

Peregrine

AssetCenter 4.3

管理

© Copyright 2004 Peregrine Systems, Inc.
All rights reserved.

本書に記載されている情報は、Peregrine Systems, Incorporatedが所有し、Peregrine Systems, Inc.の書面による許可なく使用または開示することはできません。本書の一部または全部を、Peregrine Systems, Inc.の事前の書面による許可なく無断で複製することを禁じます。本書に記載されている商品名は、該当する各社の商標または登録商標です。

Peregrine Systems ® および AssetCenter ® は、Peregrine Systems, Inc.の商標です。

本書で説明されているソフトウェアは、Peregrine Systems, Inc.とエンドユーザ間で締結されるライセンス契約に基づいて提供されます。契約の条項に従って、ソフトウェアを使用する必要があります。Peregrine Systems, Inc.は、本書の内容については一切の責任を負いかねます。また、本書の内容が予告なく変更されることもあります。本書の最終バージョンの日付を確認するには、Peregrine Systems, Inc.のカスタマサポートまでお問合せください。

デモ用データベースと本書の例に使用されている団体名および個人名は架空のものであり、本ソフトウェアの使用方法を説明するためのものです。現在、過去を問わず、実在する団体や個人とのいかなる類似もまったくの偶然によるものです。

本製品に関する技術情報の請求、またはライセンスをお持ちの製品に関するマニュアル類の請求については、Peregrine Systemsのカスタマサポート (support@peregrine.com) までお寄せください。

本マニュアルに関するご意見やご要望は、Peregrine Systems, Inc.の出版部 (doc_comments@peregrine.com) までお寄せください。

本書の内容は、ライセンス契約に基づくプログラムのバージョン4.3に適用されます。

AssetCenter

目次

はじめに（管理）	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. AssetCenter Database Administrator - インタフェース	13
はじめに	13
概要	14
2. データベースの標準記述ファイル	21
はじめに	21
データベースの定義	22
Database.txtとTables.txtの構造	23
テーブルについて	24
フィールドについて	25
リンクについて	32
インデックスについて	40
データベースのカスタマイズ	41
3. AssetCenterデータベースの作成	47
DBMSを使って空のAssetCenterデータベースを作成する	48

空のシェルへの接続を指定する	61
管理者パスワードを入力してデータベース構造を作成する	61
AssetCenter Database Administratorを使ってデータベースを作成する	61
ライセンスファイルを選択する	65
ライセンスを変更する	66
AssetCenterクライアント言語	67
DBMSを変更する	70
4. データベース記述ファイルの作成	73
はじめに	74
データベース記述パラメータ	75
記述テンプレートのシンタックス	78
特定のデータベース記述パラメータについて	89
5. データベースの診断と修復	113
6. データベース構造の更新	117
7. ODBCドライバを使ったデータベースへのアクセス	119
AssetCenterデータベースへのアクセス	119
例：ODBCドライバでCrystal Reportsのレポートを作成する	122
8. データベースのカスタマイズ	123
既存のオブジェクトをカスタマイズする	123
新規オブジェクトの作成	130
テスト用データベースで用意されたカスタマイズ	140
複数のオペレータによって同時に行なわれたカスタマイズ	145
9. アーカイブ	171
アーカイブを有効にする	171
アーカイブの原則	173
アーカイブの無効化	178
10. データベースオプション	179
データベースオプションを設定する	179
データベースオプションの設定ウィンドウ	180
11. インポート	189
データのインポートの概要	190
推奨事項	191

フィールドにデータをインポートする際の制約事項	192
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	195
インポートスクリプトを保存および実行する	222
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	224
12. データベースへのアクセスコントロール	227
アクセス権限の管理の重要性和概要	228
データのセキュリティと機密性を確実に保護する方法	229
アクセス権限の定義	229
アクセス条件を定義する	231
AssetCenterのユーザを定義する	236
データベースへの接続数を管理する	238
パスワードを管理する	241
13. AssetCenter Server	247
AssetCenter Serverの概要	247
AssetCenter Serverを実行する	248
AssetCenter Serverのメイン画面	256
AssetCenter Serverの全般オプション	257
AssetCenter Serverでモニタするモジュールを設定する	259
手動で検証モジュールを起動する	281
AssetCenter ServerをWebブラウザで管理する	282
14. メッセージシステム	287
メッセージの送受信	287
AssetCenterでメッセージシステムを指定する	288
一般的な接続エラー	298
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	301
概要	301
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	301
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	302
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	307
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	309
16. AssetCenterをDDEサーバとして使用する	319
DDEサーバの定義	319
DDEコールのメカニズム	319

DDEコマンドの概要	321
グローバルコマンド	321
テーブルに関連するコマンド	326
テーブルおよびフィルタまたはリンクに関連するコマンド	333
DDEコール例の概要	339
例1: AssetCenterの内部DDEコール	340
例2: Microsoft ExcelからのDDEコール	342
例3: Visual BasicからのDDEコール	343
17. WANネットワークにおけるAssetCenterの最適化	347
[編集／オプション] メニューのオプション	347
リスト	348
表示を制限する	352
接続キャッシュ	352
アクセス制限	352
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	352
18. AutoCADの統合	355
AutoCAD／AssetCenter統合の概要	355
インストール	356
設定	356
AutoCAD統合を使用する	357
インデックス	361

図の一覧表

1.1. データベースを開く画面	16
2.1. 中間テーブルの役割	35
2.2. 【資産】テーブルと【固定資産】テーブルとのリンク	35
2.3. 【履歴】テーブルとターゲットテーブルとのリンク	39
3.1. データベースの作成画面	62
8.1. AssetCenter Database Administrator - メイン画面	124
8.2. 複数のオペレータによるカスタマイズ - 単純なプロセス	148
11.1. インポートするデータのタイプを選択する	189
12.1. ユーザプロファイルの詳細画面	228
12.2. ユーザプロファイル - アクセス制限	235
14.1. 内部メッセージシステムの仕組み	288
15.1. スケジュールのグラフィック表示 - 作業指示での例	306

表の一覧表

2.1. フィールド - タイプ	26
2.2. データの入力形式とタイプ	28
2.3. [入力タイプ] のフィールド値	30
2.4. 既存のリンクのタイプ	33
2.5. リンクにより保存される情報の属性	34
2.6. フィールドとリンクのヘルプが認識するHTMLタグ	45
2.7. 文字参照	46
3.1. 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	50
3.2. サーバとデータベースの重要なパラメータ	52
3.3. サーバの重要なパラメータ	54
3.4. AssetCenter Windowsクライアント - 言語はオブジェクトの属性に 依存	69
3.5. AssetCenter Windowsクライアント - レコードタイプに基づく表示 言語	69
4.1. Databaseインスタンスのプロパティ	90
4.2. Tableインスタンスのプロパティ	91
4.3. Tableインスタンスのオブジェクト	92
4.4. Fieldインスタンスのプロパティ	93
4.5. Fieldインスタンスのオブジェクト	95
4.6. Typeプロパティの値	96
4.7. UserTypeプロパティの値	96

4.8. Linkインスタンスのプロパティ	97
4.9. Linkインスタンスのオブジェクト	99
4.10. Typeプロパティの値	99
4.11. UserTypeプロパティの値	100
4.12. Indexインスタンスのプロパティ	100
4.13. Indexインスタンスのオブジェクト	101
4.14. Scriptインスタンスのプロパティ	101
4.15. Scriptインスタンスのオブジェクト	102
4.16. FeatDescインスタンスのプロパティ	103
4.17. FeatDescインスタンスのオブジェクト	104
4.18. FeatParamDescインスタンスのプロパティ	104
4.19. FeatParamDescインスタンスのオブジェクト	105
4.20. FeatClassDescインスタンスのプロパティ	106
4.21. FeatParamDescインスタンスのオブジェクト	106
4.22. CalcFieldDescインスタンスのプロパティ	107
4.23. CalcFieldDescインスタンスのオブジェクト	107
4.24. Viewインスタンスのプロパティ	107
4.25. Viewインスタンスのオブジェクト	108
4.26. Pageインスタンスのプロパティ	109
4.27. PageItemインスタンスのプロパティ	109
4.28. PageItemインスタンスのオブジェクト	110
4.29. SysEnumValueインスタンスのプロパティ	111
4.30. Stringインスタンスのプロパティ	111
4.31. ScriptFieldインスタンスのプロパティ	111
4.32. グローバル変数	112
7.1. ODBCドライバの利点	120
8.1. データベースのカスタマイズ - 考慮するテーブル	163
10.1. データベースオプション - 使用可能なオプション	181
13.1. プログラムのアイコンとメニュー	256
13.2. 実行日	260

はじめに（管理）

管理モジュールの対象ユーザ

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

対象ユーザは以下のとおりです。

- ネットワーク管理者
- データベース管理者
- AssetCenter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、以下の操作を実行できます。

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

上記の操作を実行することにより、以下の内容が可能になります。

- 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AssetCenterを最適化する。
- 自動モニタリングプロセスを設置する。
- ユーザプロフィールを作成、更新、削除する。

1 | AssetCenter Database Administrator - インタフェース

はじめに

AssetCenter Database Administratorは、AssetCenter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AssetCenter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。
- AssetCenterに「Admin」でログインすると、データベースを変更できます（オブジェクトの設定、リンクの設定、など）。しかし、AssetCenterとAssetCenter Database Administrator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。
- AssetCenter Database Administrator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。

概要

AssetCenter Database Administratorは、AssetCenter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:

AssetCenter Database Administrator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。最後に使用した文書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**【文書】**項目で**【起動時に前回最後に使用した文書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

AssetCenter Database Administrator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
- AssetCenter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。

AssetCenter Database Administrator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AssetCenter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、AssetCenterのマニュアル『はじめに』を参照してください。

言語の選択

ツールバーの右側には、マルチリンガルデータベースオブジェクトの表示言語（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）を選択するためのフィールドがあります。

▶ オブジェクトの属性に基づく表示言語 [修献 68]

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AssetCenter Database Administrator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]）。

▶ AssetCenterクライアント言語 [修献 67]

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
- 詳細
- 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

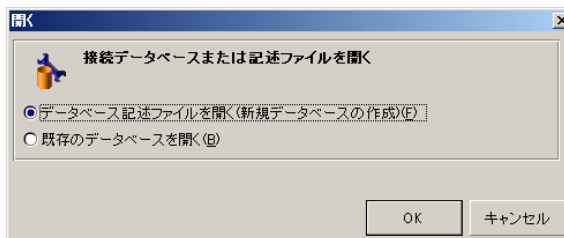
【ファイル】メニューには、ファイルの読み込みと保存に関する機能がまとめられています。

ファイルを開く

【ファイル／開く】メニューを選択します。

【ファイル／開く】メニューを選択すると、次の画面が表示されます

図 1.1. データベースを開く画面



この画面では、AssetCenter Database Administrator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、AssetCenter Database Administrator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つの文書がデフォルトで表示されます。ここから直接選択するとファイルをすばやく開くことができます。メモリに保存される文書の数に指定するには、【編集／オプション】メニューの【表示／タブ】項目の【メモリに格納する文書の最大数】オプションを使用します。

【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）】

AssetCenter Database Administrator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AssetCenter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、AssetCenter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、AssetCenter Database Administrator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AssetCenter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、AssetCenterと同じ接続ウィンドウが表示されます。

本マニュアルでは、 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、マニュアル『はじめに』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AssetCenter Database Administratorセッションが終了します。変更した場合は、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、AssetCenter Database Administratorのタイトルバーに表示されるファイル名にアスタリスク（*）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保存]** メニューで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、AssetCenter Database Administrator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

AssetCenter Database Administrator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、フィールド編集領域以外では使えません。

検索を実行する

AssetCenter Database Administrator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。

AssetCenter Database Administrator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AssetCenter Database Administrator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。

2 データベースの標準記述ファイル

本章では、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、AssetCenter APIをお使いください。

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。

AssetCenterデータベースの構造の記述を理解するには、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、AssetCenter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、使用中のデータベースにAssetCenter Database Administratorソフトウェアを接続して使います。

- AssetCenter Database Administratorプログラム：AssetCenter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。

- AssetCenter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AssetCenter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。AssetCenter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:**

AssetCenter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。

- AssetCenterプログラム

データベースの定義

AssetCenter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。

AssetCenter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。

AssetCenterでは複数のデータベースを作成でき、1回のユーザセッションでは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。AssetCenter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。

Database.txtとTables.txtの構造

データベースの構造は、AssetCenter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、AssetCenter Database Administrator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、AssetCenter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：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。

ファイルには一般的に使う可能性のある情報が多く入力されていますが、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、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、AssetCenterの全言語のバージョンに共通です。

