が在庫規則の詳細画面の【発注点】（SQL名：lReordLevel）フィールドに指定した値を下回ると、Asset Manager Automated Process Managerが自動的に購入依頼を作成します。
 - 購入依頼に必要な情報は、在庫の詳細画面の【自動依頼】タブページと【管理】タブページで確認できます。
 - 購入依頼に、再発注する数量（在庫規則の詳細画面の【発注数】（SQL名：lQtyToOrder）フィールドの値）が指定されます。
- 購入依頼した物品が完全に受領されるまでは、Asset Manager Automated Process Managerは生成した在庫規則を検証しません。つまり、新しい依頼を送信しません。
- 注文品がすべて受領されると、Asset Manager Automated Process Managerは直ちに次の作業を行います。
 - 在庫数を再調整します。
 - 在庫規則の詳細画面の【依頼明細】（SQL名：ReqLine）フィールドの値を消去します。
 - 在庫規則を再度有効にします。

【Enterprise Discovery棚卸結果を使用してデータベースを更新】モジュール (EdAc)

このモジュールでは、ソフトウェア配布ツールによって実行された棚卸の結果のAsset Managerデータベースへのインポートをプログラムできます。

インポートはHP Connect-Itシナリオ「edac.scn」で行います。

【Enterprise Discovery棚卸結果を使用してデータベースを更新】モジュールを有効にする前に、次のパラメータが適切であることを確認してください。

- Asset Manager Automated Process Manager [Enterprise Discovery 棚卸結果を使用してデータベースを更新] モジュールのパラメータ
- HP Connect-It 「edac.scn」シナリオ (HP Connect-Itインストールフォルダの「scenario\ed\ed<HP Discovery and Dependency Mapping Inventoryバージョン番号>ac<Asset Managerバージョン番号>」サブフォルダ内) のパラメータ。

パラメータが正しく設定されていることを確認した後：

- [Enterprise Discovery 棚卸結果を使用してデータベースを更新] モジュールの詳細を表示します。
- [有効] オプションを選択します。
- [検証スケジュール] フレームのオプションを入力します。

【Enterprise Discovery 棚卸結果を使用してデータベースを更新】モジュールのパラメータ (【ユーザデータ】フィールド)

デフォルトのパラメータは次のとおりです。

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/.scenario/ed/ed2ac$version$/edac.scn' -dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

上記のパラメータに関する説明

- **\$connectit_exedir\$**は、Windowsのレジストリにある「conitsvc.exe」プログラムのパスを格納します。
このパラメータを変更する必要はありません。
- **-once**は、「conitsvc.exe」を1度のみ実行することを意味します (HP Connect-Itの「Once」スケジュールを使用)。
このパラメータは、変更しないでください。Asset Manager Automated Process Managerアプリケーションでスケジュールを定義します。
- **-wpplog**は、Asset Manager Automated Process Managerによる読み取りが可能なフォーマットでシナリオの実行ログを生成するのに使用します。
- **\$connectit_exedir\$/.scenario/ed/ed2ac\$version\$/edac.scn**は、使用するHP Connect-Itシナリオへのパスです。
別のHP Connect-Itシナリオを使用する場合は、このパラメータを変更します。
- **-dc:'Asset Management.SERVER'=\$cnx\$ -dc:'Asset Management.LOGIN'=\$login\$ -dc:'Asset Management.TEXTPASSWORD'=\$pwd\$**には、Asset Manager Automated Process Managerによって開かれているAsset Manager接続名、その接続で使用されているログインおよびパスワードが格納されています。
上記のパラメータは、「edac.scn」シナリオのAsset Managerコネクタレベルで定義された値を上書きします。

これらのパラメータを変更する必要はありません。

HP Connect-It 「edac.scn」 シナリオのパラメータ

「edac.scn」シナリオを変更するには：

- 1 HP Connect-Itシナリオエディタを起動します。
- 2 「edac.scn」シナリオを開きます（HP Connect-Itインストールフォルダの「ed/ed2ac<Asset Managerバージョン番号>」サブフォルダ内に格納されています）。
- 3 **[Enterprise Discovery]** タイトルバー（ボックス内ではありません）をクリックし、**[Scenario diagram]** ウィンドウの **[Enterprise Discovery]** コネクタを選択します。
- 4 **[ツール/設定]** メニューを選択します。
- 5 ウィザード内のページに入力します。
- 6 **[Finish]** をクリックして確認します。

Asset Managerデータベースにどのデータが入力されるかを知るには、「edac.scn」シナリオの詳細を確認します。

[Enterprise Discovery] 棚卸結果を使用して、データベースにあるすべてのモバイル機器を更新 (EdAcMD) モジュール

このモジュールでは、ソフトウェア配布ツールによって処理されるモバイル機器棚卸の結果のAsset Managerデータベースへのインポートをプログラムできます。

このインポートは、「edac-mobiledevices.scn」HP Connect-Itシナリオによって実行されます。

[Enterprise Discovery] 棚卸結果を使用して、データベースにあるすべてのモバイル機器を更新 モジュールを有効にする前に、次のパラメータが適切であることを確認してください。

- Asset Manager Automated Process Manager **[Enterprise Discovery] 棚卸結果を使用してデータベースにあるすべてのモバイル機器を更新** モジュールのパラメータ
- HP Connect-It 「edac-mobiledevices.scn」シナリオ（HP Connect-Itインストールフォルダの「scenario\ed\ed<HP Discovery and Dependency Mapping Inventoryバージョン番号>ac<Asset Managerバージョン番号>」サブフォルダ内）のパラメータ。

パラメータが正しく設定されていることを確認した後：

- **[Enterprise Discovery] 棚卸結果を使用して、データベースにあるすべてのモバイル機器を更新** モジュールの詳細を表示します。
- **[有効]** オプションを選択します。
- **[検証スケジュール]** フレームのオプションを入力します。

【Enterprise Discovery】棚卸結果を使用して、データベースにあるすべてのモバイル機器を更新」モジュールのパラメータ（【ユーザデータ】フィールド）

デフォルトコマンドラインのパラメータは次のとおりです。

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$./scenario/ed/ed2ac$version$/edac-mobiledevices.scn' -dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

上記のパラメータに関する説明

- **\$connectit_exedir\$**は、Windowsのレジストリにある「conitsvc.exe」プログラムのパスを格納します。
このパラメータを変更する必要はありません。
- **-once**は、「conitsvc.exe」を1度のみ実行することを意味します（HP Connect-Itの「**Once**」スケジューラを使用）。
このパラメータを変更しないでください。スケジュール情報はAsset Manager Automated Process Managerで定義されます。
- **-wpplog**を使用して、Asset Manager Automated Process Managerによる読み取りが可能なフォーマットでシナリオの実行ログを生成します。
- **\$connectit_exedir\$./scenario/ed/ed2ac\$version\$/edac-mobiledevices.scn**は、使用するHP Connect-Itシナリオのパスを指定します。
別のHP Connect-Itシナリオを使用する場合は、このパラメータを変更します。
- **-dc:'Asset Management.SERVER'=\$cnx\$ -dc:'Asset Management.LOGIN'=\$login\$ -dc:'Asset Management.TEXTPASSWORD'=\$pwd\$**には、Asset Manager Automated Process Managerによって開かれているAsset Manager接続名、その接続で使用されているログインおよびパスワードが格納されています。
上記のパラメータは、「edac-mobiledevices.scn」シナリオのAsset Managerコネクタに定義された値を上書きします。
これらのパラメータを変更する必要はありません。

HP Connect-It「edac-mobiledevices.scn」シナリオのパラメータ

「edac-mobiledevices.scn」シナリオを変更するには：

- 1 HP Connect-Itシナリオエディタを起動します。
- 2 「edac-mobiledevices.scn」シナリオを開きます（HP Connect-Itインストールフォルダの「ed/ed2ac<Asset Managerバージョン番号>」サブフォルダ内に格納されています）。
- 3 **【Enterprise Discovery】** タイトルバーをクリックし（ボックス内の他の部分をクリックしないこと）、**【Scenario diagram】** ウィンドウの**【Enterprise Discovery】** コネクタを選択します。
- 4 **【ツール／設定】** メニューを選択します。

- 5 ウィザード内のページに入力します。
 - 6 **【完了】** をクリックして入力事項を確定します。
- Asset Managerデータベースにどのデータが入力されるかを知るには、「edac-mobiledevices.scn」シナリオの詳細を確認します。

Asset Manager Automated Process Managerの検証を即時に起動する

定義した時間を待たずに、手動で検証モジュールを起動することもできます。手動で検証を実行するには、Asset Manager Automated Process Managerの**【アクション/起動】**メニューを選択します。

起動する検証モジュールのチェックボックスをオンにします。

Asset Manager Automated Process ManagerをWebブラウザで管理する

Windows用のAsset Managerインストールプログラムは、Asset Manager Automated Process ManagerをNTサービスとしてインストールします。

そのため、Windows NTでは、AssetManager Serverを次の2つの方法で起動できます。

- Asset Manager Automated Process Managerのグラフィカルインタフェースを使用する
- または、Asset Manager Automated Process ManagerをNTサービスとして起動する

Asset Manager Automated Process ManagerをNTサービスとして起動すると、Webブラウザを使って機能を制御できます。

このセクションでは、次の内容について説明します。

- Asset Manager Automated Process ManagerをNTサービスとして起動する
- Webブラウザ経由でAsset Manager Automated Process Managerサービスにアクセスする
- Webブラウザ経由でAsset Manager Automated Process Managerサービスを制御する

Asset Manager Automated Process Managerをサービスとして起動する

Windows NTにAsset Managerをインストールすると、次のようになります。

- Asset Manager Automated Process Managerはサービスとしてインストールされます。この時点ではサービスは起動しません。

- Asset Manager Automated Process ManagerサービスへのWebアクセスは無効です。

 **注意:**

Asset Manager Automated Process Managerを正しくインストールするには、以下の手順に従ってください。

- 1 (このサービスのインストール先コンピュータ上に) Windowsのユーザアカウントを作成します。

このアカウントには、Asset Manager Automated Process Managerサービスを起動するために必要な権限がなければなりません。

このアカウントの環境は、Asset Manager Automated Process Managerサービスのコンピュータ上にインストールされたDBMSのクライアント層の使用を、許可しなければなりません。

ローカルシステムアカウントは、デフォルトではシステムの環境変数にしかアクセスしないことを念頭に置いてください。

- 2 Asset Manager Automated Process Managerサービスをこのアカウント上にインストールします。

Asset Manager Automated Process ManagerサービスへのWebアクセスを有効にする

Webアクセスを有効にするには：

- データベースへの接続を選択します。
- [この接続をサービスモードで使用] オプションを選択します。

次の方法も可能です。

- 「amsrvcf.ini」 ファイルを編集します。
このファイルの場所：▶ 『**Asset Manager - インストールとアップグレード**』 ガイドの「**.iniおよび.cfgファイル**」の章。
- [GLOBAL] セクションで、[WebAdmin] キーの値を変更します。
 - WebAdmin = 1の場合は、Webアクセスが有効になります。
 - WebAdmin = 0の場合は、Webアクセスが無効になります。
- [GLOBAL] セクションで、Asset Manager Automated Process Managerサービスが使うTCP/IPポートの値を確認します。この値はWebPortキーに保存されており、デフォルトは82です。このポートが別のプログラムで既に使われている場合は値を変更します。

Asset Manager Automated Process Managerサービスを起動する

Asset Manager Automated Process Manager NTサービスを起動するには：

- 1 コントロールパネルの [サービス] をダブルクリックします。
- 2 Asset Manager Automated Process Managerサービスを選択します。
ここで、すぐにサービスを起動する場合は次の手順を実行します。

- **【開始】** をクリックします。Asset Manager Automated Process Manager NTサービスでは、**【スタートアップパラメータ】** フィールドに直接パラメータを指定しないことをお勧めします。

Asset Manager Automated Process Managerサービスを設定する場合は、次の手順を実行します。

- 1 **【スタートアップ】** ボタンをクリックします。
- 2 **【スタートアップの種類】** を次の中から選択します。
 - **【自動】** : Windowsを起動するとこのサービスが起動します。
 - **【手動】** : サービスを手動で起動する必要があります。Windows NTの**【サービス】** アプレットで**【開始】** をクリックしてください。
 - **【無効】** : このNTサービスを起動できません。

Webブラウザ経由でAsset Manager Automated Process Managerサービスにアクセスする

注意:

注意 : Asset Manager Automated Process ManagerサービスにWebブラウザを使ってアクセスするには、サービスを起動する必要があります。

Asset Manager Automated Process Managerサービスにアクセスするには :

- 1 普段使用しているWebブラウザを起動します。
- 2 Asset Manager Automated Process Managerサービスが起動しているコンピュータのアドレスの後に、Asset Manager Automated Process Managerサービスが使うTCP/IPポート番号を続けて入力します。コンピュータのアドレスとポートはコロン (:) で区切ります。

アドレスの例を次に示します。

- "http://colombo.taltek.com:82"
- "http://laguardia.taltek.com:800"

Asset Manager Automated Process Managerサービスが稼動するコンピュータのTCP/IPアドレスの後にポート番号を続けて入力することもできます。

例 :

"127.0.0.1:82".

- 3 ホームページが開きます。このページ上の**【接続】** ボタンをクリックします。
- 4 Asset Manager Automated Process Managerサービスへのアクセスを認証するウィンドウが表示されます。ここで、次の情報を入力します。
 - 1 ユーザ名 : WebAdmin
 - 2 ユーザ名「WebAdmin」のパスワード。デフォルトでは、このパスワードは空です。

Webブラウザ経由でAsset Manager Automated Process Managerサービスを制御する

このセクションでは、Asset Manager Automated Process Managerサービスに接続した後の処理で指定するコマンドについて説明します。

新しいデータベースに接続

このコマンドを使うと、次の処理が可能です。

- Asset Manager Serverデータベースに手動で接続します。
- Asset Manager Automated Process Managerの起動時に自動的にAsset Managerデータベースに接続するように設定します。手順は次のとおりです。
 - 1 **【起動時に再接続】** オプションをチェックします。
 - 2 Asset Manager Automated Process Managerサービスが自動的に再接続するデータベースの接続名を入力します。
 - 3 **【ログイン】** 名とパスワードを指定します。

注意:

「Amsrvcf.ini」ファイルの [Database] セクションの [AutoLogin] キーで指定して、Asset Manager Automated Process Managerサービスの自動接続を設定することもできます。AutoLogin = 0 : 自動接続は無効になります。AutoLogin = 1 : 自動接続は有効になります。

このファイルの場所 : ▶ 『Asset Manager - インストールとアップグレード』ガイドの「.iniおよび.cfgファイル」の章。

ステータス

このコマンドを使うと、Asset Manager Automated Process Managerログファイルの最新のメッセージを100件まで表示できます。これらのメッセージは、GUIバージョンのAsset Manager Automated Process Managerのメイン画面に表示されるメッセージに当たります。

表示されたメッセージをすべて消去する場合は、**【クリア】** をクリックします。

注意:

メッセージの最大表示数は変更できません。

設定

このコマンドを使うと、どのモジュールを検証するかを定義できます。

**注意:**

モジュールの検証スケジュールをWebから変更することはできません。変更するには、Asset Manager Automated Process Managerのグラフィカルユーザインタフェースから [オプション/設定] メニューを選択する必要があります。

トリガ

このコマンドを選択すると、直ちに特定の検証を起動します。

パスワード

このコマンドを使うと、「WebAdmin」のパスワードを変更できます。デフォルトでは、このパスワードは空です。

終了

このコマンドボタンをクリックすると、Asset Manager Automated Process Managerサービスへの接続を終了します。

**注意:**

Webアクセスがアイドル状態のときに、アクセスを自動的に切断するオプションがあります。このオプションは、「Amsrvcf.ini」ファイルの [SESSION] セクションの [TimeOut] キーで定義します。デフォルトでは10分に設定されています。

このファイルの場所 : ▶ 『**Asset Manager** - インストールとアップグレード』ガイドの「.iniおよび.cfgファイル」の章。

14 メッセージシステム

Asset Managerでは、次の2種類のメッセージを管理できます。

- Asset Managerから発行され、内部メッセージシステムを介してAsset Managerデータベースに送信されたメッセージ
- Asset Managerで作成され、外部メッセージシステムを介して送信されたメッセージ

この章では、使用するプロトコルのタイプに応じたメッセージシステムの設定方法を説明します。

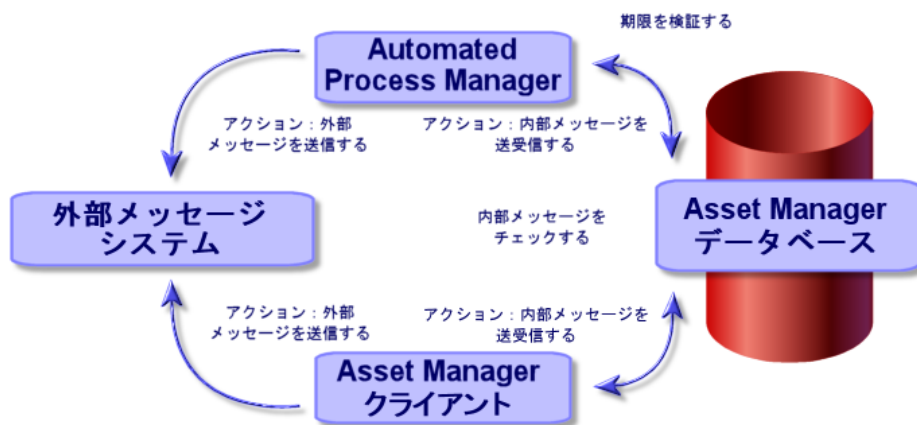
メッセージシステムの仕組み

Asset Managerは、以下のプロトコルを使用してメッセージの送信を管理します。

- AM (Asset Manager)
- SMTP
- MAPI
- VIM

受信時には、Asset ManagerはAM (Asset Manager) タイプのメッセージのみを管理します。

図 14.1. 内部メッセージシステムの仕組み



メッセージの送信、受信、閲覧方法については、『高度な使い方』ガイドの「メッセージ」のセクションを参照してください。

Asset Managerでメッセージシステムを指定する

使用するプロトコルに応じてAsset Managerの設定を行います。

Asset ManagerとAsset Manager Automated Process Managerで外部メッセージシステムにメッセージを送信するには、以下の操作が必要です。