テーブルについて

ここでは、AssetCenter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。

- テーブルの表記名
- テーブルのレコード名の表記文字列

テーブルの表記名

AssetCenter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：AssetCenterテーブルのSQL名は英語で表記され、先頭に「am」が付きます。AssetCenterのすべての言語バージョンで同じSQL名を使っています。
- ラベル：AssetCenterで表示されるテーブル名です。
AssetCenter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：AssetCenter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表記文字列

AssetCenterの各テーブルに含まれるレコード名の表記方法（特殊文字列）を定義するには、AssetCenter Database Administratorを使います。

AssetCenterの各テーブルの特殊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。

また特殊文字列により、ウィンドウ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。

テーブルの特殊文字列には、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

特殊文字列(S): [Model.Brand.Name] [Model.Name] (As

このテーブルの例では、メーカーが「Asus」、モデルが「AsusLX512」、資産タグが「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。

ツリー構造のドロップダウンリストの場合は、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

ここでは、AssetCenter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。

- 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

フィールドの表記名

AssetCenter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。AssetCenter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

"b"

"d"

フィールドのデータ型

Boolean (Yes/No)

日付

接頭コード	フィールドのデータ型
"dt"	Date and time
"i"	Integer (32 bit)
"m"	Monetary
"p"	Percentage
"se"	System itemized list
"ts"	Duration
"mem"	Long text field
なし	Text

- ラベル：AssetCenter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）に使用します。
- 説明：AssetCenter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と入力フォーマット

ここでは、次に内容について説明します。

- AssetCenterのフィールドのデータ型
 - AssetCenterのフィールドのデータ入力時のフォーマットとデータ
- 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型を表示するには、次の方法があります。
- AssetCenter Database Administratorで、**【全般】** タブページの **【データ型】** フィールドを使用します。
 - 「database.txt」ファイルを読み込みます。このファイルは、AssetCenter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。

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 2.1. フィールド - タイプ

「database.txt」 ファイルの [Field Type] 列の値	AssetCenter Database Administrator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

「database.txt」 ファイルの [Field Type] 列の値	AssetCenter Database Administratorの [データ型] フィールドの値	説明
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	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	固定長のテキストフィールド。あらゆる 文字を入力できます
Binary field	可変長バイナリ フィールド。	可変長のバイナリフィールド (BLOB = Binary large object) には、画像や書式な などを、サイズに関係なく保存できます。
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 ドを並べ替えるできません。
Date and time	日付 + 時刻	日付と時刻型のフィールド
日付	日付	日付型のみのフィールド (時刻なし)
Time	時刻	時刻型のみのフィールドのみ (日付な し)

データ入力フォーマットとデータ型

データ入力フォーマットとデータ型は、次のように示されます。

- AssetCenter Database Administrator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 2.2. 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	AssetCenter Database Administratorの [入力タイ 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。
Number	数値	数値
Boolean (Yes/No)	ブール (はい／いいえ)	ブール値
Monetary	金額	金額
Date	Date	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。
Duration	期間	時間型の値が入ります。 使用可能なデフォルトの単 位は、[フォーマット] (UserTypeFormat) プロパ ティで定義します。
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	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。
Percentage	パーセント	パーセントが入ります。 Pourcentage形式では、常に 10進法2桁の形式で表示し ます。

「Database.txt」ファイル の [Field data display and entry type] 列の値	AssetCenter Database Administratorの [入力タイ プ] フィールドの値	説明
Feature value	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。
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。
Table or field SQL 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
N/A	不特定	その他のタイプ

AssetCenter Database Administrator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、フィールドのフォーマット情報を定義できます。

表 2.3. [入力タイプ] のフィールド値

AssetCenter Database Administratorの [入力タイプ] フィールドの値	フォーマット
期間	表示フォーマット %U1[l][d][%U2[l][d]]...[%Un[l][d]] シンタックス： Ux Ux は次のいずれかの単位です。 • Y：年 • M：月 • D：日 • H：時間 • N：分 • S：秒 必要に応じて「d」パラメータに、単位を指定しない場合に適用するデフォルトの単位を指定します。単位は1つだけ割り当てます。 また、「l」パラメータには、必要に応じて期間をLong型で表示することを指定します（例えば98のLong型は1998になります）。 例 • +HM,...H：時間の次に分が表示されます。デフォルトでは時間を入力します。 • %Yl%Mld%DI：年、月、日の順に表示されます。
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
オブジェクト	AssetCenterプログラムの内部処理用に使います。

dtLastModifフィールド

- SQL名が「dtLastModif」のフィールドは、AssetCenter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）：変更日

- 説明：変更日

このフィールドは、AssetCenter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、AssetCenter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。つまり、レコードの変更日または作成日を表します。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でなくインポートしたデータの日付値になります。

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。

例は以下のとおりです。

- 場所のテーブル：場所の完全名は、所在地名の前に親所在地（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、AssetCenter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、AssetCenter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

ここでは、AssetCenter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

AssetCenter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：SQL名は英語です。AssetCenterのすべての言語バージョンで同じ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「lLocalId」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「lLocalId」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。AssetCenter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、【資産】テーブルと、【部署と従業員】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、資産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

AssetCenterの既存のリンクのタイプは、次の表のとおりです。

表 2.4. 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。
Overflow	ソーステーブルまたはターゲットテーブルのレコードが破棄された場合、ソーステーブルまたはターゲットテーブルの対応するレコードも破棄されます。 ソーステーブルまたはターゲットテーブルのレコードが複製された場合、ソーステーブルまたはターゲットテーブルの対応するレコードも複製されます。

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 2.5. 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。
コメント	コメントフィールドを保存します。
画像	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。
任意管理項目値	任意管理項目の値を保存します。

リンク数

AssetCenter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つのレコードを、テーブルBの1つのレコードにリンクできます。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ヶ所の所在地に関連付けられます。
- 「1-->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。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件の契約が複数の資産に関連付けられます。

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。

中間テーブル

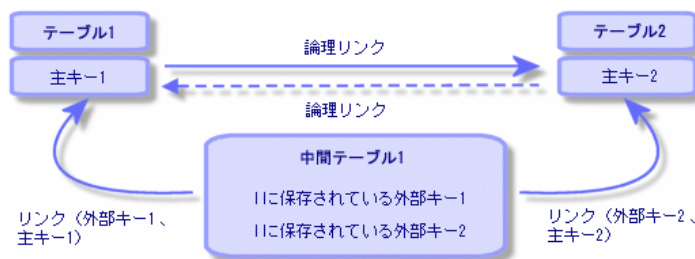
中間テーブルは、「n-n」タイプの論理リンクの場合にのみ使われます。

このテーブルは、AssetCenter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。

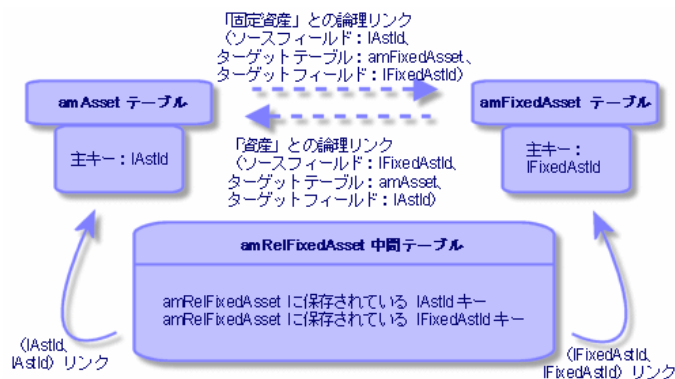
次に、中間テーブルと2つのテーブルの関係図を示します。

図 2.1. 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 2.2. **【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（オブジェクトのSQL名：Assets））。
- **【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、AssetCenter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テーブル（amSoftInstall）
- 2 電話テーブル（amPhone）
- 3 コンピュータテーブル（amComputer）

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 資産テーブル（amAsset）

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。

参照テーブルは以下の通りです。

- ポートフォリオ品目テーブル (amPortfolio)
- 契約テーブル (amContract)
- 作業指示テーブル (amWorkOrder)
- ケーブルテーブル (amCable)
- 研修テーブル (amTraining)

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

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

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。

手順は以下の通りです。

- 新しいテーブルとオーバーフローリンクを AssetCenter Database Administrator で作成します。
 - その動作を AssetCenter 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 以下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]) 。
 - 以下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:** ポートフォリオ
 - **ターゲットフレーム、テーブル:** ポートフォリオ品目 (amPortfolio) 、**Nom SQL:** 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。
以下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。

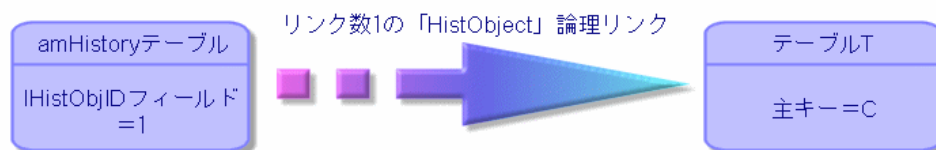
- 4 オーバーフローリンクの画面を作成するには:
以下のようにフィールドに入力します。
 - **【全般】** タブ:
 - 1 **SQL名:** scrCar
 - 2 **ラベル:** Cars
 - 3 **説明:** Cars
 - **【ページ】** タブ:
 - 1 **【詳細ページ】** フレームに、scrGen carとPgGenDocのページを追加します。
- 5 データベースを保存します。
- 6 AssetCenterを開き、**【属性】** 画面を表示します。
- 7 **【オーバーフローテーブル】** フィールドに新しいリストデータ値「Car」が入っていることを確認します。
- 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。

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 2.3. 【履歴】 テーブルとターゲットテーブルとのリンク



インデックスについて

ここでは、AssetCenter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。AssetCenterのすべての言語バージョンで同じ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属性

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。

AssetCenter Database Administrator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

【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得ません。

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。

データベースのカスタマイズ

AssetCenterのデータベースは、社内の業務形態に合わせてカスタマイズすることができます。

この場合のカスタマイズは、データベースの管理者が行います。

変更した結果は、AssetCenterのユーザの画面でも確認できます。管理者が定義した設定は、すべてのユーザのデータベース画面に同じように表示されます。管理者がフィールドまたはリンクの名前を変更すると、AssetCenterの各ユーザの画面上（特にリスト画面、詳細画面、クエリの作成画面など）にもその名前で表示されます。

ここでは、AssetCenterデータベース内の様々なオブジェクトをカスタマイズする手順について詳しく説明します。

テーブルをカスタマイズする

テーブルの「説明」、「ラベル」、「特殊文字列」は、AssetCenter Database Administratorを使ってカスタマイズできます。

フィールドとリンクをカスタマイズする

データベースのフィールドおよびリンクをカスタマイズするには、次の方法があります。

- **【オブジェクトの設定】** ポップアップメニューを使う。
【オブジェクトの設定】 ポップアップメニューを表示するには、設定するフィールドに移動し、マウスの右ボタンをクリックします。

注意:

ショートカットメニューを使った場合、変更が記録されるのは、データベースを閉じて承認メッセージを確認するか、**【管理／データベース設定の保存】** メニューを選択した場合です。

- AssetCenter Database Administratorを使う。
AssetCenter Database Administratorの**【全般】** および**【スクリプト】** タブページを使います。これらのタブページでは、次の内容を定義できます。
 - フィールドとリンクのラベル
 - フィールドとリンクの説明
 - フィールドとリンクのデフォルト値
 - 必須のフィールドとリンク

- ログ変更履歴を記録するフィールドとリンク
- 読み出し専用のフィールドとリンク

 注意:

データベースの作成時に、AssetCenter Database Administratorでテキスト型のフィールドの最大長を設定することもできます。

- AssetCenter Database Administratorの【ヘルプ】タブページで、フィールドまたはリンクのヘルプの内容を次の構成に従って変更することもできます。
 - 説明：フィールドまたはリンクの内容を説明します。
 - 例：フィールドまたはリンクの入力例を示します。
 - 注：データ入力に関する注意事項、関連の自動処理機能などを示します。

フィールドのヘルプシステムは、HTML言語のサブセットです。

インデックスをカスタマイズする

データベース内のインデックスをカスタマイズするには、AssetCenter Database Administratorを使います。

インデックスごとに、次の定義が可能です。

- ラベル
- 説明

フィールドとリンクのデフォルト値

フィールドとリンクのデフォルト値は、次の値で構成されます。

- 引用符 (" ") で囲まれた定数値
- スクリプト関数が返す値
- データベース内で参照する他のフィールドの値

**注意:**

AssetCenterで作成した新しいレコードには、自動的にデフォルト値が適用されます。そのレコードを作成または変更したユーザは、デフォルト値を変更できます。

デフォルト値は、レコード作成時にのみ適用されます。

例えば、レコードAを作成したときにフィールドxのデフォルト値がレコードBのフィールドyを参照している場合は、レコードAを作成した後にフィールドyまたはレコードBを変更しても、フィールドxの値は変わりません。

例

```
RetVal="DS"+AmCounter("納入",2)+AmDate()
```

- AmDate()は、レコードの作成日を表します。
- AmCounter("納入",2)は、1ずつ増減する「納入」カウンタの現在の値を2桁で表示することを表しています。

特殊フィールドの場合

【特殊文字列】および【BASICスクリプト】タイプの特殊フィールドのみが、標準フィールドのデフォルト値の計算に使用されます。

【コメント】テーブル（SQL名：amComment）へのリンクの場合

このテーブルへのリンクに、デフォルト値を割り当てることはできません。

例

【コメント】（SQL名：Comment）

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、すべてのユーザが参照できるフィールドとリンクを参照するように設定してください。

フィールドのデフォルト値のカウンタ

フィールドのデフォルト値がカウンタを参照するように定義することができます。

カウンタを定義すると、そのフィールドを含んでいるレコードを新しく作成するたびに、AssetCenterが前のレコードの数値に1を足した数値を新しいレコードに挿入します。

カウンタは、管理者が【管理／カウンタ】メニューを使って管理します。

フィールドのデフォルト値の式にカウンタを挿入するには、このメニューを使って事前にカウンタを作成しておく必要があります。カウンタ名にスペース、「\$」、「|」、または「|」の記号を使うことはできません。

必要な数だけカウンタを作成できます。

AssetCenterユーザのインタフェースには、カウンタ名は表示されません。

【新規作成】ボタンをクリックすると、直ちにカウンタが増加します。【作成】ボタンをクリックしないでレコードの作成をキャンセルしても、カウンタの数値は減少しません。

カウンタの値は再調整できます。

カウンタの詳細画面に表示される値は、最後にカウンタを実行したときの結果です。



注意:

AmCounter(<カウンタ名>)の代わりにAmCounter(<カウンタ名>, [n])などの式をフィールドのデフォルト値に入力すると、カウンタの値はn桁で表示されます。

デフォルト値の記述エラー

デフォルト値を記述するときに犯しやすい間違いは、以下のとおりです。

実行エラー - タイプエラー

変数がフィールドのデータ型と互換性がない場合があります。例えば、日付型のフィールドにAmLoginName()を定義する場合はこれに該当します。

フィールド「XXX」が「XXX」に存在しない

リンクを参照することを定義する場合は、名前でリンクを参照するように「Link.Link.Field」フォーマットを使う必要があります。このフォーマットによって、AssetCenterは直接リンクをたどることができます。

例えば、資産のデフォルトの責任者を設定する場合は、
「CostCenter.Supervisor」と指定するとコストセンタの責任者をデフォルト値として適用することができます。
AssetCenterのユーザが資産のコストセンタのフィールドに入力すると、その値が直ちに資産の責任者フィールドにデフォルト値として入力され、データベースに保存されます。

フィールドとリンクのヘルプが認識するHTMLタグ

AssetCenterフィールドの状況依存ヘルプシステムでは、HTML言語のサブセットを使用しています。主にツール名の書式設定などに使われている数種類のタグを採用しています。ここでは、HTML言語の詳しい説明は省略し、AssetCenterで管理するHTMLタグについてのみ簡単に説明します。

 注意:

ユーザが入力しなければならないHTMLタグの一覧は、次の表のとおりです。これらのタグは、その他のオンラインヘルプに使用されている表記規則とは異なります。

表 2.6. フィールドとリンクのヘルプが認識するHTMLタグ

HTMLタグ	説明
	このタグに続くテキストを表示するフォントを定義します。このタグの後に別のフォントタグの指定がない限り、このフォントが適用されます。
	このタグに続くテキストを表示するフォントの色を定義します。このタグの後に別の色タグの指定がない限り、この色が適用されます。
	フォントサイズを"n"レベル大きくします。このタグの後に別のサイズタグの指定がない限り、このサイズが適用されます。
	フォントサイズを"n"レベル小さくします。このタグの後に別のサイズタグの指定がない限り、このサイズが適用されます。

HTMLタグ	説明
 	この2つのタグで囲んだテキストが太字で表示されます。
<I> </I>	この2つのタグで囲んだテキストが斜体で表示されます。
	このタグは、箇条書きの開始を表します。テキストを改行し、テキストの先頭に行頭記号を挿入します。
<HR>	セクションの区切りを示す水平線を引きます。

HTML言語の詳細については、HTMLに関する書籍を参照してください。

文字参照

HTML言語の文字参照は次の表のとおりです。これらの文字を入力しても、画面にはそのとおりには表示されません。希望の文字を表示するには、対応するタグを代わりに入力する必要があります。

表 2.7. 文字参照

表示する文字	文字を表示するためのタグ（文字参照）
"<"	"<"
"&"	"&"
連続するスペース（空白文字）	" "

3 | AssetCenterデータベースの作成

データベースを作成するには、以下の手順に従います。

- 1 DBMSを使って空のデータベースを作成します。
- 2 空のシェルへの接続を指定します。

 注意:

データベースを作成するには、空のシェルを作成する必要があります。作成方法は使用しているDBMSによって異なり、通常はデータベース管理者が作成します。詳しくは、お使いのDBMSの説明書を参照してください。

-
- 3 AssetCenter Database Administratorを使ってデータベースを作成します。
 - 4 データベースのライセンスファイルを選択します。ライセンスファイルによって、AssetCenterのライセンス契約で既定されているアクセス権限が決まります。
 - 5 管理者のパスワードを入力または変更します。管理者のログインおよびそのパスワードは、DBMSで空のシェルを作成した時に定義したものを 사용합니다。
 - 6 AssetCenterデータベースの構成要素（テーブル、フィールド、リンクなど）をすべて空のシェル内に作成します。

- 7 AssetCenter Serverを起動してから、作成したデータベースに接続します。この操作によりライセンスファイルが有効になり、他の一般ユーザがデータベースにアクセスできるようになります。

DBMSを使って空のAssetCenterデータベースを作成する

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。

本マニュアルは、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のマニュアル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、データのセキュリティ、AssetCenterソフトウェアの性能、およびアーキテクチャのスケーラビリティが左右します。

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の種類（例：Windows NT）
- 企業内の規則
- 通常使用している管理ツール
- 多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。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、ペレグリンシステム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）を含むディスク用には、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、Sybase Adaptive ServerおよびUDB DB2では「データベース」という用語が用いられていますが、Oracleでは「スキーマ」という用語が使われています。

DBMSを使ったデータベース作成の手順

データベースの作成手順は以下のとおりです。

- 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、AssetCenterの接続で使用することも可能です。この場合、AssetCenter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。

注意事項

AssetCenter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350個
- インデックス1400個



ヒント:

AssetCenter Database Administrator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。

Sybase Adaptive Serverの例：同時にオープン可能なオブジェクトの数は、「sp_configure」のパラメータ「number of open objects」により指定されます。

Oracle

表 3.1. 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
文字セット	半角英数字を使う言語（英語、フランス語、ドイツ語、イタリア語、スペイン語など）用には、「WE8ISO8859P15」文字セットをお勧めします。 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。 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	データベースのメモリキャッシュのサイズ このサイズは、db_block_sizeの単位で表記されています。 200 MBのキャッシュの例： - db_block_buffers = 25000 if db_block_size = 8192 - db_block_buffers = 100000 if db_block_size = 2048 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。 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。）
shared_pool_size	Oracle 8.0.xの場合：ユーザ数（10から50）に応じて、 9 000 000 から 1 5000 000 クライアント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）。
log_buffer	このパラメータの単位はバイトです。 「163840」をお勧めします。
processes	<同時ユーザ数> + <Oracleシステムプロセス数>。 10ユーザまでは、「50」が許容されます。 Oracleのライセンスで許可されている権限に応じて指定します。
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（AssetCenter Server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	MTS以外のサーバの場合、「sort_area_size」はメモリのソート用のメモリのバイト数に当たります。
sort_area_retained_size	「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。 「65256」とまず指定し、必要に応じてこの値を増加します。 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。65256 * <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。

AssetCenter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）、AssetCenter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、AssetCenter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：10MB）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AssetCenter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または2000を準備する

Microsoft SQL Enterprise Managerユーティリティを使用します。

表 3.2. 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 NT4で最低でも64MB、同時ユーザ250人用に128MB）。Windows 2000の場合は、128から256MBを確保します。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モ用データベースで指定されている値	すべてのAssetCenter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

パラメータ	デフォルト値	推奨値
auto update statistics	データベースの作成時にデモ用データベースで指定されている値	すべてのAssetCenter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 AssetCenter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」を割り当てるようお勧めします。

- 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（ODBC SQL Serverドライバ）を各クライアントコンピュータにインストールします。

Sybase Adaptive Server

表 3.3. サーバの重要なパラメータ

パラメータ	デフォルト 値	推奨値
memory	7 500	このパラメータはメモリキャッシュ（特にデータとプロシージャを保存するキャッシュ）の合計サイズを指定します。 この値は、2KBのブロック単位で表記されています。 サーバに512MBのRAMがある場合、384MBのキャッシュが適当です（パラメータは、192000となります）。
procedure cache	30	解析とSQLクエリを保存するために割り当てられたメモリキャッシュです。 この設定は%で表示されます。 メモリキャッシュの拡張に反比例して少なくする必要があります。 384KBのメモリキャッシュの場合、約3から10%に指定します。
maximum network packet size	512	8 192
default network packet size	512	512
additional netmem	0	<ユーザの接続数> * 3 * (<ネットワークパケットの最大サイズ> - <ネットワークパケットのデフォルトサイズ>)
extent i/o buffers	0	(<メモリ> / 8 MB + 1) * 10
tempdb	2MB	少なくとも20MB。必要に応じて増やします。

データベースの作成手順

- 1 Sybase Adaptive Serverを使用します。
- 2 例えばSybase Centralを起動します。
- 3 データベースを作成するのに十分なディスク領域 (database device) を確保します (サイズの小さいAssetCenterデータベースの場合、少なくとも150MB)。
- 4 少なくとも150 MBのDATAセグメントを使ってSybaseデータベースを作成します (サイズの小さいデータベースの場合)。作成後にデータベースオプションで、[Truncate Log on Checkpoint] チェックボックスをオンにします。LOGを使用する場合は、DATAセグメントの分離セグメント上または追加セグメント上のいずれの場合も、約20MBを確保します。
- 5 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。
- 6 少なくとも30MBのメモリキャッシュを使用するようSybaseを設定します。サーバで使用可能なメモリに応じて、この値を最大化するようお勧めします。サーバがスワップしない限り、パフォーマンスが大きく向上します。

AssetCenterでデータベースを作成した後のロッキングモード

バージョン11.9を使用する場合は、**Data Only Locking** (特に、競合を減らす **Row-Level Locking**) を利用するよう強くお勧めします。これには、データベース管理者がAssetCenterのすべてのテーブルを作成後に設定しなおす必要があります (各テーブル: `alter table <table> lock datarows`)。

サイズの大きいテーブル (【履歴】テーブル、【コメント】テーブル) で実行するクエリのパフォーマンスが低下すると判断される場合は、これらのテーブル用のキャッシュを設置します。この結果、これらのテーブルでクエリを実行すると、大きいテーブルのデータが他のテーブルのデータをキャッシュで上書きすることを避けられます。

クエリの実行速度を向上させるためには、AssetCenter ServerまたはSQLスクリプトを使ってテーブルの統計を実行する必要があります。

DB2 UDB

ここで説明する情報は、DBMSの設定後、そしてDBMSでAssetCenter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AssetCenter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

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。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望ましいです。

- CREATE DATABASE <データベース名>
- Update database configuration for <データベース名> using APPLHEAPSZ 2048
- Update database configuration for <データベース名> using APP_CTL_HEAP_SZ 2048
- Update database configuration for <データベース名>; using DBHEAP 4096
- Update database configuration for <データベース名>; using LOGFILSIZ 500
- Update database configuration for <データベース名>; using DFT_QUERYOPT 2
- CATALOG TCPIP NODE <DB2_NT|SUN_NODE|AIX_NODE|Etc.> REMOTE <サーバ名> SERVER 50000 REMOTE_INSTANCE DB2 SYSTEM <サーバ名> OSTYPE <NT|SUN|AX|Etc.>
- CATALOG DATABASE <データベース名>; AT NODE <DB2_NT|SUN_NODE|AIX_NODE|Etc.>
- AssetCenterデータベースでアーカイブ機能を使用する場合、DB2は8Kページを処理する必要があります。