- 従業員の詳細画面で指定する。
- アクションのフィールドに入力する。
- Asset Manager Automated Process Managerを設定して起動する。
- 新規メッセージをチェックする周期を設定する。

Asset Managerの【編集/オプション】メニューの【メッセージ】項目で設定します。

警告:

Asset Managerの使用中には、現行のメッセージプロトコル以外のプロトコルを使用することはできません。

内部メッセージプロトコル「Asset Manager (AM)」のみを他のメッセージプロトコルと共に使用できます。

複数の受信者

複数の受信者にメッセージを送信する場合には、使用するプロトコルに関係なく以下のシンタックスを使用します。

```
SMTP:[name@address.domain], SMTP:[name2@address.domain]
```

例：

```
SMTP:jdoe@hp.com, SMTP:administrator@prgn.com
```

SMTPプロトコル

従業員の詳細

メッセージを送受信するには、送信者のアカウント（[部署と従業員]（amEmplDept）テーブルの[メッセージ]タブページ）と受信者のメッセージアドレス（アクションで指定）をAsset Managerで指定しておく必要があります。

発信者を設定する

- 発信者のアカウントは以下のフィールドで指定します。
 - [アカウント]（SQL名：MailLogin）フィールド：以下のように指定します。

```
SMTP:[名前]
```

- [パスワード]（SQL名：MailPassword）フィールド：SMTPサーバがログインが必要とする場合以外は、空欄にします。

部署名/姓: 本多	名前: 京介
役職: 施設保全課長	性別: 男性
部署: 管理	
プロフィール	メッセージ
アカウント: SMTP'honda	アプリケーションサービス
パスワード: *****	契約

- 以下のシンタックスを使って、[全般]タブページの[電子メール]フィールドに入力します。

SMTP:[name@address.domain]

全般 グループ 研修 ポートフォリオ コスト ファイナ

住所

場所: /札幌事務所/ビル02/2階/001- 事務所/

電話番号: 011-5572-9000 ファックス: 011-5572-9099

携帯電話: 011-5572-9000 自宅電話: 011-5572-9000

電子メール: SMTP:honda@taltek.co.jp

フィールド1:

フィールド2:

フィールド3:

ID番号: DEMO-M031 バーコード: DEMO-U004

雇用日: 2010/12/21 退職日:

コメント:

受信者を設定する

メッセージの受信者は、メッセージタイプのアクションの詳細の【受信者】（SQL名：MsgTo）フィールドに指定します。電子メールは、プレーンテキストまたはHTML形式のどちらかで送信できます。

- 以下の例のように【受信者】（SQL名：MsgTo）フィールドにアドレスを入力します（必要に応じて【Cc】フィールドと【Bcc】フィールドにも入力します）。

SMTP:[name@address.domain]

または特殊文字列

名前: Email

コンテキスト: 部署と従業員 (amEmpDept)

タイプ: メッセージ テスト(T)...

説明: メッセージ 履歴 ドキュメント

参照オブジェクト:

優先度: 普通

配信済通知

HTML形式

受信者: [EMail]

Cc:

Bcc:

件名:

メッセージ

この例では、従業員の詳細画面 / [全般] タブページ / [電子メール] フィールドの値が、[受信者] フィールドに入力されます。メッセージのコピーが管理者に送信されます。

- 電子メールをHTML形式で送信する場合は、[HTML形式] (bHTML) オプションを選択して、メッセージの内容をHTMLタグで囲んで入力します。



HTML形式は、SMTPプロトコルで送信するメッセージのみに適用できます。

「mail.ini」内のメッセージパラメータ

Asset Managerでメッセージシステムを使用するには、「mail.ini」ファイルを手動で作成し、次のコマンドラインを追加する必要があります。

```
[mail] SMTP=1 SMTPserver=[server name] email=[messaging address@domain name] displayname=[user's full name]
```

例

```
[mail] SMTP=1 SMTPserver=mail.prgn.com email=sblaine@prgn.com displayname=Steven Blaine
```



「mail.ini」は、特定のフォルダに作成する必要があります。このファイルの場所については、▶ 『Asset Manager - インストールとアップグレード』ガイドの「.iniおよび.cfgファイル」の章を参照してください。これはWindowsとUNIXの両方のプラットフォームに適用されます。

MAPIプロトコル

従業員の詳細

メッセージを送受信するには、送信者のアカウント（[部署と従業員] テーブルの [メッセージ] タブページ）と受信者のメッセージアドレス（アクションで指定）をAsset Managerで指定しておく必要があります。

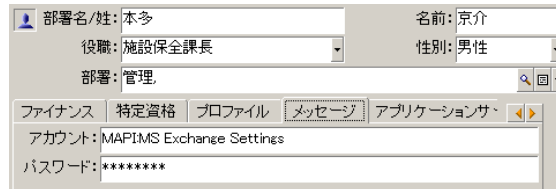
発信者を設定する

- 送信者のプロファイルは以下のフィールドで指定します。
 - [アカウント] (SQL名: MailLogin) フィールド: 以下のように入力します。

```
MAPI:[ユーザプロファイル名]
```

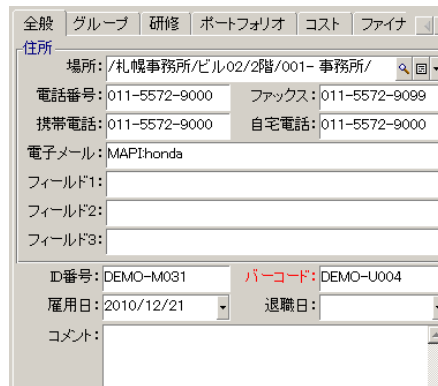
ユーザプロフィールを確認するには、Windowsのコントロールパネルで、
[メール] セクションを開き、[プロフィールの表示] ボタンをクリック
します。

- [パスワード] (SQL名: MailPassword) フィールド: メール用のパスワードを入力します。



- 以下のシンタックスを使って、[全般] タブページの [電子メール] フィールドに入力します。

[MAPI:[メールボックス名]]



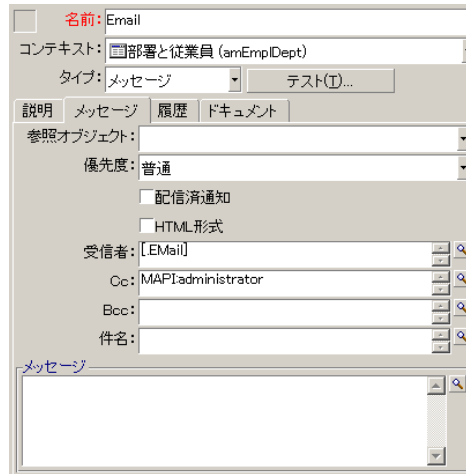
受信者を設定する

メッセージの受信者は、メッセージタイプのアクションの詳細の [受信者] (SQL名: MsgTo) フィールドに指定します。

以下の例のように [受信者] (SQL名: MsgTo) フィールドにアドレスを入力します (必要に応じて [Cc] フィールドと [Bcc] フィールドにも入力します)。

[MAPI:[メールボックス名]]

または特殊文字列



この例では、従業員の詳細画面 / [全般] タブページ / [電子メール] フィールドの値が、[受信者] フィールドに入力されます。

「mail.ini」内のメッセージパラメータ

Asset Managerでメッセージシステムを使用するには、「mail.ini」ファイルを手動で作成し、次のコマンドラインを追加する必要があります。

```
MAPI=1 MAPIX=1
```

注意:

「mail.ini」は、特定のフォルダに作成する必要があります。このファイルの場所については、▶ 『Asset Manager - インストールとアップグレード』ガイドの「.iniおよび.cfgファイル」の章を参照してください。これはWindowsとUNIXの両方のプラットフォームに適用されます。

VIMプロトコル

従業員の詳細

メッセージを送受信するには、送信者のアカウント（[部署と従業員] テーブルの[メッセージ] タブページ）と受信者のメッセージアドレス（アクションで指定）をAsset Managerで指定しておく必要があります。

発信者を設定する

- 発信者のアカウントは以下のフィールドで指定します。
 - **【アカウント】** (SQL名 : MailLogin) フィールド : 以下のように指定します。

VIM:[名前/ドメイン]

部署名/姓: 金田	名前: 雅美
役職: 交換室主任	性別: 女性
部署: 管理	
アカウント: VIMkaneda/TALTEK	パスワード: *****

- **【パスワード】** (SQL名 : MailPassword) フィールド : メール用のパスワードを入力します。
- 以下のシンタックスを使って、**【全般】** タブページの **【電子メール】** フィールドに入力します。

VIM:[名前/ドメイン]

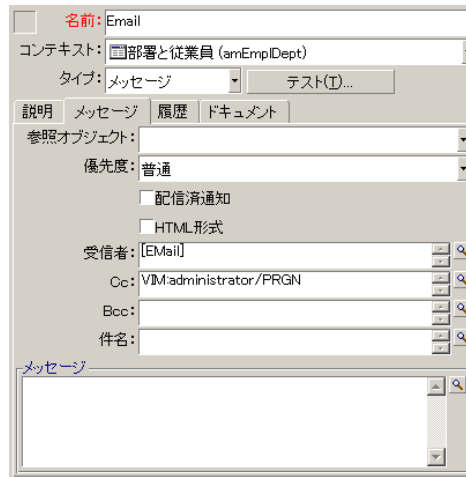
住所	場所: /札幌事務所/ビル01/ホール/001 - 受付/
電話番号: 011-5572-9000	ファックス: 011-5572-9099
携帯電話: 011-5572-9000	自宅電話: 011-5572-9000
電子メール: VIMkaneda/TALTEK	
フィールド1:	
フィールド2:	
フィールド3:	
ID番号: DEMO-M041	バーコード: DEMO-U005
雇用日: 2010/12/21	退職日:
コメント:	

受信者を設定する

メッセージの受信者は、メッセージタイプのアクションの詳細の **【受信者】** フィールドに指定します。

以下の例のように【受信者】フィールドにアドレスを入力します（必要に応じて【Cc】フィールドと【Bcc】フィールドにも入力します）。

VIM:[名前/ドメイン]



この例では、従業員の詳細画面／【全般】タブページ／【電子メール】フィールドの値が、【受信者】フィールドに入力されます。メッセージのコピーが管理者に送信されます。

Windowsの設定

Asset ManagerがVIMプロトコルをサポートできるようにするには、Windowsの「Path」を設定する必要があります。Windowsのバージョンに応じて、手順が少し変わる可能性があります。

- 1 「vim32.dll」ファイル（Lotus Notesフォルダにある）へのアクセスパスを書き留めます。
- 2 コンピュータのPath変数を編集して、このディレクトリへのパスを含めるようにします。

「mail.ini」内のメッセージパラメータ

Asset Managerでメッセージシステムを使用するには、「mail.ini」ファイルを手動で作成し、次のコマンドラインを追加する必要があります。

SMI=1

 注意:

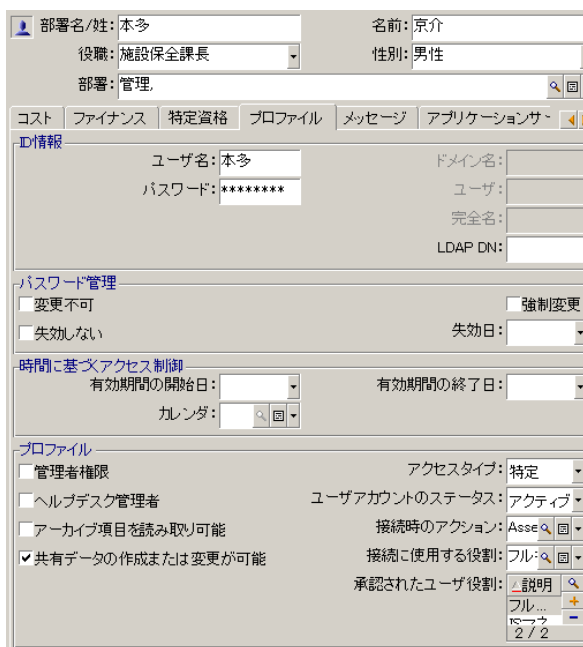
「mail.ini」は、特定のフォルダに作成する必要があります。このファイルの場所については、▶『**Asset Manager** - インストールとアップグレード』ガイドの「**.ini**および**.cfg**ファイル」の章を参照してください。これはWindowsとUNIXの両方のプラットフォームに適用されます。

AMプロトコル

発信者を設定する

内部メッセージを送信できるようにするには、[部署と従業員] (SQL名: amEmplDept) テーブルで以下の情報を指定する必要があります。

- [部署と従業員] (amEmplDept) テーブル ([プロファイル] タブ) でユーザのログインを指定します。



- 以下のシンタックスを使って、従業員の詳細画面 / [全般] タブページ / [電子メール] フィールドに入力します。

AM:[ユーザログイン]

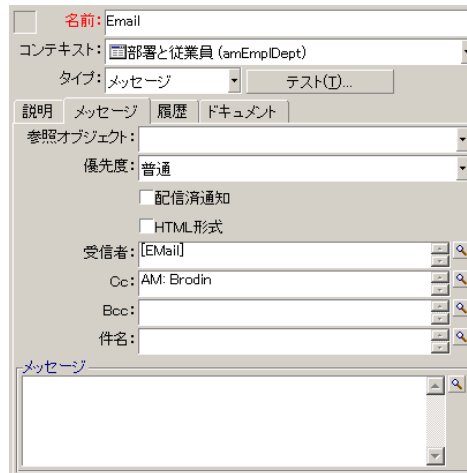
受信者を設定する

メッセージタイプのアクションの詳細でメッセージの受信者を指定します。

- 以下の例のように【受信者】（SQL名：MsgTo）フィールドにアドレスを入力します（必要に応じて【Cc】フィールドと【Bcc】フィールドにも入力します）。

AM:[受信者のログイン]

または特殊文字列



The screenshot shows a configuration window for an email message. The 'Name' field is 'Email'. The 'Context' is '部署と従業員 (amEmpDept)'. The 'Type' is 'メッセージ'. The 'Priority' is '普通'. The 'Recipient' field is '[EMail]'. The 'Cc' field is 'AM: Brodin'. The 'Bcc' field is empty. The 'Subject' field is empty. The 'Message' field is empty.