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

```
CREATE Bufferpool ACBP DEFERRED SIZE 250 PAGESIZE 8 K ;

CREATE REGULAR TABLESPACE ACTS PAGESIZE 8 K MANAGED BY
SYSTEM USING ('F:\DB2\ACTS') EXTENTSIZE 16 OVERHEAD 10.5 PREFETCHSIZE 16 TRANSFERRATE 0.33 BUFFERPOOL ACBP;

CREATE SYSTEM TEMPORARY TABLESPACE ACTMPTS PAGESIZE 8
K MANAGED BY SYSTEM USING ('F:\DB2\ACTMPTS') EXTENTSIZE 16 OVERHEAD 10.5 PREFETCHSIZE 16 TRANSFERRATE 0.33 BUFFERPOOL ACBP;
```


カウンタのアプリケーションサーバを準備する

例えば、ORACLEでは使用できてもDB2 UDBでは使用できない**SEQUENCE**サービスを置換えるために、カウンタのアプリケーションサーバをDB2 UDBサーバにインストールすることができます。

これは、データベースに多数のクライアントが接続する場合に推奨されません。

注意:

このサービスのインストールは、任意です。Javaの知識を持つデータベース管理者がインストールを行います。インストールしない場合は、各論理接続に対して、AssetCenterクライアントの物理接続が2つ必要になります。

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。**system 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

データベースエンジンがWindows NT 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）

```
CATALOG TCPIP NODE MyNode REMOTE CALIFORNIUM SERVER 50000 R
EMOTE_INSTANCE INST_1 SYSTEM CALIFORNIUM OSTYPE NT
```

次に実行するコマンド：

```
CATALOG DATABASE SAMPLE AS SAMPLE AT NODE MyNode
```

次に実行する標準コマンドライン：

```
db2icrt MyNode
```

 ヒント:

サーバがWindows NT 4で設定されていない場合は、データベース管理者に問い合わせてください。

Runtime Sybase SQL Anywhere

AssetCenterと共に提供されるSybase SQL AnywhereのRuntimeバージョンは、AssetCenterのインストール時に自動的に提案されます。Sybaseのクライアント-サーババージョンを使用する場合、Sybase SQL Anywhere Runtimeをインストールする必要はありません。

Sybase SQL AnywhereのRuntimeバージョンがインストールされていて、空のデータベースを作成する場合は、以下の手順に従います。

- 1 ハードドライブに新しいフォルダを作成します（オプション）。
- 2 AssetCenterのインストール先フォルダに格納されている「empty42.db」ファイルをコピーします。

 ヒント:

このファイルの名前は変更可能です。

- 3 AssetCenterを起動します。
- 4 **【ファイル／接続の編集】** メニューを使用します。
- 5 **【新規作成】** をクリックします。
- 6 **【名前】** フィールドに入力します。
- 7 Sybase SQL Anywhereエンジンを選択します。
- 8 **【データソース】** フィールドの右側にある  をクリックします。
 - 1 **【Database File】** フィールドに、先ほどコピーした「Empty43.db」ファイルのパスと名前を入力します。**【Browse】** ボタンが使用できます。値の中から **【Custom】** を選択します。
 - 2 **【Data Source Name】** フィールドの値を指定します。
 - 3 **【オプション】** をクリックします。
 - 4 **【Start Command】** フィールドに「rtdsk50.exe」と入力します。
 - 5 **【Database Switches】** フィールドは空欄のままにしておき、**【Autostop Database】** チェックボックスをオンにします。
 - 6 **【OK】** をクリックして確定します。
 - 7 **【OK】** をクリックして、ODBCの設定フィールドを閉じます。
- 9 **【ユーザ】** フィールドに「itam」と入力し、**【パスワード】** フィールドに「password」と入力します。
- 10 **【作成】** をクリックします。

11 [テスト] をクリックして接続をテストします。

DBMSログインを作成する

AssetCenter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、AssetCenter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AssetCenterを設定することもできます。これにより、外部ツールを使った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。本マニュアルでは、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AssetCenter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 AssetCenter identified by <password> default tablespace <asset> temporary tablespace <temporary_data> profile default;  
grant connect, resource, create session to AssetCenter;
```

AssetCenterと共に使用するDBMSを変更する

DBMSを変更する場合は、AssetCenterの適切なバージョンを取得し、インポートモジュールを使って古いデータベースをインポートする必要があります。

AssetCenterデータベースを削除する

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

AssetCenter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。

- 例えばSybase Adaptive Serverの場合、**SQL Server Manager**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。
- 2 各クライアントコンピュータで、AssetCenter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。

ライセンス上の権限と使用可能な機能

AssetCenterは、共通のリポジトリを基盤とした統合機能を提供する情報テクノロジー管理の完全システムです。以下の機能を使用できます。

- ポートフォリオ
- 調達
- 契約
- ファイナンス
- ケーブル

AssetCenterでは、同一のインタフェースから上記の全機能にアクセスし、共通のデータベースを使用します。このため、データの重複を避けることが可能です。

注意:

取得するライセンス上の権限に応じて、使用できる機能は異なります。

ライセンス上の権限は、ペレグリンシステムズから取得するライセンスファイル「license.cfg」に格納されています。このライセンスファイルは、データベースに挿入する必要があります。

データベースにライセンスファイルを挿入する方法については、本章の「**ライセンスファイルを選択する**」の節を参照してください。

AssetCenterは機能的に設計されているため、ユーザのニーズに応じてアクセスする機能のみを選択し、インタフェースを簡略化することができます。

注意:

各クライアントコンピュータで、希望する機能を起動するよう指定できます（**【ファイル／モジュールの起動】メニュー**を使用）。

空のシェルへの接続を指定する

空のシェルへの接続を指定するには、次の手順に従います。

- 1 AssetCenterを起動します
- 2 **【ファイル／データベース接続の管理】** メニューを選択します。
- 3 **【新規作成】** をクリックします。
- 4 **【接続】** タブページでデータベース名、DBMS、およびそのDBMS特有の情報を入力します。
- 5 次に**【作成】** ボタンをクリックしてデータベースを作成します。



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。

管理者パスワードを入力してデータベース構造を作成する

データベース作成の最初の手順で、管理者のパスワードを入力する必要があります。管理者のログインおよびパスワードは、データベース管理ツールを使って空のシェルを作成するときに定義します。AssetCenterでは、データベース構造を作成する前にこの管理者パスワードを変更することができます。この処理にかかる時間は、お使いのDBMSおよびコンピュータの速度によって異なります。

データベースの作成が完了したら、AssetCenter Serverを起動してデータベースに接続し、ライセンスを確定して、他のユーザがアクセスできるようにデータベースを設定します。

AssetCenter Database Administratorを使ってデータベースを作成する

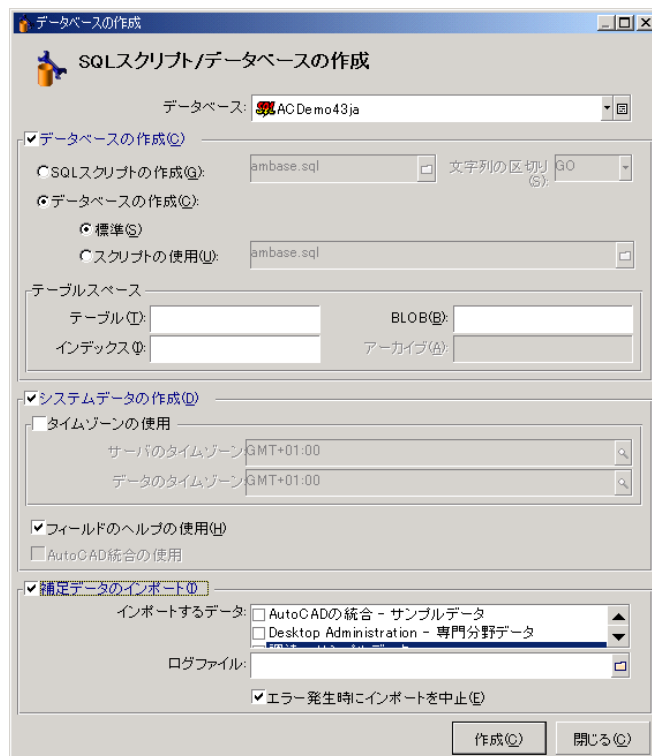
AssetCenter Database Administratorを使うと、開かれているデータベースの記述ファイルを基にAssetCenterデータベースを作成できます。

データベースを作成するには、次の手順に従います。

- 「AssetCenter/Config」フォルダに格納されている記述ファイル「gbbase.xml」を開きます。

- データベースの作成画面を開きます（【アクション／データベースの作成】メニュー）。

図 3.1. データベースの作成画面



[データベース] 枠のオプション

このオプションで、既存のAssetCenter接続を選択し、テーブルのテーブルスペースとインデックスを定義します。テーブルスペースは、事前にDBMSを使って作成する必要があります。

[データベースの作成] 枠のオプション

この枠内のオプションは、【データベースの作成】チェックボックスをオンにすると有効になります。次の作業が可能になります。

- **【データベースの作成】** と **【標準】** ボタンを選択すると、データベースを直接作成できます。
- **【SQLスクリプトの作成】** ボタンを選択すると、後でデータベースを作成するために、SQLスクリプトを作成できるようになります。☐をクリックしてスクリプトを指定し、ドロップダウンリストからSQLの区切り文字を定義します（代表的な区切り文字は、Oracleの場合はスラッシュ (/)、他のデータベースの場合はGOなどです）。

注意:

ドロップダウンリストは編集可能です。セミコロン (;) など、他の有効な区切り文字を自由に定義できます。<MySeparator>などの区切り文字も定義できますが、データベース作成スクリプトは機能しなくなります。

- **【データベースの作成】** ボタンと **【スクリプトの使用】** ボタンを選択した場合は、**【SQLスクリプトの作成】** オプションで作成するデータベース作成スクリプトを使って、データベースを作成します。☐をクリックして、SQLスクリプトを指定します。

アーカイブを有効にするメニュー

このメニューは、データベース全体のアーカイブを有効にします。

アーカイブの詳細については、[アーカイブ](#) [惨献 171]の章を参照してください。

【システムデータの作成】 枠のオプション

【システムデータの作成】 チェックボックスをオンにすると、AssetCenter Database Administratorは次のシステムデータを作成します。

- 1 システムテーブル（**【システムテーブル】**（SQL名：SysBlob）テーブル）に保存されるデータベース記述ファイル
- 2 暗号化され、**【sysblob】** テーブルに保存される「Admin」ログインのパスワード（デフォルトでは空）
- 3 「Admin」ユーザ
- 4 各テーブルの識別子がNULLのレコード（外部への接続のシミュレーションに使う）
- 5 リストデータとカウンタ

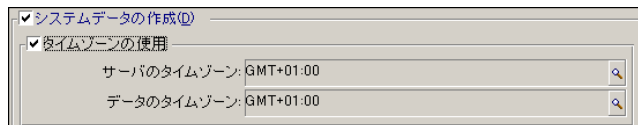
【タイムゾーンの使用】 チェックボックスをオンにすると、AssetCenter Database Administratorは次のシステムデータを作成します。

- **【アプリケーションのオプション】**（SQL名：amOption）テーブルに保存されるタイムゾーンに関する情報



注意:

【タイムゾーンの使用】 オプションを選択すると、サーバとデータのタイムゾーンをGMTで定義できます。



【フィールドのヘルプの使用】 チェックボックスをオンにすると、フィールドのヘルプ情報を入力できるようにAssetCenter Database Administratorがデータベースを調節します。

AssetCenterでAutoCADの統合機能を使用する場合は、**【AutoCADの統合の使用】** オプションを選択します。

【補足データのインポート】 枠のオプション

データベースの作成時に、AssetCenter**専門分野データ**を自動的にインポートするように選択できます。

- 1 **【補足データのインポート】** を選択します。
- 2 **【インポートするデータ】** ドロップダウンリストから、インポートする情報を選択します。必要に応じて、**専門分野データ**の一部またはすべてをインポートするよう指定できます。



注意:

データベースの作成後に、AssetCenterのインポート機能を使って、同じ情報をインポートすることも可能です。

- 3 場合に応じて**【ログファイル】** フィールドでパスを指定します。ログファイルには、インポート時に実行される処理やエラー／警告メッセージが記録されます。
- 4 **【エラー発生時にインポートを中止】** オプションを選択すると、インポート中にエラーが発生するとインポートが中止されます。

ライセンスファイルを選択する

ご使用のAssetCenterライセンスに対応するデータベースのアクセス権を有効にするには、AssetCenterでライセンスファイルを指定する必要があります。ライセンスファイルはデータベースに直接保存されます。クライアントのコンピュータで定義する必要はありません。

このライセンスを取得するには、ペレグリンシステムズ社に必要な情報を提供する必要があります。その情報とライセンスファイルは、データベースの作成開始直後に表示されるダイアログボックスで入力します。

ペレグリンシステムズ社に連絡する前に、次の作業を行います。

- 1 ライセンスファイルを定期的に検証するAssetCenter Serverのインスタンスを実行するコンピュータを定義します。これは【データベースサーバに信号送信】モジュール（AssetCenter Serverの【ツール／モジュールの設定】メニュー／【モジュール】タブで指定）を起動するコンピュータです。このAssetCenter Serverの設定は、なるべく他のコンピュータに移さないでください。移動させると、ライセンスファイルを変更する必要があります。
- 2 このコンピュータのネットワークカードのMACアドレスを指定します。

注意:

コンピュータのネットワークカードのMACアドレスを検出するには、そのコンピュータでAssetCenter Serverを実行し、【ヘルプ／バージョン情報】メニューを選択して【詳細】をクリックします。このためにデータベースに接続する必要はありません。

警告:

Windows 95でAssetCenter Serverを使用する場合は、NetBEUIプロトコルが設定されている必要があります。これにより、AssetCenter Serverは、ネットワークカードMACアドレスを適切に識別できます。

注意:

Windows NTでAssetCenter Serverを使用することをお勧めします。

ライセンスファイルによって次の内容が決まります。

- 認証されたユーザの数
- 作成可能な資産および本体の最大数
- 使用可能なDBMS
- 使用可能な機能

確認するには、次の操作を実行します。

- 1 **【ファイル／データベース接続の管理】** メニューを使って接続を指定したら、**【テスト】** をクリックしてデータベースを作成します。
- 2 ライセンスのダイアログボックスが表示されたら、ペレグリンシステムズ社にご連絡ください。
- 3 ペレグリンシステムズ社の担当者に **【AssetCenter ServerのMACアドレス】** フィールドの値と会社名を報告します。
- 4 ペレグリンシステムズ社の担当者からライセンスファイル入手します。

 注意:

AssetCenter Serverを実行するコンピュータのネットワークカードを変更する場合は、ペレグリンシステムズ社に連絡して代替のライセンスファイル入手する必要があります。

- 5 ライセンスファイルが有効になると、AssetCenterでデータベースを作成するダイアログボックスが表示されます。使用許諾契約で許可されている場合は、このダイアログボックスでタイムゾーン情報を指定できます。タイムゾーンを指定するには、**【タイムゾーンの使用】** オプションを選択し、**【サーバのタイムゾーン】** および **【データのタイムゾーン】** を選択します。
- 6 次の手順に進むには、**【作成】** をクリックします。

ライセンスを変更する

AssetCenter Serverを実行しているコンピュータのネットワークアダプタを取り替えた場合、またはライセンスの使用期限が切れた場合は、お使いのデータベースのライセンスファイルを変更する必要があります。ライセンスファイルは、データベースに接続せずにAssetCenter Database Administratorで変更できます。

AssetCenter Database Administratorを用いてライセンスファイルを変更する

- 1 AssetCenter Database Administratorを起動します。
- 2 ライセンスファイルを変更するデータベースに接続します。
- 3 **【アクション／ライセンスファイルの編集】** メニューを選択し、表示されるダイアログボックスでライセンスファイルを入力します。

AssetCenterクライアント言語

概要

AssetCenter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](#) [参考文献 68]）。

AssetCenter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](#) [参考文献 67]）。
- 2 次に、対応する言語のバージョンのAssetCenter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（▶ [AssetCenterクライアントレベルでの特定言語の表示](#) [参考文献 68]）

AssetCenter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す。

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

AssetCenter Database Administratorでデータベースを作成する際には、AssetCenter Database Administrator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AssetCenterをインストールします。

ヒント:

AssetCenter Database Administrator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AssetCenter Database Administrator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AssetCenter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\Peregrine\AssetCenter-English

- 3 [OK] をクリックして確認します。



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。

AssetCenter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AssetCenterデータベースに追加されたら:

- 1 必要な言語でAssetCenterのWindowsクライアントをインストールします。
- 2 AssetCenterクライアントを起動します。
- 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:

AssetCenter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。すなわちクライアント自身の言語です。

オブジェクトの属性に基づく表示言語

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 3.4. AssetCenter Windowsクライアント - 言語はオブジェクトの属性に依存

オブジェクトの属性	オブジェクトのテキストの保存	AssetCenterクライアントによる表示言語
- テーブル、フィールド、リンク、インデックスのラベルと記述 - 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	データベース	データベースに接続されたAssetCenterクライアントの言語
データベースに保存されたデータ	データベース	ユーザがデータを入力した言語
ワンポイント	データベース	データベースの作成に用いられたAssetCenter Database Administratorの言語
プログラムメッセージ	クライアントプログラム	データベースに接続されたAssetCenterクライアントの言語

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 3.5. AssetCenter 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
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。
書式	レポートと同じ原則
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

レコードタイプ	コメント
デモ用データベース	単一言語（AssetCenterのインストール言語） 他の言語パラメータも、プロダクション用データベースと同じ方法で追加できます。
ワークフローチャート	ワークフローチャートは、作成に用いられた言語と無関係に実行できます。 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。 ワークフローチャート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。
マニュアル	クライアントのデスクトップにAssetCenter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。

DBMSを変更する

AssetCenterでは、AssetCenterのデータベースを使用するために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入力し、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、AssetCenter 3.xをバージョン4.3へ出力することは不可能です。

以下の手順に従います。

- 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は本節では説明されていません。詳細については本章の以下の節を参照してください。

- DBMSを使って空のAssetCenterデータベースを作成する [修献 48]
 - 空のシェルへの接続を指定する [修献 61]
 - AssetCenter Database Administratorを使ってデータベースを作成する [修献 61]
-

データベースの構造とデータを出力する

出力するには、次の手順に従います。

- 1 **【開く／既存のデータベースを開く】** メニューでAssetCenterデータベースを開きます。
 - 2 **【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。
 - 3 出力名を指定し、**【保存】** をクリックします。
AssetCenter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、自動的に作成された複数のファイルから構成されます。ダンプファイルには、「xxx01.ar」、「xxx02.ar」、「xxx03.ar」のような名前がつきます。xxxが出力の名前です。

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する必要があります。

- 1 **【アクション／データベース構造とデータのインポート】** 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 AssetCenter Database Administratorが生成した第1の出力ファイル (xxx01.ar) を選択します。
AssetCenter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。

4 データベース記述ファイルの作成

AssetCenter Database Administrator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。

AssetCenter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、AssetCenter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、使用中のデータベースにAssetCenter Database Administratorソフトウェアを接続して使います。

- AssetCenter Database Administratorプログラム：AssetCenterデータベース（テーブル、フィールド、リンク、インデックス）の記述ファイルを以下の方法で作成できます。
 - AssetCenter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AssetCenter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。AssetCenter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:**

AssetCenter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、データベースのカスタマイズ情報をトラッキングする際に便利です。

- AssetCenter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】：AssetCenter Database Administrator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：AssetCenter Database Administrator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、AssetCenter Database Administrator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ことは、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ことです。AssetCenter Database Administrator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多少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次節で説明します。

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

AssetCenter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

AssetCenter Database Administratorで使うテンプレートには、抽出する情報と、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、AssetCenter Database Administrator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  
$$
```

次のように入力されます。

```
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、AssetCenter Database Administratorが認識できないようにします。

例

```
$ これはコメントです。
```

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:\modeles\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反復型で参照する場合は、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>
...
```

```
$endifor
$endifor
```

例

```
$for Tables
...
$for Fields
...
$endifor
$endifor
```

コレクション間の階層に注意する必要があります。例は次のとおりです。

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。

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

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

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, UserTypeFormat 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

AssetCenter Database Administrator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\putput.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。この関数とその後に記述する関数は、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、それ以外の場合はすべて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、それ以外の場合はすべて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

<fonction>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 = "Date and time"
case 9: StrType = "Binary field"
case 10: StrType = "Date"
case 11: StrType = "Time"
case 12: StrType = "Long text field"
case else
    Dim strError as String
    strError = "Type " + CStr(iValue) + " undefined"
    strType = LogError(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 $(IsLast) = 1
$(Sqlname)
$else
$(Sqlname)$nocr
$nocr
$endif
$endfor
...
```

次の出力が得られます。

```
...
amProduct Prod_BrandModel Brand, Model
amProduct Prod_CatalogRef CatalogRef
amProduct Prod_!CategIdBrand !CategId, 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 = "texte"
```

```
$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

ここでは、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 4.1. 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。 1：AssetCenter Database Administratorでデータベースに接続する。 0：AssetCenter Database Administratorでデータベース記述ファイルを開く。	いいえ
Connection	データベースにアクセスするときに使うAssetCenterの接続名	はい
AppInfo	AssetCenterに関する情報	いいえ
AppVersion	AssetCenterのバージョン番号	いいえ
AppBuild	AssetCenterのビルド番号	いいえ
AppLanguage	AssetCenter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 4.2. 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。 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。 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 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
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指定します。 0：先頭でない 1：先頭である	いいえ

プロパティ名	説明	接続が必要
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。 0：末尾でない 1：末尾である	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。 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。 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 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。 0：いいえ 1：はい	いいえ

オブジェクト

表 4.3. 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

オブジェクト名	説明
RelevantScript as Script	妥当性のスクリプト
Base as Database	記述したデータベース

Fieldインスタンス プロパティ

表 4.4. 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。 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。 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。 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
タイプ	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

プロパティ名	説明	接続が必要
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
ReadOnly	データベースに接続するユーザのユーザ権限に関わらず、フィールドを変更できるかどうかを定義します。 このプロパティは、次のいずれかの値をとります。 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。 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。 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。 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（詳細画面の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

プロパティ名	説明	接続が必要
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。 0：いいえ 1：はい	いいえ
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。 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。 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 2：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。 0：いいえ 1：はい	いいえ

オブジェクト

表 4.5. 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	入力するデータベース
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

オブジェクト名

RelevantScript as Script

説明

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 4.6. Typeプロパティの値

保存値	表示値	意味
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	日付と時刻
9	Binary field	画像や式など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 4.7. UserTypeプロパティの値

保存値	表示値
0	デフォルト
1	数値
2	Boolean (Yes/No)
3	Monetary
4	日付
5	Date and time
6	Time
7	System itemized list
8	Custom itemized list
9	Feature value
10	パーセント

保存値	表示値
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
- 「テーブルまたはフィールドのSQL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 4.8. 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。 • 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。 • 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

プロパティ名	説明	接続が必要
Read	このプロパティは、次のいずれかの値をとります。 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。 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する。	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。 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
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。 0：いいえ 1：はい	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	フィールドの値	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

プロパティ名	説明	接続が必要
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

オブジェクト

表 4.9. 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	入力するデータベース
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 4.10. Typeプロパティの値

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

保存値	表示値
18	Owncopy

UserTypeプロパティの値

表 4.11. UserTypeプロパティの値

保存値	表示値
0	Normal
1	Comment
2	Image
3	History
4	Feature value

Indexインスタンス プロパティ

表 4.12. Indexインスタンスのプロパティ

プロパティ名	説明
Duplicates	インデックスで、NULLでない値を複数作成できるかどうかを示します。 1：インデックス内のフィールド一式が全く同じ値であるレコードを、複数作成することができます。 0：インデックス内のフィールド一式がある一定の値をとるレコードを、複数作成することはできません。
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。 インデックスで、NULL値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）。 1：NULLのインデックスを持つレコードを複数作成できます。 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。

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。
Label	ラベル（詳細画面に表示される通り）
Desc	説明
SqlName	SQL名
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。 0：いいえ 1：はい

オブジェクト

表 4.13. 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	入力するデータベース
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 4.14. 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」か「いいえ」にするか、またはどちらにするかをスクリプトで計算することを示します。このプロパティは、次のいずれかの値をとります。 0：いいえ 1：はい 2：スクリプト

プロパティ名	説明
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。 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
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 - 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 4.15. 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 4.16. 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。 0: いいえ 1: はい
Label	任意管理項目のラベル
Desc	説明
SQLName	SQL名
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。 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。 0: いいえ 1: はい

オブジェクト

表 4.17. 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。

プロパティ名

表 4.18. 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	フィールドの値
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)
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。 0: いいえ 1: はい

プロパティ名	説明
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。 0：ログインには、任意管理項目への書き込み権限がない。 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。 0：ログインには、任意管理項目の読取り権限がない。 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。 0：ログインに任意管理項目の更新権限がない。 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 4.19. 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
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 4.20. 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 0: いいえ 1: はい

オブジェクト

表 4.21. 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 4.22. 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。 0: いいえ 1: はい

オブジェクト

表 4.23. 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 4.24. 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。

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。 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
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。 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。 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	画面の詳細のタイトル

オブジェクト

表 4.25. 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 4.26. 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。 0: いいえ 1: はい
Label	ページのラベル
SQLName	説明 SQL名

Pageltemインスタンス

プロパティ

表 4.27. Pageltem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。 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。 0: いいえ 1: はい
FrameLabel	枠のラベル
SQLName	SQL名