この例では、従業員の詳細画面／【全般】タブページ／【電子メール】フィールドの値が、【受信者】フィールドに入力されます。

メッセージを送受信する

メッセージを送信する方法

メッセージは、【メッセージ】タイプのアクションを介して作成されます。

メッセージを送信するには、まずメッセージを送信するアクションを作成する必要があります。

HTML形式のメッセージを送信するには、メッセージタイプのアクションの【メッセージ】タブページで【HTML形式】オプション（SQL名：bHTML）を選択します。



注意:

HTML形式は、SMTPプロトコルのメッセージのみに適用できます。

メッセージを送信するアクションは、以下のようにいくつかの異なる方法でトリガされます。

- 手動で
 - Windowsクライアント：[ツール/アクション] メニューに表示されるリストのアクションを選択するか、コンテキストテーブル（アクションの作成時に入力される [コンテキスト] フィールドによって定義）画面のショートカットメニューからアクションを選択してトリガする。
 - Webクライアント：[アクション] ドロップダウンリストからアクションを選択するか、コンテキストテーブル画面のショートカットメニューからアクションを選択してトリガする。
- Asset Manager Automated Process Managerによって自動的にトリガされる。
- Asset Managerによって自動的にトリガされる。
[メッセージ] タイプのアクションの作成については、『高度な使い方』ガイドの「アクション」の章、「アクションの作成/[メッセージ] タブページに入力する」のセクションを参照してください。

メッセージを参照する方法

内部メッセージシステムで受信したメッセージを参照する

エージェントは新しいメッセージが到着しているかどうかを確認し、ユーザに新着メッセージの受信を通知します。

受け取ったメッセージは次の方法で参照できます。

- Windowsクライアント：[ツール/メッセージ] メニューを使う。
- WindowsクライアントおよびWebクライアント：新着メッセージを通知するメッセージボックスで参照する。

外部メッセージシステムで受信したメッセージを参照する

メッセージの受信者は、外部メッセージシステムのクライアントプログラム上で通常どおりメッセージを参照できます。

配信済通知



注意:

Asset Managerの内部メッセージシステムや、MAPIまたはSMTPメッセージシステムで送信したメッセージの場合は、配信済通知を受信できません。


配信済通知を受信するには、[メッセージ] タイプのアクションの詳細画面で [配信済通知] (SQL名: bAcknowledgment) チェックボックスをオンにします。

この通知は、Asset Managerデータベースを開いた従業員 ([部署と従業員] テーブル) の [全般] タブページ / [電子メール] (SQL名: EMail) フィールドに指定したアドレスに送信されます。

一般的な接続エラー

メッセージの送信時に問題が発生した場合は、問題を通知するエラーメッセージが管理者に送信されます。

メッセージシステムとの接続を確認するテスト

- 1 Asset Manager Automated Process Managerを起動します。
- 2 データベースに接続します。
- 3  をクリックします。

問題が発生した場合に実行するテスト

- 1 新しい [メッセージ] [タイプ] (SQL名: seActionType) のアクションを作成し、受取人を指定します。
- 2 [ツール/アクション] メニューからアクションをトリガします。
- 3 受信者がメッセージを受信したことで、ルータからメッセージシステムにエラーメッセージ (不明な受信者) が送信されていないことを検証します。
- 4 エラーメッセージがある場合は、原因を突き止めます。

トラブルシューティング

「メッセージシステム 'XXX' との接続: メッセージシステムが指定されていません。 [従業員の詳細] ダイアログボックスの [プロファイル] タブで、メッセージのアカウントを確認してください。」

このエラーメッセージが表示される場合は、従業員の詳細画面の [メッセージ] タブページ [アカウント] (SQL名: MailLogin) フィールドの値の先頭に、次の文字列を追加する必要があります。

- MAPI: MAPI標準のメッセージシステム (Microsoft Outlook、Microsoft Exchangeなど) を使う場合

- VIM : VIM標準のメッセージシステム (Lotus Notes、CCMailなど) を使う場合
- SMTP : SMTP標準のメッセージシステム (インターネット標準) を使う場合

「メッセージシステム 'XXX' に接続できません。」

このエラーメッセージが表示される場合は、従業員の詳細画面の [メッセージ] タブページ / [アカウント] フィールドで接頭語の「MAPI:」または「VIM:」が正しく指定されていても、アカウント名が正しくないことを示します。正しく入力されているかどうか確認してください。

「メッセージのアカウント 'VIM' : パスワードを空白にしておくことはできません。」

VIMメッセージシステムを使う場合は、従業員の詳細画面の [メッセージ] タブページ / [パスワード] (SQL名: MailPassword) フィールドにパスワードを指定する必要があります。パスワードを空にすることはできません。

「メッセージのアカウント 'XXX' : パスワードが正しくありません。」

このエラーメッセージが表示される場合は、従業員の詳細画面の [メッセージ] タブページ / [パスワード] フィールドに指定したパスワードが正しくありません。

「'XXX' にメッセージを送信できません : メッセージシステムは使用不可能です。」

これにより「mail.ini」ファイルの問題が明らかになります。

Asset ManagerでMAPIメッセージシステムを正しく使うには、「mail.ini」ファイルの [Mail] セクションに次の行を記述する必要があります。

MAPI=1

MAPIX=1

Asset ManagerでVIMメッセージシステムを正しく使うには、「mail.ini」ファイルの [Mail] セクションに次の行を記述する必要があります。

SMI=1

Asset ManagerでSMTPメッセージシステムを正しく使うには、「mail.ini」ファイルの [Mail] セクションに次の行を記述する必要があります。

SMTP=1

SMTPServer=<メール送信サーバ>

次の行はオプションです。

SMTPPort=<メール送信サーバのポート番号> (デフォルトでは25)

SMTPTimeOut=<タイムアウト値> (デフォルトでは20秒)

これらのメッセージシステムの行を一緒に指定することもできます。

上記のいずれかの行がない場合または値が0に設定されている場合は、対応するメッセージシステムが正しく機能することを確認する必要があります。確認する

には、MAPIの場合はMicrosoft Internet Mail、VIMの場合はLotus Notesなどのプログラムを使います。メッセージシステムが正しく機能していて、次のような状況でない場合は、「mail.ini」ファイルの [Mail] セクションを上記のように変更してください。

 **警告:**

MAPIが1に設定されていても、MAPIXが1でない場合、メッセージシステムは拡張MAPIをサポートできません。まずこの点を確認してください。メッセージシステムが拡張MAPIと互換性がない場合、Asset Managerは正しく機能しません。

「VIMセッションを開始するときのエラー：パスワードが必要です。」

VIM標準のメッセージシステムには、パスワードが必要です。メッセージシステムにパスワードを追加し、そのパスワードをAsset Managerで従業員の詳細画面の [メッセージ] タブページ / [パスワード] (SQL名: MailPassword) フィールドに指定します。

「VIMセッションを開始するときのエラー：パスワードが正しくありません。」

パスワードが間違っています。従業員の詳細画面の [メッセージ] タブページ / [パスワード] (SQL名: MailPassword) フィールドの値を変更してください。

「メッセージコンテナが壊れています」または「ステーションの設定が無効です」

VIMプロトコルは、Lotus Notesの「notes.ini」ファイルに含まれている名前と場所に関する情報を取得します。情報が正しくない場合、メッセージを送信できません。このファイルを編集して、パラメータを変更してください。

15 スケジュールのグラフィック表示画面のカスタマイズ

概要

スケジュールのグラフィック表示のページを、以下の箇所に追加することができます。

- テーブルの詳細画面（Asset Manager Application Designerを使って追加する）
- ウィザード（Asset Managerを使って追加する）

追加は、少なくとも以下の情報がある場合に可能になります。

- 開始を格納するフィールド（日付のみまたは日付と時刻）
- 終了を格納するフィールド（日付のみまたは日付と時刻）

スケジュールのグラフィック表示のページ機能をカスタマイズする

スケジュールのグラフィック表示の全ページの機能を設定するオプションは、**【編集／オプション】**メニューの**【表示／スケジュールのグラフィック表示】**項目で定義できます。

注意:

土曜日と日曜日は休日として表示されます。これは変更不可能です。

テーブルにスケジュールのグラフィック表示のページを追加する

テーブルにスケジュールのグラフィック表示のページを追加する

Asset Manager Application Designerを使うと、詳細画面にスケジュールのグラフィック表示のページを追加できます（または詳細画面からページを削除できます）。

- 1 Asset Manager Application Designerを起動します。
- 2 変更するデータベースへ接続します（【ファイル／開く】メニューの【既存のデータベースを開く】オプション）。
- 3 変更するテーブルを選択します。
- 4 テーブルのページ（【表示／ページ】メニュー）を表示します。
- 5 ページを追加します（【新規作成】ボタン）。
- 6 【全般】タブのフィールドにデータを入力します。
- 7 【内容】タブページを表示します。
- 8 以下の形式の文字列をタイプして、編集枠で新規フィールドを定義します。

```
PlannerViewer-[A] | StartDate=[B] | EndDate=[C] | StartBefore=[D] | EndBefore=[E] | StartAfter=[F] | EndAfter=[G] | ItemColor=[H] | GenerateColors=[I] | CompleteDates=[J] | LineLabel=[K] | ItemLabel=[L] | Progress=[M]
```

シンタックス：▶参考情報：スケジュールのグラフィック表示のページのパラメータのシンタックス [P. 289]

- 9 定義されたフィールドを追加します（▶ボタン）。
- 10 ページを保存します（【作成】ボタン）。
- 11 テーブルの詳細を表示します（【表示／画面】メニュー）。
- 12 上記で作成したページの追加先となる詳細画面を選択します。
- 13 【ページ】タブページを表示します。
- 14 上記で作成したページを、使用可能ページのリストから詳細ページのリストへ移動させ（▶ボタン）、他のページの中でこのページが占める位置を選択します。
- 15 変更された詳細画面を保存します（【変更】ボタン）。
- 16 変更事項をデータベース内で保存します（【ファイル／データベース構造の更新】メニュー）。



ヒント:

【変換ファイル】フィールドが表示されても、ここには何も入力しません。

実用例

- 1 Asset Manager Application Designerを起動します。
- 2 デモ用データベースへ接続します（ [ファイル/開く] メニューの [既存のデータベースを開く] ）。
- 3 [作業指示]（amWorkOrder）テーブルを選択します。
- 4 テーブルのページ（ [表示/ページ] メニュー）を表示します。
- 5 ページを追加します（ [新規作成] ボタン）。
- 6 [全般] タブの以下のフィールドに値を入力します。

フィールド	値
SQL名	pg.In_progress
タイトル	進行中

- 7 [内容] タブページを表示します。
- 8 以下の文字列をタイプして、編集枠で新規フィールドを定義します。

```
PlannerViewer-!WorkOrderId |StartDate=dtActualFixStart |EndDate=dtActualFixed |EndBefore=dtSchedFixed |StartBefore=dtSchedFixStart |ItemColor=!TechId |GenerateColors=true |CompleteDates=true |LineLabel=self |ItemLabel=Title
```

定義されたパラメータの値は以下のように解釈されます。

パラメータ	値	効果
PlannerViewer	!WorkOrderId	イベントは作業指示です。
StartDate	dtActualFixStart	イベントの開始は [作業開始日] フィールドで定義されます。
EndDate	dtActualFixed	イベントの終了は [作業完了日] フィールドで定義されます。
EndBefore	dtSchedFixed	[終了予定日] フィールドで定義された日時にイベントが終了しない場合、対応する丸印●と角印■は赤で表示されます。
StartBefore	dtSchedFixStart	[開始予定日] フィールドで定義された日時にイベントが開始しない場合、対応する丸印●と角印■は赤で表示されます。
GenerateColors	true	[技術者] フィールドの値に応じてイベントを表記する横棒
ItemColor	!TechId	□に、色が自動的に循環して割り当てられます。
CompleteDates	true	[作業開始日]（dtActualFixStart）フィールドまたは [作業完了日]（dtActualFixed）が空の場合、値は現在の日時に置き換えられます。
LineLabel	self	[作業指示]（amWorkOrder）テーブルの記述文字列によって、  のウィンドウの左側のリストにあるイベントを識別できるようになります。

パラメータ	値	効果
ItemLabel	Title	[タイトル] フィールドの値は、📍のリスト内のイベントを表す横棒 <input type="checkbox"/> 内に表示されます。

- 9 定義されたフィールドを追加します (▶ ボタン)。
- 10 ページを保存します ([作成] ボタン)。
- 11 テーブルの詳細を表示します ([表示/画面] メニュー)。
- 12 **sysamWorkOrder**の詳細を選択します。
- 13 [ページ] タブページを表示します。
- 14 ページ「**In progress**」を、使用可能ページのリストから詳細ページのリストへ移動させ (▶ ボタン)、他のページの中でこのページが占める位置を選択します。
- 15 変更された詳細画面を保存します ([変更] ボタン)。
- 16 変更事項をデータベース内で保存します ([ファイル/データベース構造の更新] メニュー)。



ヒント:

[変換ファイル] フィールドが表示されても、ここには何も入力しません。

- 17 Asset Managerを起動します。
- 18 デモ用データベースへ接続します ([ファイル/データベースに接続] メニュー)。



警告:

デモ用データベースに既に接続している場合は、切断してから再接続する必要があります。

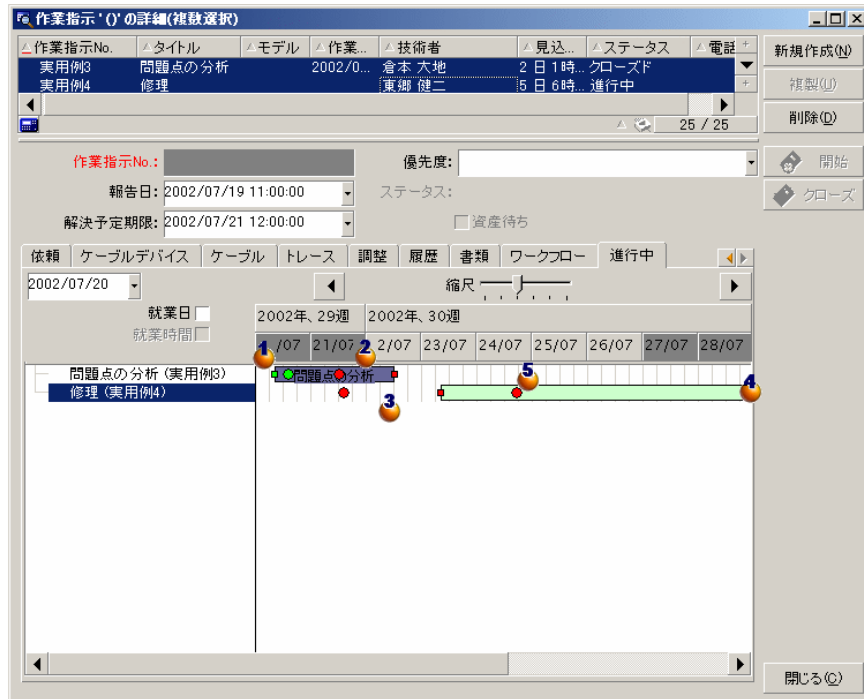
- 19 作業指示のリストを表示します ([資産ライフサイクル/作業指示/作業指示] メニュー)。
- 20 作業指示を2つ作成し、次のフィールドに値を入力します。

タブ	フィールドの 名前	フィールドの SQL名	作業指示1の値	作業指示2の値
	作業指示番号	WONo	CP3	CP4
	報告日	dtNotif	2002年7月19日 11:00:00	2002年7月19日 11:00:00
全般	タイプ	seType	内部メンテナンス	内部メンテナ ンス
全般	タイトル	Title	問題点の分析	修復

タブ	フィールドの 名前	フィールドの SQL名	作業指示1の値	作業指示2の値
トラッキング ／技術者	技術者	Contact	技術者を1人選択してください。	技術者をもう1人 選択してくださ い。
トラッキング ／スケジュール	開始予定日	dtSchedFixStart	2002年7月20日 14:00:00	2002年7月21日 14:00:00
トラッキング ／スケジュール	終了予定日	dtSchedFixed	2002年7月21日 12:00:00	2002年7月24日 17:00:00
トラッキング ／スケジュール	作業開始日	dtActualFixStart	2002年7月20日 08:00:00	2002年7月23日 08:00:00
トラッキング ／スケジュール	作業完了日	dtActualFixed	2002年7月22日 12:00:00	

- 21 実用例3と実用例4を両方選択します。
- 22 作成した【進行中】タブページを選択します。
- 23 【縮尺】カーソルおよび◀と▶ボタンを使って、実用例3の作業指示の全期間と、実用例4の作業指示の開始を表示します。

図 15.1. スケジュールのグラフィック表示 - 作業指示での例



- 1 作業指示「実用例3」：【作業開始日】は【開始予定日】よりも前であり、【開始予定日】はStartBeforeパラメータにより宣言されています。このため、丸印●と角印■は緑色になります。
- 2 作業指示「実用例3」：【作業完了日】は【終了予定日】よりも後であり、【終了予定日】はEndBeforeパラメータにより宣言されています。このため、丸印●と角印■は赤色になります。
- 3 作業指示「実用例4」：【作業開始日】は【開始予定日】よりも後であり、【開始予定日】はStartBeforeパラメータにより宣言されています。このため、丸印●と角印■は赤色になります。
- 4 作業指示「実用例4」：【作業完了日】には値が入力されていません。この値は現在の日付に置き換えられます。これはパラメータCompleteDatesの値が「true」であるためです。横棒□は現在の日付まで延長されます。

- 5 作業指示「**実用例3**」：現在の日付は【終了予定日】の後であり、【終了予定日】はEndBeforeパラメータにより宣言されています。このため、丸印●は赤色になります（角印■はこの例では表示されていません）。

ウィザードにスケジュールのグラフィック表示のページを追加する

ウィザードにスケジュールのグラフィック表示のページを追加する

Asset Managerのウィザードエディタを使うと、ウィザードにスケジュールのグラフィック表示のページを追加できます。

- 1 Asset Managerを起動します。
- 2 変更するデータベースへ接続します（【ファイル/データベースに接続】メニュー）。
- 3 アクションのリストを表示します（【ツール/アクション/編集】メニュー）。
- 4 変更するウィザードを選択します。または新規ウィザードを作成します。

注意:

【タイプ】（seActionType）フィールドの値は「ウィザード」でなければなりません。

- 5 【ウィザード】タブページを表示します。
- 6 既存のページを選択します。または新しいページを作成します。
- 7 このページに**PLANNER**ノードを追加します。
- 8 **PLANNER**ノードのパラメータに値を入力します。特に、Asset Manager APIライブラリの**amGeneratePlanningData(strTableSqlName, strProperties, strIds)** APIを参照しなければならない値を含む**CONTENT**ノードに、値を入力します。
 - **strTableSqlName**は、スケジュールのグラフィック表示のページに表示されるイベントを指定するための開始テーブルを指定します。
例："amWorkOrder"
 - **strProperties**は、スケジュールのグラフィック表示のページのパラメータのリストを、以下の形式で提供します。

```
"MainField=[A] | StartDate=[B] | EndDate=[C] | StartBefore=[D] | EndBefore=[E] | StartAfter=[F] | EndAfter=[G] | ItemColor=[H] | GenerateColors=[I] | CompleteDates=[J] | LineLabel=[K] | ItemLabel=[L] | WhereCond=[N] | OrderBy=[O] | Progress=[M]"
```

シンタックス：▶ 参考情報：スケジュールのグラフィック表示のページのパラメータのシンタックス [P. 289]