プロパティ名	説明
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。 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。 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。 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。 0: いいえ 1: はい
BaseType	オブジェクトがフィールドの場合、フィールドのタイプを表します。

オブジェクト

表 4.28. 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
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 4.29. 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 4.30. 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 4.31. 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

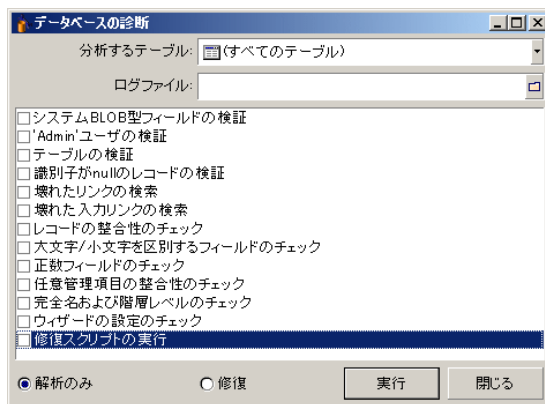
グローバル変数

表 4.32. グローバル変数

変数名	説明
Userlogin	データベースに接続するときのログイン
Time	AssetCenter Database Administratorを実行した時刻
Date	AssetCenter Database Administrator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

5 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AssetCenter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。



データベースの全体を解析または修復するためには、[(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。このファイル（デフォルトの拡張子は「.log」）を読むには、AssetCenterのログビューアを使用することをお勧めします（【スタート／プログラム／Peregrine／AssetCenter／ログビューア】メニュー）。

2つのオプションがあります。

- **【解析のみ】**：AssetCenter Database Administratorはデータベースの診断のみを実行し、問題点の修復は実行しません。
- **【修復】**：AssetCenter Database Administratorはデータベースを診断し、必要に応じて修復します。

警告:

修復方法は、【修復スクリプトの実行】オプションに左右されます。このオプションを選択しないと、自動修復のみが実行されます。このオプションを選択すると、自動修復後に修復ウィザードが起動します。

選択したオプションに応じて、AssetCenter Database Administratorは以下の要素を診断し修復します。

- データベース内に【sysblob】テーブル（データベースに統合されたシステムデータ）が存在するかどうか

注意:

【sysblob】テーブルの分析のみが可能です。

- **【部署と従業員】**テーブルに「Admin」ユーザ（データベース管理者）が存在するかどうか
- AssetCenter内に存在するすべてのテーブルの有無
- 識別子がNULLのレコードの有無（すべてのテーブル内）
- 壊れたリンク、または壊れた入力リンクの検索
- レコードの整合性のチェック

注意:

修復は、自動的に起動されるウィザードを介して行われます。

- 大文字/小文字を区別するフィールドのチェック（例：「User」と「user」の区別）
- 正数フィールドのチェック

注意:

修復は、自動的に起動されるウィザードを介して行われます。

- 任意管理項目の整合性のチェック

 注意:

修復は、任意管理項目の値に応じて（テキスト、数値、日付）、自動的に行われる場合と、ウィザードにより行われる場合があります。

- 完全名および階層レベルのチェック（これは特に、**【部署と従業員】**テーブルに關与します。）
 - ウィザードの設定の検証
-

 注意:

分析のみ可能です。

【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ことを表します。

6 データベース構造の更新

AssetCenter Database Administrator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。

AssetCenterデータベースのアップグレードに関する詳細は、マニュアル『**マイグレーション**』を参照してください。

7 | ODBCドライバを使ったデータベースへのアクセス

本章では、AssetCenter用に開発したODBCドライバを使ってAssetCenter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:

AssetCenter 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。

AssetCenter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

AssetCenter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
- 他のインストール済パッケージでドライバが必要な場合

ODBCドライバ名は、Peregrine AssetCenter Driverです。adbc32.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、AssetCenterの言語バージョンやご使用のDBMSに関係なく、すべてに共通して同一になっています。

ODBCドライバを使う場合

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AssetCenter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
AssetCenter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AssetCenter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、AssetCenter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。
AssetCenter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AssetCenter接続を使わずに直接データベースにアクセスします。

	ODBCドライバを使う	ODBCドライバを使わない
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DBMS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、データベースのDBMS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
- ODBCドライバによって次の情報を参照できます。
- 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名で表されます。



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**AssetCenter 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AssetCenterデータベースにアクセスできます。

- 標準の**AssetCenter Databases**接続
- 独自に作成した接続

標準のAssetCenter 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AssetCenter接続は、レポートの作成時と使用時に選択します。AssetCenter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 "Peregrine AssetCenter 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Crystal Reportsのレポートを作成する

- Crystal Reports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。
- **AssetCenter Databases** ODBC接続を選択します。
- 標準のAssetCenter接続ダイアログボックスが表示されます。
- 適切なAssetCenter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 データベースのカスタマイズ

AssetCenter Database Administratorでは様々なタイプのカスタマイズを実行できます。

- 既存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、画面など）
- 新規オブジェクトの作成によるカスタマイズ

既存のオブジェクトをカスタマイズする

AssetCenter Database Administratorでは、データベースの既存のオブジェクトの一部のみをカスタマイズできます。AssetCenter使用時の問題発生を避けるために、一定の値とオブジェクトは読取り専用です。また、既存の詳細ページも変更不可能です。

データベースのオブジェクトをカスタマイズするためには、データベース記述ファイル（「gbbase.xml」）を編集します

AssetCenter Database Administratorでデータベースをカスタマイズするには次の2通りの方法があります。

- データベースを作成する前にカスタマイズする。
- データベースを作成した後にカスタマイズする。



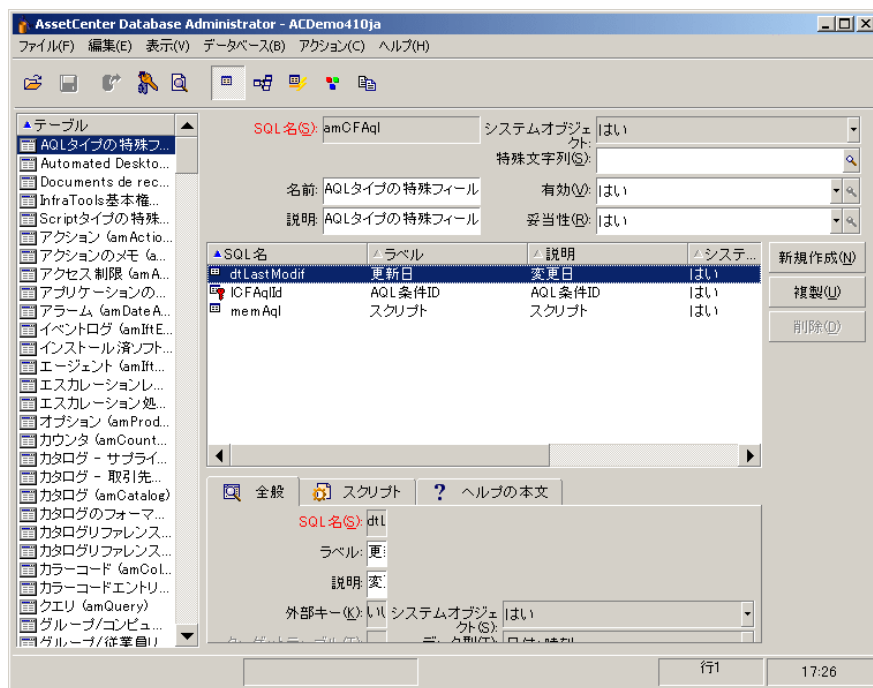
注意:

ただし、テキストフィールドのサイズだけは、データベースを作成する前でないとは変更できません。

上記のそれぞれの方法について、次の2段階のカスタマイズが可能です。

- テーブルのカスタマイズ
- テーブルのオブジェクト（フィールド、リンク、インデックス、画面、ページ）のカスタマイズ

図 8.1. AssetCenter Database Administrator - メイン画面



テーブルのカスタマイズ

テーブルをカスタマイズする場合は、次の変更が可能です。

- **【説明】** フィールド
- **【名前】** フィールド。AssetCenterで表示するテーブル名です。

- **【特殊文字列】** フィールド。AssetCenterでこのテーブルのレコードを表示する文字列を構成できます。
- **【有効】** フィールド。データベース内の特定のテーブルの全レコードに適用されます。選択しているテーブル内のレコードを作成または変更できるかどうかを定義できます。
 - このフィールドを **【はい】** に設定すると、そのテーブルで常時レコードを作成または変更できます。
 - このフィールドを **【いいえ】** に設定すると、そのテーブルでレコードを作成または変更できません。
 - このフィールドを **【スクリプト】** に設定すると、そのテーブルでレコードを作成または変更できる条件を指定するスクリプトを定義できます。

例えば、**【数値】** 型の任意管理項目の値が特定の限界値を超えた場合は、レコードを作成しないように制限することができます。次のスクリプトは、**[amFeature]** テーブルの **【数値】** フィールドを定義した例です。

```
if [seDataType] = 1 and [fMin] > [fMax] Then
  Err.Raise(-1, " 最小 フィールドに 最大 フィールドより大きい
  値を入力することはできません。")
  RetVal = FALSE
Else
  RetVal = TRUE
End If
```

注意:

【有効】 フィールドで、選択しているテーブルでのレコードの作成または変更を無効に設定する場合は、標準のBasic関数のErr.Raiseを使ってユーザの画面にエラーメッセージが表示されるように定義してください。エラーメッセージにより、ユーザにレコードを変更または作成できない理由を知らせることができます。

【SQL名】 フィールドは変更できません。このフィールドはテーブルのSQL名を示します。

- **【任意管理項目を含むことができる】** オプションを選択すると、テーブルに任意管理項目を含めることができます。
この操作は、取り消し不可能です。

オブジェクトのカスタマイズ

2番目のカスタマイズ枠では、選択したタイプのオブジェクトがすべて一覧表示されます。

リスト内のオブジェクトを選択すると、第3のカスタマイズ枠にそのオブジェクトの詳細情報が表示されます。

注意:

赤の  アイコンは、このテーブルの主キーを表します。

黄色の  アイコンは、外部キーを表します。

 アイコンは、インデックスに属するフィールドを表します。

 アイコンは、同じインデックスが付いたフィールド一式が同じテーブル内に複数存在できないことを表します。

 アイコンは、NULL値を除き、同じインデックスが付いたフィールド一式が同じテーブル内に複数存在できないことを示します。NULL値の場合は、何回でも複製できます。

 アイコンは、1リンクを表します。

 アイコンは、nリンクを表します。

フィールド、リンクとインデックスのカスタマイズ

このタイプのオブジェクトでは、次の変更が可能です。

【全般】 タブページでは、次の情報を指定できます。

- 【名前】 フィールド
- 【説明】 フィールド
- 【サイズ】 フィールド：テキスト型のフィールドのサイズを指定します。このフィールドは、データベースを作成する前にカスタマイズする場合、またはデータベース記述ファイルを変更する場合にのみ編集できます。最大サイズは255文字（半角の場合）です。

【スクリプト】 タブページでは、次の情報を指定できます。

注意:

次の属性は、指定した場合を除き、ユーザインタフェースを介してまたは外部ツールを使ってデータベースにアクセスする場合にも有効です。

- 【履歴保持】 フィールド：フィールドの変更履歴を保持するかどうかと、適当な条件（スクリプトによる）を指定します。

- **【読取り専用】フィールド**：ユーザインタフェースを介してフィールドを変更できるかどうかと、読取り条件（スクリプトによる）を指定します。

 注意:

外部ツールを使ってデータをインポートする場合は、この属性は適用されません。また、マッピング時にも読取り専用以外の属性は適用されません。一方、インポートスクリプトが、ソースフィールドを読取り専用のターゲットフィールドにマップした場合は、読取り専用フィールドも変更されます。

- **【無関連】フィールド**では、選択されたレコードの詳細画面での表示方法を設定します。
 - **【はい】**：要素を表示しません。
 - **【いいえ】**：要素を表示します。
 - **【スクリプト】**：表示条件を指定するスクリプトが、要素に関連付けられていることを意味します。

```

If [bDepartment] = 1 Then
  RetVal = 1
Else
  RetVal = 0
End If

```

選択されている要素が無関係の場合は表示されません。無関係でない場合（つまり関係ある場合）は表示されます。

- **【フィルタ】フィールド**は、このバージョンのAssetCenterでは使用されません。
- **【書式】フィールド**：フィールド値をデータベースに保存する前に自動的に書式化できます。

テキストフィールドには次の書式があります。

- **【標準】**：入力した通りに保存します。
- **【大文字】**：大文字に変換してから保存します。
- **【小文字】**：小文字に変換してから保存します。
- **【自動】**：先頭の文字を大文字に変換してから保存します。

数値フィールドには次の書式があります。

- **【標準】**：正と負のすべての値を指定できます。
- **【正】**：負の値を指定できません。警告メッセージが表示されます。

 注意:

この属性を変更しても、データベース内の既存の値は変換されません。

- **【必須】** フィールドでは、必須フィールドにする条件を定義できます。

 注意:

フィールドを必須にすると、常に表示されるわけではない場合（他のフィールドの値に応じて表示される場合など）に問題が発生することがあります。フィールドを設定する場合またはスクリプトを記述する場合は、この点に注意してください。

- **【デフォルト】** フィールドでは、フィールドのデフォルト値を指定します。AssetCenterで新しいレコードを作成すると、ここで指定した値が自動的にフィールドに入力されます。デフォルト値はBasicスクリプトを使って定義します。

 注意:

特殊フィールドは、フィールドの値が **【特殊文字列】** タイプまたは **【BASICスクリプト】** タイプの標準フィールドのデフォルト値を計算する場合にのみ使用できます。

 注意:

デフォルト値を **【ウィンドウ／リストのみ】** テーブル（SQL名：amComment）にリンクするように指定することはできません。

【ヘルプの本文】 タブページでは、データベースの各オブジェクト固有のヘルプをカスタマイズできます。AssetCenterでこのヘルプを表示するには、**【Shift+F1】** キー（または **【ヘルプ／フィールドのヘルプ】** ポップアップメニュー）を使います。このヘルプの本文は、デフォルトで「説明」、「例」、「注」の3つのセクションから構成されます。これらのセクションのタイトルは、**【フィールドのヘルプ】** のテーブル（SQL名：amHelp）で、SQL名が「Comment」、「Sample」、「Warning」のリンクのラベル（名前）を変更すればカスタマイズできます。

 注意:

カスタマイズ画面の下部に表示されるその他のフィールドは、情報の提供のみを目的としており、カスタマイズはできません。この画面で実行できるオブジェクトの設定は、AssetCenterの **【オブジェクトの設定】** ポップアップメニューを使って同じように実行できます。

詳細のカスタマイズ

このタイプのオブジェクトでは、次の変更が可能です。

【全般】 タブページでは、次の情報を指定できます。

- 【名前】 フィールド
- 【リスト／詳細】 タブページでは、次の情報を指定できます。
- 【リスト/詳細の割合】
- 【詳細のタイトル】
- 【リストのタイトル】
- 【システムフィルタ】
- 【機能ドメイン】
- 【リストの列】

ボタン

【ボタン】 タブでは、画面上のボタンの作成、変更、削除、整理を実行できます。

ボタンは以下の任意のオブジェクトタイプに関連付けることができます。

- アクション
- 画面
- ビュー
- 書式
- レポート

ボタンを作成するには:

- 1  ボタンをクリックします。
- 2 タブ内のボタンのリストに行が追加されます。行内の各セルをクリックし、ボタンの次のプロパティを定義します。
 - 名前：ボタンの内部名。ボタンを固有の名前で識別できるようになります。
 - テキスト：ボタン上に表示されるテキスト
 - 説明：AssetCenterの画面上に表示されるボタンのタイトル
 - 複数のレコードの選択：アクションが複数のレコード上に実行されるかどうかを指定します。
 - 選択なし：レコードを選択せずにアクションを実行できるかどうかを指定します。
 - 関連アクション：ボタンをクリックすると実行されるアクションを定義します。このフィールドのシンタックスは以下のとおりです。

`<アクションタイプ>:<アクションまたはビューなどのSQL名...>`

このシンタックスでは、アクションタイプ用に以下の値を使用できます。

- A：アクション
- S：画面
- V：ビュー
- F：書式
- R：レポート
- 妥当性：ボタンが表示／非表示であるか、またはスクリプトが表示を条件付けるかどうかを指定します。

ボタンを変更するには

- 1 ボタンに対応する行を選択します。
- 2 各セルをクリックしてボタンのプロパティを定義します。

ボタンが標準ボタンの場合、主機能が引き続き使用可能です（例えば、【複製】ボタンは常にレコードを複製します）。ただし、以下のボタンは例外です：、および。これらのボタンをオブジェクトに関連付けることは可能ですが、主機能は使用できなくなります。

関連付けられたオブジェクトは以下のようにトリガされます：

- 【新規作成】 および 【複製】 ボタンの主機能の前
- 【作成】、【変更】 および 【作成／複製】 ボタンの主機能の後

関連付けられたオブジェクトが正しく実行されなかった場合、オブジェクトの主機能がキャンセルされる場合があります。

ボタンを削除するには

- 1 【ボタン】 タブ内のリストからボタンを選択します。
- 2  をクリックします。
- 3 【変更】 をクリックします。

ボタンの順序を変更するには

詳細画面に表示されるボタンの順序を変更するには、 ボタンと  ボタンを使用します。

新規オブジェクトの作成

AssetCenter Database Administratorではデータベース用に新規オブジェクトを作成できます。

**警告:**

空のデータベースを使用するようお勧めします。本番データベースを変更する場合は、変更事項を保存した後にデータベースの検証と修復を実行します（**【アクション／診断／データベースの修復】**メニュー）。

特殊フィールドの作成方法

以下の手順は新規オブジェクトの作成方法の1例です。ここでは新規テーブルの作成を例に取ります。各手順は各節ごとに説明されています。

- テーブルを作成する
- テーブルのフィールド、リンクとインデックスを作成する
- テーブルの詳細を作成する
- 必要に応じて、詳細のアクション用ボタンを作成する
- 詳細のページを作成する
- ページを詳細に追加する
- 変更を保存する
- またはカスタマイズを伝達し、適用する

テーブルの作成

新規テーブルを作成するには、

- **【データベース／テーブルの追加】** メニューを選択します。
- テーブル作成用のウィンドウが表示されます。
- このウィンドウでテーブルに関連する一般的なフィールドに値を入力します。
 - **【SQL名】** フィールドに、新規テーブルを識別するための固有の名前を指定します。この名前はBasicスクリプト内でも参照されます。
 - **【ラベル】** フィールドに、AssetCenterで表示されるテーブル名を入力します。
 - **【説明】** フィールド
 - **【主キー】** フィールドに、テーブルの主キーとして使用されるフィールドのSQL名を入力します。
- 任意管理項目を新規テーブルに関連付ける場合は、**【任意管理項目の追加】** オプションを選択します。AssetCenter Database Administratorは、任意管理項目に必要なテーブルを自動的に追加作成します。
- **【作成】** をクリックすると、AssetCenterはテーブルと主キー用のフィールドを作成します。この時点でテーブルのフィールドを編集できます。

フィールド、リンクとインデックスの作成

フィールドを作成するには、

- 1 **【データベース／フィールドの追加】** メニューを選択します。
 - 2 フィールド作成用のウィンドウが表示されます。
 - 3 このウィンドウで以下のフィールドに値を入力します。
 - ・ **【SQL名】** フィールド
 - ・ **【ラベル】** フィールド
 - ・ **【説明】** フィールド
 - ・ **【入力タイプ】** フィールド
 - 4 インデックスを作成する場合は、**【このフィールドのインデックスを作成】** オプションを選択します。
- フィールドを作成した後は、フィールドのサイズまたは形式のみを変更できます。

注意:

フィールドの入力タイプに関する注意事項

- ・ **【カスタムリストデータ】** タイプのフィールドは、以下のシンタックスに従う必要があります。

自由入力のテキスト

入力されたテキスト（**【フォーマット】** フィールド）は、リストデータの詳細の**【識別子】** フィールドに対応します。

- ・ **【システムリストデータ】** タイプのフィールドは、以下のシンタックスに従う必要があります。

未承認 0 承認済 1 却下 2

- ・ **【期間】** タイプのフィールドは、以下のシンタックスに従う必要があります。

%Yl%Ml%Dld%H%N%S

- ・ **Y、M、D、H、N、S**は、それぞれ年、月、日、時間、分、秒を定義します。
- ・ **l** (long) は、期間の単位を完全表示するかどうかを指定します。
- ・ **dl**は、期間をデフォルト値として定義します。

リンクを作成するには、

- 1 **【データベース／リンクの追加】** メニューを選択します。
- 2 リンクの作成用のウィンドウが表示されます。
- 3 このウィンドウで以下のフィールドに値を入力します。

- **【リンクタイプ】** フィールド
 - ソーステーブル用には、以下のフィールドに値を入力します。
 - **【SQL名】** フィールド
 - **【ラベル】** フィールド
 - **【説明】** フィールド
 - ターゲットテーブル用には、以下のフィールドに値を入力します。
 - **【テーブル】** フィールド。ターゲットフィールドを指定します。
 - **【SQL名】** フィールド
 - **【ラベル】** フィールド
 - **【説明】** フィールド
- 4 インデックスを作成する場合は、**【このリンクのインデックスを作成】** オプションを選択します。
- ▶ **リンクについて** [参考文献 32]



注意:

リンクのタイプに関する注意事項

- リンク「1-N」のタイプには、Normal、Define、Own、OwnCopy、およびOwnがあります。
- リンク「1-1」のタイプには、Normal、OwnCopy1、Copy1、Own1、およびNeutDef1があります。
- リンク「N-N」のタイプには、RelNN、NNCopy、およびCopyNNがあります。

インデックスを作成するには、

- 1 **【データベース／インデックスの追加】** メニューを選択します。
- 2 各オブジェクトのプロパティに値を入力します。プロパティに関しては「**オブジェクトのカスタマイズ**」の節を参照してください。

階層テーブルを作成する

階層テーブルを作成するとは、同テーブル内に親リンクを作成することを意味します。

あるテーブル内に親リンクが作成されると、同テーブル内に子リンクが存在することになります。

親リンクを作成するには、

- 1 **【データベース／親リンクの追加】** メニューを選択します。
- 2 親リンクの作成用のウィンドウが表示されます。
- 3 このウィンドウで以下のフィールドに値を入力します。
 - **【SQL名】** フィールド

- **【ラベル】** フィールド
 - **【説明】** フィールド
- 4 以下のシンタックスに従って、**【階層】** フィールドに値を入力します。

<親リンクのID>,<階層用に使用されるフィールドの名前>

例えば、**【製品】** テーブルを例に取り上げると、親リンクの **【階層】** フィールドには、以下のように値が入力されているはずです。

[ParentId,InternalRef]

上記の操作が実行されると、**FullName**フィールドが作成されます。このフィールドは、テーブル内の固有の識別子の1つです。

詳細の作成

詳細には、テーブルのオブジェクト内に格納された情報がグラフィカルに表示されます。AssetCenterでメニューやビューを選択したり、ツールバーのアイコンをクリックしたりする時に表示される画面は、詳細の例にあたります。詳細には複数のページがあり、AssetCenterのインタフェースではタブとして表示されています。

注意:

AssetCenterに付属している標準の詳細は、アプリケーションの機能をサポートしています。これらの詳細を編集または変更することはできません。既存の詳細にページを追加する場合は、既存の詳細を複製し、名前を付けてから必要な要素を追加します。

詳細を作成するには、

- 1 左側の枠内で、新規詳細の作成先のテーブルを選択します。
- 2 **【表示／詳細】** メニューを選択するか、またはツールバーの  をクリックします。
- 3 **【データベース／詳細の追加】** メニューを選択するか、または **【新規作成】** ボタンをクリックします。
- 4 **【全般】** タブの以下のフィールドに値を入力します。
 - **【SQL名】**：詳細のSQL名。この名前により、詳細の識別が可能になり、詳細をスクリプトやクエリ内で参照できるようになります。
 - **【ラベル】**：詳細のラベル
- 5 **【リスト／詳細】** タブの以下のフィールドに値を入力します。
 - **【リストのタイトル】**：画面のリスト内（AssetCenterの **【管理／画面一覧】**）に表示される詳細のタイトルを指定します。

- ・ **【詳細のタイトル】**：AssetCenterに表示されるウィンドウのタイトルを指定します。記述文字列がこのタイトルに追加されます。
- ・ **【ドメイン】**：詳細の機能ドメイン。入力する機能ドメインがデータベース内に存在する場合（AssetCenterの**【管理／機能ドメイン】**メニューで作成できます）、自動的にリンクが作成され、機能ドメイン枠内から画面に直接アクセスできるようになります。
- ・ **【リスト／詳細の割合】**：詳細とリスト間の割合。
- ・ **【リストの列】**：このフィールドでは、リストに表示される列を定義できます。シンタックスは以下のとおりです。

<列用のフィールドのSQL名1>,<列の幅>,...