- **strIds**は、ウィザードによって返されるレコードで、そのイベントの表示を希望するもののリストを識別します。

 **注意:**

ウィザードが表示するスケジュールのグラフィック表示のページでは、表記されているイベントを変更することはできません。


実用例

- 1 Asset Managerを起動します。
- 2 デモ用データベースへ接続します（[ファイル/データベースに接続] メニュー）。
- 3 アクションのリストを表示します（[ツール/アクション/編集] メニュー）。
- 4 新規ウィザードを作成します。
- 5 次のフィールドに値を入力します。

フィールドの名前	フィールドのSQL名	値
名前	Name	CP
コンテキスト	ContextTable	(テーブルなし)
タイプ	seActionType	ウィザード

- 6 [ウィザード] タブページを表示します。
- 7 **PAGE1**を選択します。
- 8 このページに**PLANNER**ノードを追加します（右クリックし、[Edition/**PLANNER**] ショートカットメニューを選択します）。
- 9 **PLANNER**ノードの**CONTENT**パラメータに以下の値を入力します。

```
RetVal = amGeneratePlanningData("amWorkOrder","MainField=lWorkOrderId | StartDate=dtActualFixStart | EndDate=dtActualFixed | EndBefore=dtSchedFixed | StartBefore=dtSchedFixStart | ItemColor=lTechId | GenerateColors=true | CompleteDates=true | LineLabel=self | ItemLabel=Title | WhereCond=seType\=0 | OrderBy=Technician.Name, WONo", "")
```

- 10 ウィザードを保存します（[作成] ボタン）。
- 11 ウィザードをテストします（ アイコン）。

参考情報：スケジュールのグラフィック表示のページのパラメータのシンタックス

このセクションでは、スケジュールのグラフィック表示のページのパラメータを入力する方法を説明します。

スケジュールのグラフィック表示のページを、以下の箇所に追加することができます。

- テーブルの詳細画面（Asset Manager Application Designerを使って追加する）
使用可能なパラメータのシンタックスとリスト：▶テーブルにスケジュールのグラフィック表示のページを追加する [P. 282]
- ウィザードのページ（Asset Managerを使って追加する）
使用可能なパラメータのシンタックスとリスト：▶テーブルにスケジュールのグラフィック表示のページを追加する [P. 282]

PlannerViewer-[A]（テーブル）またはMainField=[A]（ウィザード）

用途	表示するイベントを指定します。
値	<ul style="list-style-type: none">■ イベントが直接、テーブルの一部を成す場合 <div style="border: 1px solid black; padding: 2px; margin: 5px 0;"><テーブルの主キー></div>他のパラメータにコンテキストの役目を果たすテーブルは次の通りです。<ul style="list-style-type: none">■ スケジュールのグラフィック表示のページの追加先テーブル■ ウィザードを定義する amGeneratePlanningData(strTableSqlName, strProperties, strIds) APIのstrTableSqlNameテーブル■ イベントが、リンクしているテーブルの一部を成す場合、 <div style="border: 1px solid black; padding: 2px; margin: 5px 0;"><リンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名></div>他のパラメータに対してコンテキストの役目を果たすテーブルは、リンクNのターゲットテーブルになります。 例外：LineLabel=[K]パラメータでは、リンクNのソーステーブルがコンテキストの役目を果たします。
例1	【作業指示】（SQL名：amWorkOrder）テーブルへスケジュールのグラフィック表示のページを追加する際に、作業指示自体も表示する場合は、[A] は IWorkOrderId になります。
例2	【作業指示】（SQL名：amWorkOrder）テーブルへスケジュールのグラフィック表示のページを追加する際に、内部技術者の不在を表示する場合は、[A] は Technician.Absences になります。
デフォルト値	なし

必須 はい

StartDate=[B]

用途	イベントの開始を定義します。
値	次の方法で指定されている日付または日付と時刻タイプのフィールド <ul style="list-style-type: none">■ <コンテキストテーブルの直接フィールドのSQL名>■ <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名>
	注意: コンテキストテーブルは、パラメータPlannerViewer-[A]（テーブル）またはMainField=[A]（ウィザード）によって定義されます。
例	[不在]（amAbsence）テーブルの[開始日]フィールドでは、[B]は dtOut になります。
デフォルト値	なし
必須	はい

EndDate=[C]

用途	イベントの終了を定義します。
値	次の方法で指定されている日付または日付と時刻タイプのフィールド <ul style="list-style-type: none">■ <コンテキストテーブルの直接フィールドのSQL名>■ <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名>
	注意: コンテキストテーブルは、パラメータPlannerViewer-[A]（テーブル）またはMainField=[A]（ウィザード）によって定義されます。
例	[作業指示]（amWorkOrder）テーブルの[終了予定日]フィールドでは、[C]は dtSchedFixed になります。
デフォルト値	なし
必須	はい

StartBefore=[D]

用途	イベントの開始時が越してはならない期限を定義します。
注意事項	パラメータStartAfterとStartBeforeを同時に使用してはなりません。

値	<p>次の方法で指定されている日付または日付と時刻タイプのフィールド</p> <ul style="list-style-type: none"> ■ <コンテキストテーブルの直接フィールドのSQL名> ■ <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名> <p>注意:</p> <p>コンテキストテーブルは、パラメータPlannerViewer-[A] (テーブル) または MainField=[A] (ウィザード) によって定義されます。</p>
例	<p>【作業指示】 (amWorkOrder) テーブルの [解決予定期限] フィールドでは、 [D] はdtResolLimitになります。</p>
デフォルト値	なし
必須	いいえ

EndBefore=[E]

用途	イベントの終了時が越してはならない期限を定義します。
注意事項	パラメータEndAfterとEndBeforeを同時に使用してはなりません。
値	<p>次の方法で指定されている日付または日付と時刻タイプのフィールド</p> <ul style="list-style-type: none"> ■ <コンテキストテーブルの直接フィールドのSQL名> ■ <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名> <p>注意:</p> <p>コンテキストテーブルは、パラメータPlannerViewer-[A] (テーブル) または MainField=[A] (ウィザード) によって定義されます。</p>
例	<p>【作業指示】 (amWorkOrder) テーブルの [終了予定日] フィールドでは、 [E] はdtSchedFixedになります。</p>
デフォルト値	なし
必須	いいえ

StartAfter=[F]

用途	期限を定義してイベントがその後を開始するようにします。
注意事項	パラメータStartBeforeとStartAfterを同時に使用してはなりません。
値	<p>次の方法で指定されている日付または日付と時刻タイプのフィールド</p> <ul style="list-style-type: none"> ■ <コンテキストテーブルの直接フィールドのSQL名> ■ <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名> <p>注意:</p> <p>コンテキストテーブルは、パラメータPlannerViewer-[A] (テーブル) または MainField=[A] (ウィザード) によって定義されます。</p>
例	<p>【作業指示】 (amWorkOrder) テーブルの [報告日] フィールドでは、 [F] はdtNotifになります。</p>

デフォルト値	なし
必須	いいえ

EndAfter=[G]

用途	期限を定義してイベントがその後に終了するようにします。
注意事項	パラメータEndBeforeとEndAfterを同時に使用してはなりません。
値	次の方法で指定されている日付または日付と時刻タイプのフィールド <ul style="list-style-type: none"> ■ <コンテキストテーブルの直接フィールドのSQL名> ■ <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名>

注意:

コンテキストテーブルは、パラメータPlannerViewer-[A]（テーブル）またはMainField=[A]（ウィザード）によって定義されます。

デフォルト値	なし
必須	いいえ

GenerateColors=[I]

用途	イベントを表す横棒 <input type="checkbox"/> を彩色するために、色が自動的に使用されるべきかどうかを指定します。
注意事項	他の色を生成するために参照される基準色は、パラメータItemColor=[H]で定義されます。
値	<ul style="list-style-type: none"> ■ true : 参照色の値が変更するたびに、色は自動的に循環して変更されます。 ■ false : 色は、あるフィールドによってRGBフォーマットで定義されます。

デフォルト値	true
必須	いいえ

ItemColor=[H]

用途	イベントを表す横棒 <input type="checkbox"/> を彩色するために使用する参照色を決定します。
注意事項	[H] がとる値は、GenerateColors=[I]パラメータに指定された値に応じて変化します。

GenerateColors=[I] [H] は、横棒の色を条件付けるフィールドまたはリンクを指定します。このフィールドまたはリンクの値が変化すると、横棒の色は自動的に循環して変化します。

が**true**である場合の値

[H] の形式は次のうちの1つになります。

- <コンテキストテーブルの直接フィールドのSQL名>
- <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>
- <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名>

注意:

コンテキストテーブルは、パラメータPlannerViewer-[A] (テーブル) または MainField=[A] (ウィザード) によって定義されます。

GenerateColors=[I] [H] は、ボックスの色をRGBフォーマットで格納する整数 (32ビット) のタイプを指定します。

が**false**である場合の値

[H] の形式は次のうちの1つになります。

- <コンテキストテーブルの直接フィールドのSQL名>
- <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名>

注意:

コンテキストテーブルは、パラメータPlannerViewer-[A] (テーブル) または MainField=[A] (ウィザード) によって定義されます。

ヒント:

色のRGB値を決定する方法については、『はじめに』ガイドの「AssetManagerを初めて使用する」の章、「レコードリスト」のセクション、「メインリストを設定する」/「[リストの設定] ショートカットメニューを使う」を参照してください。

デフォルト値 なし
必須 はい

CompleteDates=[J]

用途 StartDateとEndDateで定義されているフィールドの内、最低でも1つが入力されていない場合に実行する操作を定義します。

使用例 進行中の作業指示の管理


値 ■ **true** : これらのフィールドに日付が入力されていない場合は、現在の日付に置き換えられます。
 ■ **false** : これらのフィールドに日付が入力されていない場合、日付は空のままになります。

デフォルト値 false


必須

いいえ

ItemLabel=[L]

用途	 のリスト内でイベントを表記する横棒で 사용되는タイトルを識別します。
値	<ul style="list-style-type: none">■ self : タイトルは、テーブルの記述文字列の値になります。■ <コンテキストテーブルの直接フィールドのSQL名> : タイトルはフィールド値になります。■ <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名> : タイトルはフィールド値になります。■ <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名> : タイトルはフィールド値になります。 <p>注意:</p> <p>コンテキストテーブルは、パラメータPlannerViewer-[A] (テーブル) または MainField=[A] (ウィザード) によって定義されます。</p>
デフォルト値	なし
必須	いいえ

LineLabel=[K]

用途	 ウィンドウの左にあるイベントのリストで、イベントを識別するために使用されるタイトルを識別します。
値	<ul style="list-style-type: none">■ self : タイトルは、テーブルの記述文字列の値になります。■ <PLANNERビューページを追加するテーブルの直接フィールドのSQL名> : ラベルはフィールド値になります。■ <PLANNERビューページを追加するテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名> : ラベルは、リンクNテーブルの記述文字列の値になります。■ <PLANNERビューページを追加するリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名> : ラベルはフィールド値になります。 <p>注意:</p> <p>コンテキストテーブルは、パラメータPlannerViewer-[A] (テーブル) または MainField=[A] (ウィザード) によって定義されます。</p>
デフォルト値	<ul style="list-style-type: none">■ [A] がテーブルの主キーである場合 : テーブルの記述文字列■ [A] がリンクの連結である場合 : 最終リンクのテーブルの記述文字列

必須 いいえ

Progress=[M]

用途	イベントの進行割合を記録するフィールドを識別します。 このフィールドのユーザタイプは「パーセント」でなければなりません。 注意: Asset Manager Application Designerで新しいフィールドを作成することができます。 注意: 特殊フィールドは使用できません。
値	<ul style="list-style-type: none">■ <テーブルの直接フィールドのSQL名>■ <リンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名>
デフォルト値	なし
必須	いいえ

WhereCond=[N]

用途	表示するイベントにフィルタを適用します。
値	AQL言語で表現された条件 AQL条件は区切り文字で囲まれていません。 しかし、「=」と「 」の前には、エスケープ文字「\」が必要です。
例	【作業指示】 (amWorkorder) テーブルで、【タイプ】フィールドが「内部メンテナンス」であるレコードのみを保持する場合、 [N] はseType\ \=0 になります。
デフォルト値	なし
必須	いいえ

OrderBy=[O]

用途 表示するイベントを並べ替えます。

値

文字「,」で区切られたフィールドまたはリンクのリスト
フィールドは次のいずれかのシンタックスを使って表現されます。

- <コンテキストテーブルの直接フィールドのSQL名>
- <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名>

リンクは次のいずれかのシンタックスを使って表現されます。

- <コンテキストテーブルのリンク1のSQL名> : これはリンクレコードの主キーの値を戻します。
- <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.<フィールドのSQL名>
- <コンテキストテーブルのリンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>.self

例

【作業指示】 (amWorkOrder) テーブルヘスケジュールのグラフィック表示のページを追加する際に、技術者で作業指示を並べ替える場合は、作業指示番号 [O] は **Technician.Name, WONo** になります。

デフォルト値

なし

必須

いいえ

16 Asset ManagerをDDEサーバとして使用する

このセクションでは、Asset ManagerをDDEサーバとして使う場合に、認識されるDDEコールについて詳しく説明します。
理論的な説明の後に、実際のDDEコールの使用例を示します。

DDEサーバの定義

DDEはDynamic Data Exchangeの略です。Windows上の複数のアプリケーション間でデータを交換するためのメカニズムで、動的データ交換ともいいます。ここでは、他のアプリケーションからAsset Managerコマンドを実行するために使うDDEのことを指します。

DDEコールのメカニズム

DDEメカニズムは「サービス」に基づいています。DDEメカニズムを実行するには、「コマンド」を実行するために必要なコンテキストを提供する「トピック」を定義する必要があります。整合性を維持するため、コンテキストを変更するたびに、前のコンテキストを終了する必要があります。

この章では、次の内容について説明します。

- DDEサービス
- DDEトピック

- DDEコマンド

DDEサービス

ほとんどの場合、「サービス」はメモリに読み込まれる実行可能プログラムの名前です。Asset ManagerをDDEサーバとして使う場合、サービスは「**am**」になります。

DDEトピック

トピックでは、アクションを実行するコンテキストを定義できます。Asset Managerの場合、このトピックは「**Asset Manager**」です。

DDEコマンド

DDEコマンドは、Asset Managerに送信される実行コマンドです。これらはいくつかのグループに分けられます。

- グローバルコマンド。実行にテーブル名やフィールド名は必要ありません。
- テーブルに関連するコマンド。実行するには、パラメータとしてテーブルのSQL名が必要です。
- テーブルおよびフィールドまたはリンクに関連するコマンド。実行するには、パラメータとしてテーブルのSQL名とフィールドまたはリンクのSQL名が必要です。
- ヘルプデスクの応答に関連するコマンド。

これらのグループに属するコマンドは、次の2つのタイプに分けることができます。

- **Excute**（実行）：Asset Managerでタスクを実行できます。
- **Request**（要求）：Asset Managerに情報の提供を要求できます。

テーブル、フィールド、またはリンクのSQL名の見つけ方

テーブルのフィールドをマウスの右ボタンでクリックすると、ショートカットメニューが表示されます。【**オブジェクトの設定**】メニューを選択すると、ウィンドウに選択しているテーブルやオブジェクト（リンクまたはフィールド）のSQL名が表示されます。

DDEコマンドの概要

手順

DDEコマンドを正しく実行するには、次の手順に従う必要があります。

- 1 使用する「サービス」と「トピック」を指定して、コマンド実行のコンテキストを明確に定義します。Asset ManagerをDDEサーバとして使う場合、「トピック」は常に「Asset Manager」です。

 **注意:**

一旦コンテキストを定義すると、新しいコンテキストを定義するまで、このコンテキストが後続のすべてのDDEコマンドでデフォルトとして使われます。

- 2 コマンド自体を起動します。2種類のコマンドがあります。
 - **Execute:** <コマンド><パラメータ>
 - **Request:** <コマンド><パラメータ>
- 3 前に定義したコンテキストを閉じます。

特徴

Windowsの各アプリケーションは、DDEコマンドを独自の方法で送受信します。後半では、次の内容について説明します。

- Asset Managerが受信できるDDEコマンドの一覧。これらのコマンドのシNTAXについて詳しく説明します。
- DDEメカニズムによるAsset Managerの駆動例。これらのコマンドの使い方の説明します。これらの例では、異なるプログラミング言語を使います。

グローバルコマンド

「グローバル」コマンドは、特定のテーブルやフィールドに依存しません。特に、引数としてテーブルやフィールドのSQL名を指定する必要はありません。この章では次のコマンドについて説明します。

- Connect(Cnx, User, Password)
- Disconnect()
- ExecuteAction(ActionName)
- ListAllTables([Mask])
- ListOpenTables([Mask])
- OpenView(ViewName)

Connect(<Cnx>, <User>, <Password>)

アクションのタイプ

Execute

詳細

次のパラメータを使って、データベースに接続します。

<Cnx>

このパラメータには、次のどちらかを含めることができます。

- **Asset Manager**で定義された接続名（「Amdb.ini」ファイルで検索できます）
このファイルの場所：▶ 『**Asset Manager - インストールとアップグレード**』ガイドの「**.iniおよび.cfgファイル**」の章。
- 次に示すシンタックスで定義するデータベース接続

```
[<データベースエンジン>;<データベースの場所>;<データベースエンジンの  
ログイン>;<データベースエンジンのパスワード>]
```

<User>

このパラメータには、データベースへの接続に使う**Asset Manager**ユーザ名が入ります。

<Password>

このパラメータには、ログイン（<User>パラメータの値）に関連付けられたパスワードが入ります。

例

次のコマンドにより、**Asset Manager**で接続が既に定義されている**Oracle**データベースに接続できます。接続名は「Tdemo」です。接続には「Admin」ログインが使われます。パスワードは「password」です。

```
Execute:Connect(TDemo, Admin, password)
```

次のコマンドは、**Asset Manager**で接続を定義しなくても、上のコマンドと同じ接続を実行します。「Tdemo」データベースは「Joshua」というサーバにあります。**Oracle**接続のパスワードは「Root」です。

```
Execute:Connect([Oracle;Joshua;TDemo;Root], Admin, password)
```

Disconnect()

アクションのタイプ

Execute

詳細

Asset Managerと現在のデータベースの接続を終了します。

例

次のコマンドは、Asset Managerデータベースとの接続を終了します。

```
Execute:Disconnect()
```

ExecuteAction(<ActionName>)

アクションのタイプ

Execute

詳細

<ActionName>というアクションをトリガします。

<ActionName>

このパラメータには、Asset Managerのアクションの詳細画面の【名前】（SQL名：Name）フィールドで定義したアクション名が入ります。

例

次のコマンドは、「Reminder: work order not completed」というアクションをトリガします。

```
Execute:ExecuteAction(Reminder: work order not completed)
```

ListAllTables([Mask])

アクションのタイプ

依頼

詳細

データベース内のすべてのテーブルを一覧表示します。テーブルのSQL名を<Mask>で検索し、リストに表示します。

<Mask>

この引数では、次の文字を使ってデータを検索できます。

- 1文字を検索するワイルドカードとして疑問符 (?) を使用できます。
- 任意の文字または文字のグループを検索するときは、アスタリスク (*) を使用できます。

例

次のコマンドは、現在のデータベースに存在するすべてのテーブルのSQL名を一覧表示します。

```
Request:ListAllTables()
```

次のコマンドは、現在のデータベースからSQL名が「amA」で始まるすべてのテーブルのSQL名を一覧表示します。

```
Request:ListAllTables(amA*)
```

次のコマンドは、現在のデータベースから「v」という文字を含むすべてのテーブルのSQL名を一覧表示します。

```
Request:ListAllTables(*v*)
```

次のコマンドは、現在のデータベースから「am」で始まり、4番目の文字に「t」を持つすべてのテーブルのSQL名を一覧表示します。

```
Request:ListAllTables(am?t*)
```

ListOpenTables([Mask])

アクションのタイプ

依頼

詳細

データベースで開いているすべてのテーブルのSQL名を一覧表示します。このリストは、<Mask>パラメータを使って検索できます。

<Mask>

この引数では、次の文字を使ってデータを検索できます。

- 1文字を検索するワイルドカードとして疑問符 (?) を使用できます。
- 任意の文字または文字のグループを検索するときは、アスタリスク (*) を使用できます。

例

次のコマンドは、現在のデータベースで開いているすべてのテーブルのSQL名を一覧表示します。

```
Request:ListOpenTables()
```

次のコマンドは、現在のデータベースで開いているすべてのテーブルから、SQL名が「amA」で始まるテーブルのSQL名を一覧表示します。

```
Request:ListOpenTables(amA*)
```

「amAsset」、「amAction」、「amModel」という3つのテーブルを開いているとします。上のコマンドを使うと、「amA」で始まる2つのテーブルのSQL名、つまり「amAsset」と「amAction」が返されます。

OpenView(<ViewName>)

アクションのタイプ

Execute

詳細

Asset Managerで定義されたビューを開きます。

<ViewName>

このパラメータには、Asset Managerで定義されているビューのSQL名が入ります。

例

次のコマンドは、**[CurrentAssets]** というSQL名のビューを開きます。

```
Execute:OpenView(CurrentAssets)
```

テーブルに関連するコマンド

ここでは、テーブルに依存するコマンドについて説明します。これらのコマンドを実行するには、引数としてテーブルのSQL名を指定する必要があります。

この章では次のコマンドについて説明します。

- OpenTable(Table)
- CloseTable(Table)
- Table.GetRecordCount()
- Table.SetViewMode(Mode)

- Table.SetRecordMode(Mode)
- Table.ListAllFields([Mask])
- Table.ListAllLinks([Mask])
- Table.SetFilter(Condition)
- Table.SetSelection(Condition)
- Table.GetSelectionId()

OpenTable(<Table>)

アクションのタイプ

Execute

詳細

SQL名が<Table>のテーブルを開きます。

<Table>

この引数には、開くテーブルのSQL名が入ります。

例

次のコマンドは、[資産] (SQL名 : amAsset) テーブルを開きます。

```
Execute:OpenTable(amAsset)
```

CloseTable(<Table>)

アクションのタイプ

Execute

詳細

現在Asset Managerで開いているテーブルを閉じます。

<Table>

この引数には、閉じるテーブルのSQL名が入ります。

例

次のコマンドは、[資産] (SQL名 : amAsset) テーブルを閉じます。

```
Execute:CloseTable(amAsset)
```


<Table>.GetRecordCount()

アクションのタイプ

依頼

詳細

<Table>というSQL名を持つテーブルのレコード数を返します。このコマンドを実行するには、このコマンドに関連するテーブルを開いておく必要があります。

<Table>

この引数には、レコード数を確認するテーブルのSQL名が入ります。

例

次のコマンドは、[資産]（SQL名：amAsset）テーブルのレコード数を返します。

```
Request:amAsset.GetRecordCount()
```

<Table>.SetViewMode(<Mode>)

アクションのタイプ

Execute

詳細

開いているテーブルの表示モードを定義します。

<Table>

この引数には、対象のテーブルのSQL名が入ります。

<Mode>

この引数には、次のいずれかの値を設定できます。

- **Arbo**：テーブル<Table>内のレコードがツリー構造で表示されます。
- **List**：テーブル<Table>内のレコードがリスト形式で表示されます。
- **ListOnly**：テーブル<Table>内のレコードのリストだけが表示されます。
- **DetailOnly**：テーブル<Table>内で選択したレコードの詳細だけが表示されます。
- **ListDetail**：テーブル<Table>内のレコードのリストとこのリストで選択したレコードの詳細の両方が表示されます。

例

次のコマンドは、[ポートフォリオ品目]（SQL名：amPortfolio）テーブルをツリー構造で表示します。

```
Execute:amPortfolio.SetViewMode(Arbo)
```

<Table>.SetRecordMode(<Mode>)

アクションのタイプ

Execute

詳細

開いているテーブルのレコードとの対話モードを定義します。

<Table>

この引数には、対象のテーブルのSQL名が入ります。

<Mode>

この引数には、次のいずれかの値を設定できます。

- **New**：テーブル<Table>で新規レコードの作成を開始します。このコマンドは[新規作成] ボタンをクリックする操作に対応します。
- **"Duplicate"**：<Table>テーブルで選択したレコードを複製します。このコマンドは[複製] ボタンをクリックする操作に対応します。
- **"Delete"**：<Table>テーブルで選択したレコードを削除します。このコマンドは[削除] ボタンをクリックする操作に対応します。
- **Modify**：テーブル<Table>で選択したレコードに加えた変更を確定します。このコマンドは[変更] ボタンをクリックする操作に対応します。
- **Create**：テーブル<Table>での新規レコードの作成を確定します。このコマンドは[作成] ボタンをクリックする操作に対応します。
- **CreateContinue**：作成と複製を組み合わせます。このコマンドは**CreateContinue** ボタンをクリックする操作に対応します。
- **Cancel**：新規レコードの作成または選択したレコードに加えた変更を取り消します。このコマンドは[キャンセル] ボタンをクリックする操作に対応します。
- **Close**：前に開いたテーブル<Table>を閉じます。このコマンドは[閉じる] ボタンをクリックする操作に対応します。

例

次の例では、[資産]（SQL名：amAsset）テーブルを開き、新規レコードの作成を開始し、この作成を取り消します。

```
Execute:OpenTable(amAsset) Execute:amAsset.SetRecordMode(New) Execut  
e:amAsset.SetRecordMode(Cancel)
```

<Table>.ListAllFields([Mask])

アクションのタイプ

依頼

詳細

前に開いたテーブル<Table>に存在するすべてのフィールドのSQL名を返します。

<Table>

この引数には、対象のテーブルのSQL名が入ります。

<Mask>

この引数では、次の文字を使ってデータを検索できます。

- 1文字を検索するワイルドカードとして疑問符 (?) を使用できます。
- 任意の文字または文字のグループを検索するときは、アスタリスク (*) を使用できます。

例

次のコマンドは、資産のテーブルに存在するすべてのフィールドのSQL名を返します。

```
Request:amAsset.ListAllFields
```

次のコマンドは、資産のテーブル (SQL名 : amAsset) から、SQL名が「se」で始まるすべてのフィールドのSQL名の一覧を返します。

```
Request:amAsset.ListAllFields(se*)
```

<Table>.ListAllLinks([Mask])

アクションのタイプ

依頼

詳細

前に開いたテーブル<Table>に存在するすべてのリンクのSQL名を返します。

<Table>

この引数には、対象のテーブルのSQL名が入ります。

<Mask>

この引数では、次の文字を使ってデータを検索できます。

- 1文字を検索するワイルドカードとして疑問符 (?) を使用できます。
- 任意の文字または文字のグループを検索するときは、アスタリスク (*) を使用できます。

例

次のコマンドは、[資産] (SQL名 : amAsset) テーブルに存在するすべてのリンクのSQL名を返します。

```
Request:amAsset.ListAllLinks
```

次のコマンドは、[資産] (SQL名 : amAsset) テーブルから、SQL名が「se」で始まるすべてのリンクのSQL名の一覧を返します。

```
Request:amAsset.ListAllLinks(se*)
```

<Table>.SetFilter(<Condition>)

アクションのタイプ

Execute

詳細

<Condition>引数に従って、テーブル<Table>にフィルタを適用します

<Table>

この引数には、対象のテーブルのSQL名が入ります。

<Condition>

この引数には、コマンドに適用する条件が入ります。これはAQL節です。

例

次のコマンドは、[資産] (SQL名 : amAsset) テーブルにフィルタを適用します。このフィルタにより、2002年8月28日15時より前に変更されたレコードだけが表示されます。

```
Execute:amAsset.SetFilter(dtLastModif<[02/08/28 15:00:00])
```

<Table>.SetSelection(<Condition>)

アクションのタイプ

Execute

詳細

前に開いたテーブル<Table>から、引数<Condition>に従って1つまたは複数のレコードを選択します。

<Table>

この引数には、対象のテーブルのSQL名が入ります。

<Condition>

この引数には、コマンドに適用する条件が入ります。これはAQL節です。

例

次のコマンドは、資産タグが「7」以上の資産を選択します。

```
Execute:amAsset.SetSelection(AssetTag>='7')
```

<Table>.GetSelectionId()

アクションのタイプ

依頼

詳細

テーブル<Table>で選択したレコードの識別子の一覧を返します。

<Table>

この引数には、対象のテーブルのSQL名が入ります。

例

次の例では、[資産] (SQL名: amAsset) テーブル内で、資産タグが「7」以上のレコードを選択し、選択したレコードの識別子の一覧を返します。

```
Execute:amAsset.SetSelection(AssetTag>='7') Request:amAsset.GetSelectionId()
```

テーブルおよびフィルタまたはリンクと関連するコマンド

ここでは、テーブル内のフィールドに依存するコマンドについて説明します。これらのコマンドを実行するには、引数としてテーブルのSQL名とこのテーブルのフィールドまたはリンクのSQL名を指定する必要があります。

この章では次のコマンドについて説明します。


- Table-Object.AddLink()
- Table-Object.GetValue()
- Table-Object.Highlight()
- Table-Object.RemoveLink()
- Table-Object.SetFocus()
- Table-Object.SetValue(Value)
- Table-Link.SetValueWhere(Condition)
- Table-Object.Show()

<Table>:<Object>.AddLink()

アクションのタイプ

Execute

詳細

リストの  ボタンをクリックするのと同じ操作を実行します。このコマンドを使ってテーブルのレコードにリンクを追加することができます。

<Table>

この引数には、操作の対象となるテーブルのSQL名が入ります。

<Object>

この引数には、操作の対象となるオブジェクトのSQL名が入ります。

例

次のコマンドは、リストデータに値を追加します。

```
Execute:amItemizedList:ItemListVals.AddLink()
```

<Table>:<Object>.GetValue()

アクションのタイプ

依頼

詳細

選択したレコードについて、テーブル<Table>の<Object>（フィールドまたはリンク）の値を返します。

<Table>

この引数には、対象のテーブルのSQL名が入ります。

<Object>

この引数には、テーブル<Table>の値を取得するフィールドまたはリンクのSQL名が入ります。

例

次のコマンドは、[資産]（SQL名：amAsset）テーブルの [フィールド1]（SQL名：Field1）フィールドの値を返します。

```
Request:amAsset:Field1.GetValue()
```

次のコマンドは、[資産]（SQL名：amAsset）テーブルの [モデル]（SQL名：Model）リンクの値を返します。

```
Request:amAsset:Model.GetValue()
```

次のコマンドは、[資産]（SQL名：amAsset）テーブルの [コメント]（SQL名：Comment）リンクの値を返します。

```
Request:amAsset:Comment.GetValue()
```

<Table>:<Object>.Hilight()

アクションのタイプ

Execute

詳細

ある1つのフィールドを強調表示します。

<Table>

この引数には、操作の対象となるテーブルのSQL名が入ります。

<Object>

この引数には、操作の対象となるフィールドのSQL名が入ります。このコマンドはリンクには使用できません。

例

次のコマンドは、[資産] (SQL名: amAsset) テーブルの [バーコード] (SQL名: Barcode) フィールドを強調表示します。


```
Execute:amAsset:Barcode.Hilight()
```

<Table>:<Object>.RemoveLink()

アクションのタイプ

Execute

詳細

リストの  ボタンをクリックするのと同じ操作を実行します。このコマンドを使ってテーブルのレコードへのリンクを削除することができます。

<Table>

この引数には、操作の対象となるテーブルのSQL名が入ります。

<Object>

この引数には、操作の対象となるオブジェクトのSQL名が入ります。

例

次のコマンドは、リストデータ内で選択した値を削除します。

```
Execute:amItemizedList:ItemListVals.RemoveLink()
```

<Table>:<Object>.SetFocus()

アクションのタイプ

Execute

詳細

選択したレコードについて、テーブル<Table>のフィールドまたはリンク<Object>にフォーカスを設定します。

<Table>

この引数には、フォーカスを設定するフィールドまたはリンクを含むテーブルのSQL名が入ります。

<Object>

この引数には、テーブル<Table>のフォーカスを設定するフィールドまたはリンクのSQL名が入ります。

例

次のコマンドは、[資産] テーブル（[amAsset]）の[資産タグ] フィールド（[AssetTag]）にフォーカスを設定します。

```
Execute:amAsset:AssetTag.SetFocus()
```

<Table>:<Object>.SetValue(<Value>)

アクションのタイプ

Execute

詳細

選択したレコードについて、テーブル<Table>のフィールド<Field>に値<Value>を入力します。

<Table>

この引数には、値を入力するフィールドを含むテーブルのSQL名が入ります。

<Field>

この引数には、テーブル<Table>内の値を入力するフィールドのSQL名が入ります。

<Value>

この引数には、選択したレコードについて、テーブル「<Table>」の「<Field>」フィールドに割り当てる値が入ります。「日付」または「日付+時刻」型のフィールドの場合は、国際日付形式（yy/mm/dd hh:mm:ss）で指定する必要があります。

例

次のコマンドは、[資産]（SQL名：amAsset）テーブルの[フィールド1]（SQL名：Field1）フィールドに「Test」という値を割り当てます。

```
Execute:amAsset:Field1.SetValue(Test)
```

次のコマンドは、選択したレコードについて、資産テーブル（SQL名：amAsset）の【受領日】（SQL名：dAccept）フィールドに「02/08/28」という値を割り当てます。

```
Execute:amAsset:dAccept.SetValue(02/08/28)
```

<Table>:<Link>.SetValueWhere(<Condition>)

アクションのタイプ

Execute

詳細

選択したレコードについて、条件<Condition>に従って、テーブル<Table>のリンク<Link>に値を入力します。

<Table>

この引数には、値を入力するフィールドを含むテーブルのSQL名が入ります。

<Link>

この引数には、テーブル<Table>の値を入力するリンクのSQL名が入ります。

<Condition>

この引数には、リンクのターゲットレコードを識別するための条件が入ります。これはAQL節です。

例

次の例は、選択したレコードについて、【資産】（SQL名：amAsset）テーブルの【モデル】（SQL名：Model）リンクに値「Test」を割り当てます。DDEコマンドを正しく実行するには、「Test」というカテゴリがなければなりません。

```
Execute:amAsset:Model.SetValueWhere(Name='Test')
```

<Table>:<Object>.Show()

アクションのタイプ

Execute

詳細

画面に表示されないフィールドまたはリンクにフォーカスを移動します。フォーカスするフィールドまたはリンクのテーブルを開いておく必要があります。

<Table>

この引数には、操作の対象となるテーブルのSQL名が入ります。

<Object>

この引数には、操作の対象となるオブジェクトのSQL名が入ります。

例

次のコマンドは、[資産] (SQL名: amAsset) テーブルの [会計コード] (SQL名: AcctCode) フィールドにフォーカスを移動します。

```
Execute:amAsset:AcctCode.Show()
```

ヘルプデスクの応答に関連するコマンド

これらのコマンドは、DDEメカニズムを介したヘルプデスクの応答に使用するよう設計されています。

この章では次のコマンドについて説明します。

- NewTicket()
- NewTicket.SetTicketMode(Mode)
- NewTicket.GetTicketNumber()
- NewTicket-Item.SetValue(Value)
- NewTicket-Item.SetValueWhere(Condition)
- NewTicket-Item.GetValue ()

NewTicket()

アクションのタイプ

Execute

詳細

ヘルプデスクの応答ツールをトリガします。

例

次のコマンドは、応答ツールをトリガし、新しいヘルプデスクチケットに入力できるようにします。

```
Execute:NewTicket()
```

NewTicket.SetTicketMode(<Mode>)

アクションのタイプ

Execute

詳細

応答画面のボタンの1つをクリックした状況をシミュレートします。

<Mode>

この引数には、次のいずれかの値を設定できます。

- **Save** : 現在のコールを保存します。このモードは、**【コールを保存】** ボタンのクリックに対応します。
- **Solved** : 現在のコールを終了します。このモードは、**【チケットをクローズ】** ボタンのクリックに対応します。
- **Proceed** : チケットを処理します。このモードは、**【処理】** ボタンのクリックに対応します。
- **SaveNOpen** : コールを保存し、チケットを開きます。このモードは、**【詳細を入力】** ボタンのクリックに対応します。
- **Cancel** : コールを取り消します。このモードは、**【キャンセル】** ボタンのクリックに対応します。
- **Close** : 応答画面を閉じます。このモードは、**【閉じる】** ボタンのクリックに対応します。
- **ForceNewFile** : 応答画面の**【チケット】** リンクに入力したチケットのサブチケットの作成をトリガします。

例

次のコマンドは、応答ツールをトリガし、取り消します。

```
Execute:NewTicket() Execute:NewTicket.SetTicketMode(Cancel)
```

NewTicket.GetTicketNumber()

アクションのタイプ

依頼

詳細

応答画面から現在のチケット番号を返します。

例

次のコマンドは、応答ツールをトリガし、作成しているチケットの番号を返します。

```
Execute:NewTicket() Execute:NewTicket.GetTicketNumber()
```

NewTicket:<Object>.SetValue(<Value>)

アクションのタイプ

Execute

詳細

応答画面のフィールド (<Object>) の1つに<Value>という値を入力します。

<Object>

この引数には、次のいずれかの値を設定できます。

- **Description** : 応答画面の **【説明】** フィールドに値を入力します
- **DescComp** : 応答画面の **【追加の説明】** フィールドに値を入力します。
- **Notes** : 応答画面の **【メモ】** フィールドに値を入力します。

<Value>

この引数には、応答画面の<Object>フィールドに割り当てる値が入ります。

例

次のコマンドは、応答ツールをトリガして、**【説明】** および **【メモ】** フィールドに値を入力します。

```
Execute:NewTicket() Execute:NewTicket:Description.SetValue(Second problem on this hardware) Execute:NewTicket:Notes.SetValue(To fix urgently)
```

NewTicket:<Object>.SetValueWhere(<Condition>)

アクションのタイプ

Execute

詳細

条件<Condition>に従って、応答画面のいずれかのリンク (<Object>) に値を入力します。

<Object>

この引数には、次のいずれかの値を設定できます。

- Caller：応答画面の【依頼主】リンクに入力します。
- File：応答画面の【チケット】リンクに入力します。
- Asset：応答画面の【資産】リンクに入力します。
- ProblemClass：応答画面の【タイプ】リンクに入力します。
- EscalScheme：応答画面の【エスカレーション】リンクに入力します。
- Contact：応答画面の【連絡先】リンクに入力します。

<Condition>

この引数には、コマンドの条件を入力します。これはAQL節です。

例

次のコマンドは、応答ツールをトリガし、【依頼主】および【タイプ】リンクに入力します。「本多和義」という従業員と「プリンタの問題」という問題のタイプのレコードがデータベース内になければなりません。

```
Execute:NewTicket() Execute:NewTicket:Caller.SetValueWhere(Name='本多和義') Execute:NewTicket:ProblemClass.SetValueWhere(Name='プリンタの問題')
```

NewTicket:<Object>.GetValue ()

アクションのタイプ

依頼

詳細

応答ツールのフィールドまたはリンク (<Object>) のいずれかの値を返します。

<Object>

この引数には、次のいずれかの値を設定できます。

- Caller：応答画面の【依頼主】リンクの値を返します。
- File：応答画面の【チケット】リンクの値を返します。
- Asset：応答画面の【資産】リンクの値を返します。
- ProblemClass：応答画面の【タイプ】リンクの値を返します。
- Description：応答画面の【説明】フィールドの値を返します。
- DescComp：応答画面の【追加の説明】フィールドの値を返します。
- EscalScheme：応答画面の【エスカレーション】リンクの値を返します。
- Contact：応答画面の【連絡先】リンクの値を返します。
- Notes：応答画面の【メモ】フィールドの値を返します。

例

次のコマンドは、応答ツールの【依頼主】フィールドの値を返します。

```
Execute:NewTicket:Caller.GetValue
```

DDEコール例の概要

ここでは、次の例を使ってDDEコールの機能を説明します。

- 例1では、DDEコールを論理的に説明します。
- 例2では、ExcelでDDEコールを使う例を紹介합니다。使用する例は、Visual Basic for Applicationsで作成しています。
- 例3では、Visual Basic 5.0で作成したアプリケーションを使います。この例では、DDEコールを使って実際に操作を実行することができます。

例1 : Asset Managerの内部DDEコール

このシナリオの目的は、実行する各アクションについて、適切なDDEコマンドと引数を指定することです。まず、DDEメカニズムを論理的に理解することが必要です。次に後述の例で実際のDDEコールを試してください。

この例では、San Diegoに位置する「Headquarters」という場所を作成します。実行するアクションは次のとおりです。

- 1 [場所] テーブルを開く。
- 2 場所の名前を入力する。
- 3 場所の市町村名を入力する。
- 4 場所の郵便番号を入力する。
- 5 場所を作成する。

はじめに

まず、DDEコマンドを実行するために必要なサービスとトピックを指定する必要があります。

この例ではAsset Managerアプリケーションの一般的なコンテキストを使うので、次のようになります。

- サービス = **am**
- トピック : **Asset Manager**

次に、[場所] テーブルを開くコマンドを実行します。

- コマンド : **OpenTable()**

- パラメータ：テーブルのSQL名。この場合は「**amLocation**」
コマンドは次のようになります。

```
OpenTable(amLocation)
```

Asset Managerは [場所] テーブルを開きます。次に、このテーブルに新規レコードを作成する必要があります。

- コマンド：**SetRecordMode()**
- コマンドの接頭語：このテーブルのSQL名 (**amLocation**)
- パラメータ：データ入力モード。ここでは「**New**」
コマンドは次のようになります。

```
amLocation.SetRecordMode(New)
```

データを入力する

次に、Asset Managerで必要なフィールドに入力するための適切なコマンドを指定する必要があります。

- [名前] (SQL名：Name) フィールド。次のコマンドと引数を使います。
 - コマンド：<テーブル>:<オブジェクト>.Set<値>(<値>)
 - * 引数<テーブル>：テーブルのSQL名 (**amLocation**)
 - * 引数<オブジェクト>：フィールドのSQL名 (**Name**)
 - * 引数<値>：フィールドの値 (**Headquarters**)

```
amLocation:TextLabel.SetValue(Headquarters)
```

- [市町村] (SQL名：City) フィールド。次のコマンドと引数を使います。
 - コマンド：<テーブル>:<オブジェクト>.Set<値>(<値>)
 - * 引数<テーブル>：テーブルのSQL名 (**amLocation**)
 - * 引数<オブジェクト>：フィールドのSQL名 (**City**)
 - * 引数<値>：フィールドの値 (**San Diego**)

```
amLocation:seDataType.SetValue(San Diego)
```

- [郵便番号] (SQL名：ZIP) フィールド。次のコマンドと引数を使います。
 - コマンド：<テーブル>:<オブジェクト>.Set<値>(<値>)
 - * 引数<テーブル>：テーブルのSQL名 (**amLocation**)
 - * 引数<オブジェクト>：フィールドのSQL名 (**ZIP**)
 - * 引数<値>：フィールドの値 (**CA 92130**)

```
amLocation:Unit.SetValue(CA 92130)
```


場所を作成する

[場所] テーブルにレコードを作成するだけで、場所を作成できます。

- コマンド : <テーブル>.SetRecordMode(<モード>)
 - 引数<テーブル> : テーブルのSQL名 (**amLocation**)
 - 引数<モード> : レコードの作成モード (**Create**)

```
amLocation.SetRecordMode(Create)
```

例3 : Visual BasicからのDDEコール

このシナリオでは、単純なグラフィカルユーザインタフェースからDDEメカニズムを試すことができるユーティリティを使います。この例から、Visual BasicでDDEメカニズムをプログラムする方法もわかります。

この例では、**DDE TestCenter.exe**を実行します。(Samples\DDE\Program) このプログラムにより、**Execute**と**Request**タイプのDDEコマンドを実行できます。

注意:

Basicプログラムから発行されたDDEコマンドを受け取るには、Asset Managerを起動する必要があります。

この章では、次の内容について説明します。

- プログラムのソースコード
- 注意事項
- プログラムを実行する

プログラムのソースコード

このプログラムのVisual Basic プロジェクト形式のコメント付きソースコードは、**sample\DDE\VisualBasic**にあります。

注意事項

インストール

このプログラムを使うには、コンピュータにVisual Basicを正しくインストールする必要があります。特に、特定のActiveXコントロールを正しく登録する必要があります。DDE TestCenterから「コントロールXXXXが登録されていません」などのエラーが返された場合は、次の操作を行ってください。

- 1 コンピュータでコントロールを検索し、そのフォルダに移動する。

- 2 次のコマンドを実行する。

```
regsvr32 XXXX
```

- 3 DDE TestCenterを再起動する。それでも起動できない場合は、Visual Basicのドキュメントを参照してください。

推奨事項

この例を実行し易くするためには、次の操作を行ってください。

- 1 Asset Managerを起動し、アプリケーションウィンドウのサイズを画面の半分程度に縮小する。
- 2 DDE TestCenter.exeを起動し、アプリケーションウィンドウをAsset Managerウィンドウの横に移動する。



注意:

これにより、Asset ManagerのDDE TestCenter.exeから発行された命令の結果を、直接見られるようになります。

シンタックス

この例ではDDEコールがVisual Basic標準に対応しています。それ以外は前の例と同じです。


プログラムを実行する

「Execute」（実行）タイプのDDEコマンド

[Execute] タブに移動します。

[コマンド] フィールドに実行するコマンドを入力します。次のシンタックスを使います。

```
Command=<コマンド><引数>
```

 ボタンをクリックしてコマンドを実行します。エラーは、[Last DDE Error] (最新DDEエラー) フィールドに表示されます。

例1

次のExecuteコマンドは、[任意管理項目] テーブルを開きます。

```
OpenTable(amLocation)
```

例2

次の**Execute**コマンドは、[予算] のテーブルを開き、新しいレコードを作成して、詳細画面の【名前】(SQL名: Name) フィールドに入力します。このコマンドは順番に実行されます。


```
OpenTable(amBudget) amBudget.SetRecordMode(New) amBudget.Name.SetValue("Test")
```

「Request」(要求)タイプのDDEコマンド

【Request】タブに移動します。

【コマンド】フィールドに実行するコマンドを入力します。次のシンタックスを使います。

```
Command=<コマンド><引数>
```

 ボタンをクリックしてコマンドを実行します。要求の結果は、【Request Result】(要求の結果)フィールドに表示されます。エラーは、【Last DDE Error】(最新DDEエラー)フィールドに表示されます。

例1

次の**Request**コマンドは、現在の接続しているデータベースのすべてのテーブルのSQL名を一覧表示します。

```
ListAllTables()
```

例2

次の**Request**コマンドは、前に開いた【任意管理項目】(amFeature) テーブルに存在するすべてのフィールドのSQL名の一覧を表示します。

```
amFeature.ListAllFields()
```


17 WANネットワークにおける Asset Managerの最適化

WANネットワークには次のような特徴があります。

- 低帯域幅
- 長い待ち時間

これらの欠点を最小限に抑えるようにAsset Managerを設定することができます。しかし、これらの設定を行うと、Asset Managerの特定の機能が利用できなくなります。

この章では、WANネットワークの欠点を調整するためのいくつかのヒントを紹介します。応答時間の高速化と機能の損失の妥協点をテストすることが重要です。

【編集／オプション】メニューのオプション

次のオプションを使って、データベースへのアクセスの長さを制限できます。

- 【ナビゲーション】項目の【半自動入力タイミング】オプション：このオプションをオフにするか、高い値（10000ミリ秒後に入力開始など）を指定します。
- 【ナビゲーション】項目の【ドロップダウンリストをツリー構造で表示】オプション：リスト形式表示のパフォーマンスに比べると、ツリー表示はパフォーマンスを低下させるため、このオプションをオフにします。
ただし、この場合は、ドロップダウンリストのツリー表示機能も使えなくなります。

次のオプションを調整して、クライアントマシンとデータベースサーバ間での情報交換を制限できます。

- **【リスト】** 項目の **【読み込み時間の限度】** および **【読み込む項目数の限度】** オプション（メインリストまたはタブページ内のリスト用）：読み込む項目数を制限するようにしてください（例えば、メインリストの最大値を50項目、タブページのリストの最大値を15項目に指定します）。読み込む項目数は、リストにフィルタを適用する場合や、指定した項目数で必要な情報を見つけることができるどうかを基準にして設定します。
- **【定期チェック】** 項目の **【メッセージ】**：このオプションを使って、着信メッセージのチェックをデータベースへの接続時のみに行ったり、チェックする間隔を設定したり（10分置きなど）することができます。
- **【キャッシュ】** 項目：キャッシュを更新する間隔（**【間隔】** 列）を長くしたり、セッション中のキャッシュの更新を行わないように設定できます。この場合は、キャッシュはデータベースへの接続時のみ読み込まれます。
キャッシュを定期的に更新しないと、最新のデータが表示されないことがあります。ただし、キャッシュされるデータ項目の大多数（リストデータ、任意管理項目のディクショナリ、業務用カレンダーなど）は、Asset Managerのインストール時に生成され、定期的に変更されることはありません。

リスト

リストの設定

メインリストとタブページのリストは、次の場合に **【リストの設定】** メニューを使って設定できます。

- メニューを使って表示したテーブルのリスト（**【資産ライフサイクル／インフラストラクチャ管理／資産】** ナビゲーションメニューなど）
- ビューによって表示されるリスト（**【ツール／ビュー】** メニュー）
- 選択リスト（**【リンクの選択】** ショートカットメニュー）
- 詳細画面の特定のタブページに表示されるリスト

リストを並べ替える

これらのリストは次の方法で並べ替えできます。

- 独自の並べ替え条件を選択（**【並べ替え】** 列）
- 定義済みのインデックスを使用（**【インデックスで並べ替え】** フィールド）

これら2つのオプションを選択した場合のそれぞれのパフォーマンスは、その時々で異なります。どちらのオプションが最適かを予測することはできません。

データベースに最適な方法を決定する前に、Asset Managerの各リストについて両方の方法をテストする必要があります。

フィルタ

リストにフィルタを適用することができます。

次の条件に比例してリストの表示時間が増加します。

- フィルタの条件数
- フィルタ条件の適用対象のテーブル間の距離
- フィルタクエリで用いる「**OR**」の数

表示する列を選択する

次の条件に比例してリストの表示時間が増加します。

- 表示する列数
- 表示するフィールド／リンクが属するテーブル間の距離

リスト形式またはツリー構造で表示する

ツリー構造で表示すると、リスト形式で表示する場合よりも時間がかかります。

リストでアイコンを表示する

アイコンは、テキストよりも表示に時間がかかります。

データベースレベルでリストのパラメータを設定する

ここに挙げるインタフェースオプションは、リストの表示時間に影響を及ぼします。

このようなオプションを表示するには：

- 1 [編集／オプション] メニューを選択します。
- 2 必要に応じて以下のオプションを変更します。
 - [リスト／その他のリスト／読み込み時間の限度]
 - [リスト／その他のリスト／読み込む項目数の限度]
 - [リスト／メインリスト／読み込み時間の限度]
 - [リスト／メインリスト／読み込む項目数の限度]

リストに読み込む項目数を削減すると、リストの表示時間が短縮されます。

リストの表示にかかる最長時間を短縮すると、特定の期間に表示される項目数が削減されます。

注意:

上記のオプションはデータベースに保存され、データベースの全ユーザに適用されます。

インタフェースのオプション設定に関する詳細は、『**Tailoring**』ガイド、「**Customizing client computers**」、「**Customizing a client workstation**」の章、「**General options**」のセクションを参照してください。

Asset Managerクライアントレベルでリストのパラメータを設定する

「**FetchingArraySize**」パラメータが、Asset Managerのデータ接続に Asset ManagerのサポートするすべてのDBMSにより使用されます。

DBMSは、Asset Managerが要求するレコードをグループ単位で送信します。このグループのサイズ（レコード数）は、「**FetchingArraySize**」パラメータにより定義されます。

各Asset Managerクライアントで、Asset Manager接続ごとに「**amdb.ini**」ファイルでこのパラメータを指定します。

このファイルの場所：▶『**Asset Manager - インストールとアップグレード**』ガイドの「**.iniおよび.cfgファイル**」の章。

「**amdb.ini**」ファイルに「**FetchingArraySize**」パラメータが含まれていない場合、このパラメータのデフォルト値は「**30**」です。

このパラメータは、以下のパラメータ（**[編集/オプション]**メニューと相互に作用します。

- **[リスト/その他のリスト/読み込む項目数の限度]**
- **[リスト/メインリスト/読み込む項目数の限度]**

WAN用に最適化されていないオプションの例

- **[読み込む項目数の限度]**を「**200**」に設定したとします。
- 「**FetchingArraySize**」が「**30**」であるとします。
- この場合、Asset Managerは7回に分けてリストにレコードを読み込むため（ $200 / 30 = 6,7$ ）、一回で表示するよりも時間がかかります。

ヒント:

ここでかかる時間はLANでは通常問題になりませんが、WANでは問題になる可能性があります。

WAN用に最適化されたオプションの例

表示時間が長すぎる場合は、表示項目が一度に取得されるように設定を変更します。

次の規則を適用します。

$\text{FetchingArraySize} = \text{読み込む項目数の限度} + 1$
--

注意:

WANで250 msの待機時間を使ってこのシナリオを試行したところ、このパラメータを最適化することで、レコード200個の表示時間が**1.5秒短縮**されました。

応用例:

- 「読み込む項目数の限度」を「**200**」に設定します。
- 「**FetchingArraySize**」を「**210**」に設定します。
- この結果、1回でリストにレコードが読み込まれます。

「amdb.ini」ファイルを変更する

- 1 「amdb.ini」ファイルを編集します。
このファイルの場所: ▶ 『**Asset Manager - インストールとアップグレード**』ガイドの「**.iniおよび.cfgファイル**」の章。
- 2 [<最適化するAsset Managerの接続名>]セクションを選択します。
- 3 同セクション内に、「**FetchingArraySize=**」から始まる項目が既に存在するかどうかを確認します。
存在する場合は、既存のパラメータを変更します。
- 4 セクション内に「**FetchingArraySize=**」から始まる項目が存在しない場合は、セクションに次の1行を追加します: **FetchingArraySize=<パラメータ値>**

ヒント:

上記の操作は、各クライアントコンピュータで実行します。

表示を制限する

本当に必要な列、リスト、およびタブだけを表示することで、画面に表示するデータ項目数を制限し、アプリケーションの応答時間を改善することができます。

接続キャッシュ

接続の管理の詳細画面の【キャッシュ】タブページから接続キャッシュを起動することができます。

接続キャッシュを起動すると、次のことが可能になります。

- データベースへの接続時間を短縮する。

- 画像やアイコンを使用している場合に、時間を節約する。
一般に、デフォルトのキャッシュサイズで十分最適化されます。

アクセス制限

使用しているログインにアクセス制限がある場合は、詳細およびリストウィンドウの表示に時間がかかります。これは、データを表示する前にテストが実行されるためです。

確認する場合は、制限のないログインでリストまたは詳細を表示し、表示速度を比較します。

必要に応じてオプションのアクセス制限を削除します。

1つのクライアントの設定を他のクライアントに適用する

1台のワークステーションのパフォーマンスを最適化したら、設定の変更を他のクライアントワークステーションにも適用する必要があります。

加えた変更内容に対応する「amdb.ini」ファイルをコピーすると、各マシンで設定する手間が省けます。

このファイルの場所：▶ 『**Asset Manager - インストールとアップグレード**』ガイドの「**.iniおよび.cfgファイル**」の章。

索引

- 「edac-mobiledevices.scn」 (シナリオ) , 259
- [Enterprise Discovery] 棚卸結果を使用して、データベースにあるすべてのモバイル機器を更新 (モジュール) , 258
- [アクション] テーブル
 - メッセージシステム, 275
 - 自動トリガ (参考 Asset Manager Automated Process Manager)
- この接続をサービスモードで使用 (オプション) , 234 , 233
- アイコン, 327
- アクセス (参考 アクセスコントロール)
- アクセスコントロール (参考 認証)
 - ユーザ, 175
 - ユーザ接続, 177
 - 概要, 163
 - 利点, 163
 - 制限, 163
 - 編集, 166
- アクセス管理専門分野データをインポートする, 166
- アクセス制限, 171
 - リンク - デフォルト値, 173 , 173
 - 低速ネットワーク - チューニング, 330
- 定義, 166
- アクセス権の定義
 - DDEコマンドサーバ, 297
 - アクセス制限, 166
 - ユーザプロファイル, 165
 - ユーザ役割, 165
 - ユーザ権限, 165
 - 機能権限, 166
- アクティブ化
 - データベース, 17
- アラームの検証 (モジュール) , 253
- インデックス
 - SQL名, 42
 - ラベル, 42
 - 数値, 48
 - 詳細, 42 , 42
- インポート, 129-161
 - ODBC - データベースの選択, 136
 - インポートするフィールド
 - マッピング, 143 , 141
 - 説明, 139
 - キー, 144
 - スクリプト, 160 , 158
 - ソーステーブル - マッピング, 142
 - テキストファイル

- マッピング, 142
- 構造, 135
- テキストファイル - 選択, 136
- テーブル - 分析の説明, 136
- データベース, 130
- データベース構造とデータ, 73
- ドキュメント, 151
- パフォーマンス, 156
- ファイル, 136
- フォルダ - 作成, 157
- プロセス, 134
- リンク - 作成パラメータ, 144
- 部署と従業員, 151
- 概要, 130
- 数値データ, 131
- 特殊フィールド, 147
- 推奨事項, 131
- 文字 - ISOコード, 130
- 制限, 132
- 単一のテキストファイル, 130
- 複数のテキストファイル, 130
- 転送 - 設定, 156
- インポートスクリプト
 - コマンドラインからの実行, 160
 - 保存と実行, 158
- エラーメッセージ
 - データベースのアクティブ化, 18
- オーバーフロー, 39
 - テーブル, 39
 - リンクの動作, 40
- オーバーフローテーブル, 39
 - 実装, 41
- オーバーフローリンク, 39
- カスタムリストデータ (入力タイプ), 33, 32, 32
- キャッシュ, 329
- キャッシュ (Oracle), 50
- キー
 - インポート, 144
 - 例, 151
- コストセンタの経費を分割する (モジュール), 251
- コメント, 82
- サーバと比較したローカルマシン時間の検証 (オプション), 237

- システムリストデータ (入力タイプ), 33, 32, 32
- シングルサインオン (参考 認証)
- Webクライアント
 - Active Directoryの使用, 197
 - IIS, 198
 - Jakarta ISAPI Redirector, 198
 - SiteMinder, 217
 - ポート転送, 200
 - 前提条件, 198
- Windowsクライアント
 - Active Directoryの使用, 193
 - 概要, 194
 - 導入, 194
 - 従業員, 196
- シングルサインオン (SSO)
 - IIS 32ビット, 201
 - IIS 64ビット, 208
- Webクライアント
 - IIS7 64ビット用のポート転送の設定, 216
 - Jakarta ISAPI Redirector, 202
 - Jakarta ISAPI Redirector : 64ビット設定, 209
 - トラブルシューティング, 216, 207
 - ポート転送の設定, 206
- スクリプト
 - オーバーフローリンク, 40
- スケジュールのグラフィック表示, 281-296
 - オプション, 281
 - スケジュールのグラフィック表示ページウィザードに追加, 287
 - テーブルに追加, 282
 - 実用例, 288, 283
 - 設定, 289
 - 概要, 281
 - 休日, 281
- スケジュールデータのグラフィック表現 (参考 スケジュールのグラフィック表示)
- スロットの自動タイムアウト (オプション), 185
- セキュリティ, 164
 - (参考 アクセスコントロール)
- セキュリティ問題
 - (参考 認証)

タイムゾーン, 61
 タイムゾーンの使用 (オプション), 61
 ツリー構造 - インポート, 149
 テキスト (データ型), 31
 テキストファイル - インポート, 135
 テンプレート (メニュー), 78
 テーブル
 SQL名, 28
 オーバーフローテーブル, 39
 ラベル, 28
 参照テーブル, 39
 数値, 48
 中間テーブル, 38
 記述文字列, 29
 説明, 28, 28
 テーブルの統計情報の更新 (モジュール), 250
 テーブルまたはフィールド名 (入力タイプ), 32, 32
 テーブルスペース (Oracle), 50
 デフォルト (入力タイプ), 31, 31
 データ (Microsoft SQL Server設定), 50
 データの機密性, 164
 データベース
 (参考 ODBC)
 (参考 アクセスコントロール)
 (参考 インデックス)
 (参考 インポート)
 (参考 テーブル)
 (参考 フィールド)
 (参考 リンク)
 (参考 接続)
 Microsoft SQL Server, 50
 UDB DB2 - クライアント, 54
 アクセスタイプ, 177
 不特定, 177
 固定, 177
 一時, 177
 アクティブ化, 17
 アクティブ化のエラー, 18
 データ
 インポート, 73
 出力, 73
 ライセンスキー, 13
 定義, 25
 削除, 74
 修理
 修復, 109-110
 作成, 45-74
 DB2 UDB, 52
 DBMS, 47
 空のデータベース, 46
 前提条件, 45
 事前検証, 50
 新規データベース, 22
 構造
 Asset Manager Application Designer, 57
 インポート, 73
 表示, 26
 出力, 73
 更新, 62
 記述 - テンプレート, 81-89
 else
 if
 グローバル変数
 コメント
 コンポーネントの参照
 プロパティ - 値
 参照したコンポーネント - 数
 抽出
 改行マーク - 削除
 固定テキスト
 並べ替え
 別のテンプレートを挿入する, 82
 関数
 記述 - ファイル, 77-108
 グローバル変数
 パラメータ
 概要
 診断, 109
 詳細
 説明, 77
 開く, 23
 データベース (オプション), 236
 データベースにNTドメインのコンピュータを追加 (モジュール), 240, 240
 データベースにNTユーザを追加 (モジュール), 242, 242
 データベースに言語を挿入 (メニュー), 64

- データベースの診断/修復 (メニュー) , 109
- データベースサーバ, 47
- データベースサーバに信号送信 (モジュール) , 251
- データベースサーバのタイムゾーンの検証 (オプション) , 237
- データベースサーバのタイムゾーンの検証 (モジュール) , 253
- データベース管理者, 176
- データベース構造とデータのインポート (メニュー) , 73
- データベース構造とデータの出力 (メニュー) , 73
- データベース記述ファイルを開く - 新規データベースの作成 (オプション) , 22
- ドキュメント
 - Asset Manager Application Designer - メモリ中の最大数, 22
 - Asset Manager Application Designer - 自動読み込み, 20
 - インポート, 151
- ドロップダウンリストをツリー構造で表示 (オプション) , 325
- パスワード
 - Adminログイン, 186
 - パスワードを忘れた場合, 190
 - 変更, 189
- パスワード (フィールド) , 197
- パフォーマンス (参考 低速ネットワーク - チューニング)
- パーセント (入力タイプ) , 32 , 32
- ファイル (オプション) , 236
- フィルタ
 - 低速ネットワーク - チューニング, 327
- フィールド
 - SQL名, 30
 - デフォルト値
 - アクセス制限, 173 , 173
 - データ型と形式, 30
 - ラベル, 30
 - 値の一意性 - インデックス, 42
 - 説明, 30 , 29
- フォルダの選択 (メニュー) , 78
- ブール (はい/いいえ) (入力タイプ) , 32
- ブール (入力タイプ) , 32

- ヘルプデスクアラームのトリガ (モジュール) , 247
- マイグレーション, 62
- メッセージ (オプション) , 236
- メッセージ (メニュー) , 276
- メッセージシステム
 - HTML, 275
 - MAPI, 271
 - mail.ini, 271
 - MAPI - 従業員の設定, 269
 - SMTP, 269
 - mail.ini, 269
 - 従業員 - 設定, 267
 - 設定, 267
- VIM, 273
 - mail.ini, 273
 - 従業員 - 設定, 271
 - 設定, 273
- インタフェース, 265-279
- テスト, 277
- メッセージの参照, 276
- メッセージの送信, 275
- 概要, 265
- 接続 - 問題, 277
 - エラー, 277
 - 問題が発生した場合に実行するテスト, 277
- 配信済通知, 276
- 受信, 275
- 送信, 275
- 複数の受信者, 267
- 複数プロトコルの使用, 266
- 設定, 266
- メモフィールド (データ型) , 31
- モジュール
 - ライセンス権限, 13
- ユーザプロファイル, 168
 - 定義, 165
- ユーザ役割, 168
 - ユーザの接続に使用する役割を選択する, 176
 - 定義, 165
- ユーザ権限, 168
 - 定義, 165
- ライセンスキー, 13

- ライセンスキーファイル
 - データベースのアクティブ化のエラー, 18
 - 変更, 17
- ライセンス権限, 17
- ラベル
 - テーブル, 28
 - フィールド, 30
- リストの更新 (メニュー), 78
- リストの設定 (メニュー), 326
- リンク
 - Cardinality, 36
 - SQL名, 35
 - インポート, 144
 - タイプ, 36
 - ラベル, 35
 - 状況依存リンク, 37
 - 説明, 35, 35
- リンクタイプ
 - 外部キー, 40
- 半自動入力のタイミング (オプション), 325
- 保存される言語, 68
- 倍精度 (データ型), 31
- 表示言語, 63
 - オブジェクトタイプ, 66
 - 置換, 65
 - 追加, 64
 - 選択, 65
- 不特定 - アクセス, 177
- 部署 - インポート, 151
- 抽出
 - データベース記述, 82
- 出カイベントテーブルの消去 (モジュール), 250
- 次へ (メニュー), 24
- 低速ネットワーク - チューニング, 325-330
 - アクセス制限, 330
 - オプション, 325
 - キャッシュ, 329
 - リスト, 326
 - 画面, 329
- 定期チェック (オプション), 326
- 定義
 - データベース, 25
- 浮動小数点数 (データ型), 31
- 更新日 (フィールド), 33
- 固定 - アクセス, 177
- 固定テキスト, 81
- 固有のインデックス
 - 削除または変更に関する注意, 25
- 管理者権限 (オプション), 176
- 画面, 329
- 画像 (リンクが記憶する情報のタイプ), 36
- 既存のデータベースを開く (オプション), 23
- 接続
 - (参考 LDAP)
 - (参考 接続スロット)
 - (参考 指定)
 - Windows XP, 56
 - 管理, 177
 - 指定, 56
- 接続スロット
 - 非アクティブユーザ, 184
 - 概要, 182
 - 更新 - 間隔, 183
 - 破壊, 183
 - 手動破棄, 183
 - 異常終了, 184
- 金額 (入力タイプ), 32, 32
- 可変長バイナリフィールド (データ型), 31
- 列
 - インポート - 区切り文字, 139
 - 低速ネットワーク - チューニング, 327
- 履歴項目の検証 (モジュール), 256
- 匿名 (アクセス), 191
- 期限 - モニタ (参考 Asset Manager Automated Process Manager)
- 期間 (入力タイプ), 33, 32, 32
- 前へ (メニュー), 24
- 任意管理項目の値 (入力タイプ), 32, 32
- 日付+時刻 (データ型), 31
- 日付+時刻 (入力タイプ), 32, 32
- 日付 (データ型), 31
- 日付 (入力タイプ), 32, 32
- 入カイベントテーブルの消去 (モジュール), 250
- 入力タイプ (フィールド), 31
- 受領品に対応する資産、消耗品などの作成 (モジュール), 247
- 数値 (入力タイプ), 32, 32

特殊フィールド
 インポート, 143
外部キー, 40
完全名 (フィールド), 34, 29
文字セット (パラメータ), 48
文字列 - 区切り文字, 139
新しいワークフロー実行グループの検索 (モジュール), 251
言語
 Asset Manager Application Designer, 20
 一時 - アクセス, 177
在庫の検証 (モジュール), 256
整数 (16ビット) (データ型), 31
整数 (32ビット) (データ型), 31
整数 (8ビット) (データ型), 31
自動切断を有効にする (オプション), 184
最大サイズ (オプション), 237
並べ替え
 データベース - 記述
 抽出, 82
 低速ネットワーク - チューニング, 326
 階層リスト, 29
実行グループにワークフロー規則を適用 (モジュール), 248
実用例
 スケジュールのグラフィック表示, 288, 283
帯域幅 (参考 低速ネットワーク - チューニング)
従業員 - インポート, 151
時刻 (データ型), 31
時刻 (入力タイプ), 32, 32
構造とデータの出力, 73
機能権限, 173
 定義, 166
規定の損失額の計算 (モジュール), 246
記述文字列, 29
認証, 186
 (参考 LDAP)
 Asset Managerのみ, 188
読み込み時間の限度 (オプション), 326
読み込む項目数の限度 (オプション), 326
賃貸料とローンの計算 (モジュール), 244
連結, 148
進行ステータス (パラメータ), 295

開いているオブジェクトの数 (パラメータ), 48, 48

A

Active Directory

 シングルサインオン, 193
 シングルサインオン (SSO) に関する設定, 201
adbc.dll, 111
addecpu.scn (シナリオ), 241
AddLink (DDE), 310
adduser.scn (シナリオ), 243
Admin (ログイン), 186, 19
alias (キーワード), 83
amdb.ini, 330, 328
amimpl.exe, 160
amsrvvcf.ini, 261
AM (メッセージシステム), 274
API, 25
APP_CTL_HEAP_SZ (UDB DB2), 52
AppBuild (プロパティ), 90
AppInfo (プロパティ), 90
AppLanguage (プロパティ), 90
APPLHEAPSZ (UDB DB2), 52
AppVersion (プロパティ), 90
Asset Manager Application Designer (参考 データベース)
 はじめに, 19
 インタフェース, 19-24, 20
 オブジェクトタイプ, 21
 オプション, 24
 テーブル, 21
 ドキュメント - 自動読み込み, 20
 言語, 20
 注意事項, 19
 検索, 24
Asset Manager Automated Process Manager, 231-264
 Web - 制限, 264
 アイコン, 235
 インストール, 238
 オプション, 236
 サービス - としての実行, 262
 サービスとして開始, 260

- データベース
 - 接続, 234
 - 接続解除, 234
- メイン画面, 235
- メッセージシステム, 234
- メニュー, 235
- モジュール - 設定, 238
- モニタ - タイムゾーン, 238
- 概要, 231
- 管理 - Web, 260
 - Asset Manager Automated Process Managerサービス, 262
 - 起動, 261, 261
 - 制御, 263
- 即時トリガ, 260
- 起動
 - DOS, 233
 - 手動起動, 233
 - 自動起動, 233
- 制限, 232
- 自動切断, 264
- 実行
 - Windows, 234
 - 推奨する画像の詳細, 232
- 終了 - 結果, 234
- Asset Manager Automated Process Managerで受領品をポートフォリオに自動的に作成 (オプション), 247
- Asset Managerデータベース (ODBC接続), 113
- Asset Managerユーザ役割を選択 (ウィザード), 176
- auto create statistics (Microsoft SQL Server), 50, 50
- AvailableScript (オブジェクト), 102

B

- BaseType (プロパティ), 106
- Base (オブジェクト), 99, 97, 94, 92
- BASICスクリプト (入力タイプ), 32, 32
- Basicスクリプト - インポート, 150

C

- CalcFieldDesc (インスタンス), 103

- CalcField (オブジェクト), 107
- CalcMode (プロパティ), 99
- CaptionDetail (プロパティ), 105
- CaptionList (プロパティ), 105
- Cardinality (プロパティ), 96
- Citrix XenApp
 - Asset Managerにアクセスするための使用
- Class (オブジェクト), 101
- CloseTable (DDE), 304
- CollectionCount (関数), 87
- CollectionCreate (関数), 85
- CollectionCurrentIndex (関数), 86
- CollectionIsFirst (関数), 86
- CollectionIsLast (関数), 86
- CollectionName (関数), 86
- CollectionNext (関数), 86
- Collection (インスタンス), 80
- Comment (リンクが記憶する情報のタイプ), 36, 36
- CompleteDates (パラメータ), 293
- ComputeString (プロパティ), 90
- config (フォルダ), 22
- Connect (DDE), 300
- Connected (プロパティ), 90
- Connection (プロパティ), 90
- Copy (リンクタイプ), 36
- Count (プロパティ), 91
- Create (プロパティ), 90
- CreationHistorized (プロパティ), 102
- CurrentIndex (プロパティ), 91

D

- database.txt, 77, 30, 26
 - 構造, 27
- Database Owner (Microsoft SQL Server), 51
- Database (インスタンス), 90, 80
- Date (Asset Managerデータベース記述インスタンスのグローバル変数), 108
- db_block_buffers (パラメータ), 49
- db_block_size (パラメータ), 48
- DB2 UDB, 52, 52
- dbase.tpl, 27

Dbb.Ext (Asset Managerデータベース記述
 インスタンスのグローバル変数) , 108
 Dbb.Fullname (Asset Managerデータベ
 ース記述インスタンスのグローバル変数) , 108
 Dbb.Name (Asset Managerデータベース記
 述インスタンスのグローバル変数) , 108
 Dbb.Path (Asset Managerデータベース記述
 インスタンスのグローバル変数) , 108
 Dbb.Shortname (Asset Managerデータベ
 ース記述インスタンスのグローバル変数) , 108
 DbbVersion (プロパティ) , 90
 dbdict.tpl, 78 , 27
 DBMS (参考 データベース)
 変更, 72
 DBMSログイン, 55
 DDE, 297-323
 グローバルコマンド, 299
 Connect, 300
 Disconnect, 301
 ExecuteAction, 301
 ListAllTables, 301
 ListOpenTables, 302
 OpenView, 303
 コマンド, 298
 はじめに, 299
 コンテキスト, 299
 コール, 297
 サービス, 298
 テーブルおよびフィルタまたはリンクと関
 連するコマンド, 310
 AddLink, 310
 GetTicketNumber, 316
 GetValue, 318 , 311
 Hilight, 311
 NewTicket, 315
 RemoveLink, 312
 SetFocus, 312
 SetTicketMode, 316
 SetValue, 317 , 313
 SetValueWhere, 317 , 314
 Show, 314
 テーブルに関連するコマンド, 303
 CloseTable, 304
 GetRecordCount, 305
 GetSelectionId, 309
 ListAllFields, 307
 ListAllLinks, 307
 OpenTable, 304
 SetFilter, 308
 SetRecordMode, 306
 SetSelection, 309
 SetViewMode, 305
 トピック, 298
 ヘルプデスクの応答に関連するコマンド,
 315
 呼び出し - 例
 Visual Basic, 321
 はじめに, 319
 内部呼び出し, 319
 前提条件, 321
 DDE TestCenter.exe, 321
 DDEコマンド (参考 DDE)
 DDEコマンドサーバ, 297
 DDEサービス, 298
 DDEトピック, 298
 DefaultScript (オブジェクト) , 102 , 94
 Define (リンクタイプ) , 36
 Delete (プロパティ) , 90
 Desc (プロパティ) , 105 , 104 , 104 , 100 , 98
 , 96 , 93 , 90
 Disconnect (DDE), 301
 dml_locks (パラメータ) , 49
 DstField (オブジェクト) , 97
 DstTable (オブジェクト) , 97
 dtLastModif (フィールド) , 33
 インポート, 134
 Duplicates (プロパティ) , 98

E

edac.scn (シナリオ) , 258
 elseif (キーワード) , 84
 else (キーワード) , 84
 EmptyOnDup (プロパティ) , 94
 EndAfter (パラメータ) , 292
 EndBefore (パラメータ) , 291
 EndDate (パラメータ) , 290
 endfor (キーワード) , 82
 endif (キーワード) , 84
 endscrip (関数) , 87

Enterprise Discovery 棚卸結果を使用してデータベースを更新 (モジュール), 256

Enum (プロパティ), 101

ExecuteAction (DDE), 301

Exist (関数), 85

F

FeatClassDesc (インスタンス), 103

FeatDesc (インスタンス), 100

FeatParamDesc (インスタンス), 101

FeatParam (オブジェクト), 107

Feature value, 36

Feature values (リンクが記憶する情報のタイプ), 36

FeatureValueTable (オブジェクト), 91

Feature (オブジェクト), 103

FetchingArraySize (パラメータ), 328

FieldCase (プロパティ), 94

FieldCount (プロパティ), 91

Field (インスタンス), 92, 80

Field (オブジェクト), 106, 100

filter (キーワード), 83

ForceDisplayScript (オブジェクト), 102

ForeignKey (プロパティ), 93

Formula (プロパティ), 104

for (キーワード), 82

FrameLabel (プロパティ), 106

FullName (フィールド), 34

注意事項, 34

FullName (プロパティ), 103

FuncDomain (プロパティ), 104

function (関数), 87

FVSourceTable (オブジェクト), 91

G

gbbase.xml, 22

gbbase*.*, 77, 26

GenerateColors (パラメータ), 292

GetRecordCount (DDE), 305

GetSelectionId (DDE), 309

GetTicketNumber (DDE), 316

GetValue (DDE), 318, 311

H

HasClass (プロパティ), 101

HasFeatureValueTable (プロパティ), 91

HasMemoField (プロパティ), 91

HasNotebook (プロパティ), 105

HasParent (プロパティ), 103

HasSystemPage (プロパティ), 105

Hilight (DDE), 311

Historized (プロパティ), 96, 93

HistoryScript (オブジェクト), 102, 97, 94

History (リンクが記憶する情報のタイプ), 36

HP AutoPass

ライセンスキー, 13

I

if (キーワード), 84

IIS

SSO用の設定, 211, 202

IIS 32ビット

シングルサインオン (SSO) に関する設定, 201

IIS 64ビット

シングルサインオン (SSO) の設定, 208

IIS7

Tomcatコネクタ (Jakarta ISAPI) : 64

ビット設定のインストール, 209

Tomcatコネクタ (Jakarta ISAPI) のインストール, 202

IIS-Tomcat Redirector, 209, 202, 198

IndexCount (プロパティ), 91

Index (インスタンス), 98, 80

InternalName (プロパティ), 105, 104, 98, 96, 93, 90

IsCalcField (プロパティ), 106

IsConsolidated (プロパティ), 101

IsFeatureValueTable (プロパティ), 91

IsFeature (プロパティ), 106

IsField (プロパティ), 106

IsFirst (プロパティ), 91

IsHorizontalSplit (プロパティ), 106

IsInherited (プロパティ), 102

IsLast (プロパティ), 91

IsLink (プロパティ) , 106
IsValidScript (オブジェクト) , 91
IsVerticalSplit (プロパティ) , 106
IsVisible (プロパティ) , 104
ItemColor (パラメータ) , 292
ItemLabel (パラメータ) , 294

J

Jakarta ISAPI Redirector, 202 , 198
Jakarta ISAPI Redirector : 64ビット設定,
209
Jakarta Tomcatコネクタ, 209 , 202

L

Label (プロパティ) , 105 , 104 , 104 , 100 ,
98 , 96 , 93 , 90

LDAP

- アクティブ化, 192
- パスワード, 191
- リマインダ, 191
- 接続, 192
- 導入, 191
- 認証, 191

LDAPディレクトリ (参考 LDAP)

LDAP認証 (参考 LDAP)

Lightweightシングルサインオン (参考

LW-SSO)

LineLabel (パラメータ) , 294

LinkCount (プロパティ) , 91

LinkFilter (プロパティ) , 102

Link (インスタンス) , 96 , 80

Link (オブジェクト) , 107

ListAllFields (DDE), 307

ListAllLinks (DDE), 307

ListAllTables (DDE), 301

ListOpenTables (DDE), 302

log_buffer (パラメータ) , 49

LOG (Microsoft SQL Server), 50

LogError (関数) , 85

LoginId (プロパティ) , 90

LoginName (プロパティ) , 90

LongHelpCommentNoHTMLTag (プロパ
ティ) , 101 , 97 , 93

LongHelpComment (プロパティ) , 101 , 97
 , 93

LongHelpDescNoHTMLTag (プロパティ) ,
102 , 97 , 93

LongHelpDesc (プロパティ) , 101 , 97 , 93

LongHelpSampleNoHTMLTag (プロパ
ティ) , 102 , 97 , 93

LongHelpSample (プロパティ) , 101 , 97 ,
93

LongHelpWarningNoHTMLTag (プロパ
ティ) , 102 , 97 , 93

LongHelpWarning (プロパティ) , 101 , 97
 , 93

LW-SSO

initstringパラメータ, 220

SAML2、問題解決, 228

システム要件, 226

セキュリティの考慮事項, 220

トラブルシューティング, 226

概要, 218

既知の問題, 225

使用例, 218

制限事項, 222

主な機能, 218

注意, 224

仮設定, 219

M

mail.ini, 273 , 271 , 269

MainField (パラメータ) , 289

MainIndex (オブジェクト) , 91

MandatoryScript (オブジェクト) , 102 , 94

MandatoryType (property), 134

MAPI (参考 メッセージシステム)

max server memory (Microsoft SQL Server),
50

MaxValue (プロパティ) , 101

Microsoft SQL Server

SQL 2000データベースのSQL 2005 Server
へのリンク, 74

クライアント, 51

サーバ, 50

データベース - 作成, 50

MinValue (プロパティ) , 101

N

Name (プロパティ), 107, 103
Neutral (リンクタイプ), 36
NewTicket (DDE), 315
Normal (リンクが記憶する情報のタイプ), 36
Normal (リンクタイプ), 36
NTサービス, 260
NullValues (プロパティ), 98
null識別子レコードの検証 (モジュール), 255
NumValue (プロパティ), 107

O

Object (インスタンス), 80
ODBC, 111-113
 DBMS, 111
 SAP Crystal Reports - 作成, 113
 ドライバ
 インストール, 111
 利点, 112
 表示されたデータ, 112
 概要, 111
 接続 - 作成, 113
 使用する接続, 113
 制限, 111
ODBCアクセス (参考 ODBC)
ODBCドライバ (参考 ODBC)
open_cursors (パラメータ), 49
OpenTable (DDE), 304
OpenView (DDE), 303
optimizer_mode (パラメータ), 49
Oracle, 48
OrderBy (パラメータ), 295
Owncopy (リンクタイプ), 36
Own (リンクタイプ), 36

P

PageItem (インスタンス), 106
Page (インスタンス), 105
ParentClass (オブジェクト), 103
PlannerViewer (パラメータ), 289
Positive (プロパティ), 94
PrimaryKey (オブジェクト), 91

PrimaryKey (プロパティ), 93
processes (パラメータ), 49
Property (インスタンス), 80

R

RAID, 47
RawSource (プロパティ), 100
ReadOnlyScript (オブジェクト), 94
ReadOnly (プロパティ), 93
 制限, 134
Read (プロパティ), 102, 96, 92
ReferencedStorageFieldCount (プロパティ), 100
RelDstField (オブジェクト), 97
RelevantScript (オブジェクト), 97, 94, 91
RelSrcField (オブジェクト), 97
RelTable (オブジェクト), 97
RemoveLink (DDE), 312
ReverseLink (オブジェクト), 97
rollback segments (Oracle), 50

S

SAP Crystal Reports, 113 (参考 SAP Crystal Reports)
ScreenFilter (プロパティ), 105
ScriptField (インスタンス), 107
ScriptType (プロパティ), 99
Script (Asset Managerデータベース記述インスタンスのオブジェクト), 104
Script (インスタンス), 99, 80
script (関数), 87
SetFilter (DDE), 308
SetFocus (DDE), 312
SetOutput (関数), 85
SetProperty (関数), 84
SetRecordMode (DDE), 306
SetSelection (DDE), 309
SetTicketMode (DDE), 316
SetValue (DDE), 317, 313
SetValueWhere (DDE), 317, 314
SetViewMode (DDE), 305
shared_pool_size (パラメータ), 49
Show (DDE), 314

Size (プロパティ), 92
制限, 134
SMTP (参考 メッセージシステム)
sort_area_retained_size (パラメータ), 49
sort_area_size (パラメータ), 49
sort (キーワード), 83
Source (プロパティ), 100
SQLName (プロパティ), 106, 105, 104, 104, 100
SqlName (プロパティ), 98, 96, 93, 90
SQL名
インデックス, 42
テーブル, 28
フィールド, 30
リンク, 35
用途, 28
SrcField (オブジェクト), 97
SrcTable (オブジェクト), 97
SSO (参考 認証)
StartAfter (パラメータ), 291
StartBefore (パラメータ), 290
StartDate (パラメータ), 290
StringValue (プロパティ), 107
String (インスタンス), 107
SysEnumValue (インスタンス), 107
SysPage (オブジェクト), 105
system data source (UDB DB2), 54
System (プロパティ), 105, 104, 100, 99, 96, 93, 91

T

TableCount (プロパティ), 90
tables.txt, 77, 26
構造, 27
Table (Asset Managerデータベース記述インスタンスのオブジェクト), 104
Table (インスタンス), 90, 80
Table (オブジェクト), 105, 102, 100, 99, 94
tempdb (Microsoft SQL Server), 51
Template.Ext (Asset Managerデータベース記述インスタンスのグローバル変数), 108

Template.Fullname (Asset Managerデータベース記述インスタンスのグローバル変数), 108
Template.Name (Asset Managerデータベース記述インスタンスのグローバル変数), 108
Template.Path (Asset Managerデータベース記述インスタンスのグローバル変数), 108
Template.Shortname (Asset Managerデータベース記述インスタンスのグローバル変数), 108
Time (Asset Managerデータベース記述インスタンスのグローバル変数), 108
tpl (ファイル), 81, 77, 26
Truncate Log on Checkpoint (Microsoft SQL Server), 51
Typed (プロパティ), 96
TypeField (オブジェクト), 97
Type (インスタンス), 97
Type (プロパティ), 104, 101, 96, 94, 92
データをインポートする際を守るべき制限, 134

U

UDB DB2
クライアント, 54
Unicode, 68
Oracleデータベースのサポートの追加, 70
Unit (プロパティ), 101
Update (プロパティ), 102, 96, 92
user data source (UDB DB2), 54
Userlogin (Asset Managerデータベース記述インスタンスのグローバル変数), 108
UserTypeFormat (プロパティ), 95, 92
UserType (インスタンス), 98
UserType (プロパティ), 104, 96, 95, 92
インポート - 制限, 132

V

ValueField (オブジェクト), 103
ValueOf (関数), 84
Value (プロパティ), 107
VbReturnType (プロパティ), 100
View (インスタンス), 104

VIM (参考 メッセージシステム)

W

WAN (参考 低速ネットワーク - チューニング)

WANネットワーク (参考 低速ネットワーク - チューニング)

WhereCond (パラメータ) , 295

Windows 64ビット対応のJava

IIS 7を使用したインストール, 209

Write (プロパティ) , 102 , 96 , 92

X

XenApp

Asset Managerにアクセスするための使用,
115