6 **【作成】** をクリックします。

これで画面の空のシェルを作成したことになります。この画面にフィールド、リンクやアクション用ボタンを含んだページを追加します。

アクション用ボタンの作成

AssetCenter Database Administratorでは、詳細用のボタンを作成することができます。ボタンはアクションの実行、画面の表示、レポートや書式の印刷、ビューの表示、などをトリガします。

ボタンを作成するには、

- 1 ボタンの追加先の詳細を選択します。
- 2 詳細の **【ボタン】** タブを選択します。
- 3  をクリックします。タブ内のボタンのリストに、新規要素が追加されます。
- 4 各プロパティをクリックして値を直接入力します。
 - ・ **名前**：ボタンのSQL名。この名前により、ボタンの識別が可能になり、詳細をBasicスクリプトやクエリ内で参照できるようになります。
 - ・ **テキスト**：ボタンのテキストを入力します。
 - ・ **説明**：AssetCenterの画面上に表示されるボタンのタイトル。ボタン上にマウスポインタを置くと表示されます。
 - ・ **複数のレコードの選択**：アクションが複数のレコード上に実行されるかどうかを指定します。
 - ・ **編集モード**：レコードの編集集中にボタンを表示するかどうかを指定します。
 - ・ **選択なし**：レコードを選択せずにアクションを実行できるかどうかを指定します。
 - ・ **関連アクション**：ボタンをクリックすると実行されるアクションを定義します。このフィールドのシンタックスは以下のとおりです。

<アクションタイプ>:<アクションまたはビューなどのSQL名...>

このシンタックスでは、アクションタイプ用に以下の値を使用できます。

- A：アクション
 - S：画面
 - V：ビュー
 - F：書式
 - R：レポート
- 妥当性：ボタンの表示パラメータを指定します（「はい」、「いいえ」、または「スクリプト」）。
- 5 **【変更】** をクリックして変更事項を確定します。

ページの作成

AssetCenterのインタフェースでは、ページはフィールドを含むタブとして表示されています。ページを作成するには、

- 1 左側の枠内で、新規ページの作成先のテーブルを選択します。
- 2 **【表示／ページ】** メニューを選択するか、またはツールバーの  をクリックします。
- 3 **【データベース／ページの追加】** メニューを選択するか、または **【新規作成】** ボタンをクリックします。
- 4 **【全般】** タブの以下のフィールドに値を入力します。
 - **【SQL名】**：ページのSQL名。この名前により、ページの識別が可能になり、詳細をスクリプトやクエリ内で参照できるようになります。
 - **【ラベル】**：ページの名前。これはAssetCenterのインタフェース内でタブ名として表示されます。
- 5 **【作成】** をクリックします。
- 6 詳細の **【内容】** タブをクリックします。作成するページ内で使用可能であるフィールドを選択します。以下の方法で選択します。
 - 左側のリスト（**フィールドとリンク**）内でページに追加するフィールドまたはリンクを選択し、 をクリックして右側のリスト（**フィールドのリスト**）に移します。左側のリストでフィールドをダブルクリックすると、上記と同様の操作を実行できます。
 - ページからフィールドを削除する場合は、右側のリストでフィールドを選択し、 をクリックします。右側のリストでフィールドをダブルクリックすると、同様の操作を実行できます。
 - ページのレイアウトやフィールドの配置場所などは、AssetCenterが自動的に計算します。フィールドの表示順を指定する場合は、左側のリ

ストで 、、 と  ボタンを使って、表示する順番にフィールドを並べ替えます。

7 **【変更】** をクリックします。

ページのレイアウト

ユーザがページに追加するオブジェクト（フィールドやリンク）は、デフォルトで1列上に順番に表示されます。しかし、このデフォルト表示方法を変更するために縦や横の区切り線を定義することも可能です。区切り線は以下の内部変数で定義されます。

- **HSplitControl**：横の区切り線
- **VSplitControl**：縦の区切り線

区切り線を追加するには、

- 1 詳細画面の **【内容】** タブページに移動します。
- 2 **【フィールドとリンク】** リストの上にあるフィールドに、上記の2つの内部変数のうちの1つを入力します。
- 3  ボタンをクリックします。
- 4 区切り線は右のフィールドのリスト内へ移されます。区切り線を表示する順番は、他のオブジェクトの順番を変更するのと同じ方法で指定できます（、、 と  ボタンを使います）。

以下の規則がページの表示に適用されます。

- 2つの**HSplitControl**変数間にあるオブジェクトは全て同じ横のブロックの一部を成します（1行または複数行）。
- **VSplitControl**変数は、横のブロック内のオブジェクトを列に分割します。

例えば、AssetCenterの従業員の **【全般】** タブページの表示方法は、以下のように定義されます。

```
HSplitControl_0
HSplitControl_1_Adresse
HSplitControl_2
Location
HSplitControl_2
VSplitControl_2
Phone
MobilePhone
VSplitControl_2
Fax
HomePhone
HSplitControl_2
EMail
Field1
```

```
Field2
HSplitControl_1
HSplitControl_0
VSplitControl_0
IDNo
dHire
dLeave
VSplitControl_0
BarCode
Field3
HSplitControl_0
MultilineEdit-Comment
```

アンダスコア（下線）は、変数間の帰属関係を定義できます。

詳細にページを追加する

ページを詳細に追加するには、

- 1 ページの追加先の詳細を選択します。
- 2 詳細の【ページ】タブを選択します。ここで、詳細に含まれるページを選択します。
 - 左側のリスト（使用可能ページ）内で詳細に追加するページを選択し、をクリックして右側のリスト（詳細ページ）に移します。右側のリストでページをダブルクリックすると、上記と同様の操作を実行できます。

ページのリンクを選択することも可能です。これは、別のテーブルにリンクしているページを表示することを意味します。ページリンクを選択するには、【使用可能ページ】の上にあるドロップダウンリストからリンクを選択します。このリストには使用可能な1対1のリンク全てが表示されます。

 - 詳細からページを削除する場合は、右側のリストでページを選択し、をクリックします。左側のリストでページをダブルクリックすると、同様の操作を実行できます。
 - タブの表示順を指定する場合は、右側のリストで、、とボタンを使って、表示する順番にタブを並べ替えます。リスト内の最初のページは最初に表示されるタブで、最後のページは最後に表示されるタブです。

サポートする他の言語へのカスタマイズの適用

データベースに追加されたオブジェクトは、変更時にAssetCenter Database Administratorで選択されていた言語で使用できます。

以下のオブジェクトは、サポートする言語のそれぞれに対して再作成する必要があります。

- 詳細
- ページ

ヒント:

このためには、サポートする言語を1つずつ選択し（AssetCenter Database Administratorのツールバーの右側のリストを使用）、最初の言語のときと同じ方法でオブジェクトを作成します。

以下のオブジェクトは全言語向けに作成されますが、ラベルと説明は翻訳する必要があります。

- テーブル
- フィールド
- リンク
- インデックス

特定の言語で追加されたオブジェクトのラベルと説明を翻訳するには:

- 1 追加でサポートする言語を1つずつ選択します（リストはAssetCenter Database Administratorのツールバーの右側にあります）。
- 2 **【アクション／説明を更新】**を選択します。
- 3 各表示オブジェクトのラベルと説明を翻訳します。
- 4 **【変更】**をクリックして変更を確認します。
- 5 更新ウィンドウを閉じます。

変更事項の保存

変更事項を保存するには、

- 1 **【ファイル／データベース構造の保存】**を選択するか、またはをクリックします。
- 2 ウィザード画面に表示される指示に従います。

変更を確認する

変更事項を確認するには、AssetCenterを起動し、カスタマイズしたデータベースに接続する必要があります。上記の例では新規テーブルを作成したので、このテーブルの詳細にアクセスします。

- 1 **【管理／画面一覧】**を選択します。
- 2 画面に表示されるリストから詳細を検索し、**【OK】**をクリックすると、詳細が表示されます。
- 3 ビューを作成すると便利です。**【ツール／ビュー／現在のウィンドウから作成】**を選択します。
- 4 **【名前】**フィールドにビューの名前を入力します。この名前は**【機能】**の枠内に表示されます。
- 5 **【ドメイン】**で機能ドメインを選択します。ビューの名前は、機能の枠内のこの機能ドメイン下に表示されます。
- 6 これで新規の詳細に直接アクセスできるようになりました。

重要

AssetCenterでは、全オブジェクト（特にアクション、ビューと画面）がSQL名で識別されています。オブジェクトをメニューまたは機能の枠で開く際に、AssetCenterはSQL名を使用します。複数のオブジェクト、例えばビューと画面などに同じSQL名が付いている場合、AssetCenterはオブジェクトを以下の順番で開きます。

- ビュー
- 画面

例えば、**【部署と従業員】**（SQL名：amEmplDept）テーブルに新規の詳細を作成し、「amEmplDept」というSQL名のビューをこの詳細に関連付けるとします。**【ポートフォリオ／部署と従業員】**メニューを選択する場合、または機能の枠内で**【従業員のリスト】**をクリックする場合、表示されるのは新規の詳細です。

テスト用データベースで用意されたカスタマイズ

AssetCenterデータベースをカスタマイズする場合、元のデータベースの整合性を保持するために、データベースのコピーを使用するようお勧めします。

本章では、同一のバージョンのAssetCenterで、テスト用データベースを本番データベースに転送する方法を説明します。

以下の操作を行う必要があります。

- 1 本番データベースのコピーを作成する。このコピーをテスト用データベースとして使用します。
 - 2 テスト用データベースを用いて、開発およびテストを実行する。
このテスト用データベースには、新しいデータが含まれます。新しいデータを分類し、本番データベースに転送します。
 - 3 テスト用データベースからプロダクション用データベースにカスタマイズを伝達します
- テスト用データベースからプロダクション用データベースにカスタマイズを伝達する代わりに、以下の方法もあります。
- 1 補足データとテスト用データベースの構造を出力する。
 - 2 補足データとテスト用データベースの構造を本番データベースにインポートする。
 - 3 Connect-Itを用いてテスト用データベースのデータ（レコード）を本番データベースに移行する。

本番データベースのコピーを作成する

！ 警告:

本番データベースのコピーを作成する方法は、お使いのデータベースエンジンに応じて異なります。データベースエンジンの付属マニュアルを参照されるようお勧めします。

空のデータベースにコピーを作成するには、以下の手順に従います。

- 1 SGBDで空のシェルを作成します。
 - ▶ DBMSを使って空のAssetCenterデータベースを作成する [修献 48].
 - 2 DBMSログインを作成します。
 - ▶ DBMSログインを作成する [修献 59].
 - 3 空のシェルへの接続を作成します。
 - ▶ 空のシェルへの接続を指定する [修献 61].
 - 4 AssetCenter Database Administratorを起動します。
 - 5 本番データベースに接続します（【ファイル／開く／既存のデータベースを開く】メニュー）。
 - 6 本番データベースの内容をコピーします（【アクション／データベースを空のデータベースにコピー】メニュー）。
- 作成した空のシェルに対する接続を選択します。

**注意:**

この操作では、データベース全体（構造およびデータ）がコピーされます。

▶ DBMSを変更する [修献 70]

テスト用データベースからプロダクション用データベースへのカスタマイズの伝達

プロダクション用データベースをカスタマイズするには2つの方法があります。

- データベースの構造を記述している「gbbase*.*」ファイルを変更し、その変更をプロダクション用データベースにインポートする。
- 空のテスト用データベースを変更し、その変更をプロダクション用データベースに伝達する。

これらのカスタマイズは、テーブル、フィールド、リンク、画面、インデックスに関連します。

プロダクション用データベースにカスタマイズを伝達するには、AssetCenter Database Administratorを使います。

カスタマイズの伝達

プロダクション用データベースにカスタマイズを伝達するには:

- 1 プロダクション用データベースに接続します。
- 2 **【アクション／カスタマイズの統合】**を選択します。
- 3 ウィザード画面に表示される指示に従います。
- 4 統合が終了したら、テスト用データベース（または「gbbase*.*」ファイル）に対して行ったカスタマイズがプロダクション用データベースに正しく伝達されていることを確認します。

カスタマイズ伝達のロジック

テスト用データベースに対して行われたカスタマイズは、プロダクション用データベースに対する変更を上書きします。

プロダクション用データベースに対する変更を保持しながらテスト用データベースに対するカスタマイズを統合したい場合は、両方のデータベースの構造を出力し、それらを比較して、統合プログラムまたはConnect-Itシナリオを使って統合することをお勧めします。

データベース構造を出力する

AssetCenter Database Administratorを使用すると、テスト用データベースを出力できます。

データベースの構造を出力するには、以下の手順に従います。

- 1 データベース構造の出力先のフォルダを作成します。
- 2 AssetCenter Database Administratorを起動します。
- 3 本番データベースに接続します（[ファイル/開く/既存のデータベースを開く] メニュー）。
- 4 データベースの構造を出力します（[ファイル/データベース構造の出力] メニュー）

データベースの構造を出力することは、以下のタイプのファイルを出力することに当たります。

- .bin
- .cfg
- .xml
- .dsd
- .str
- .stt
- .usr
- .wiz

新しいデータベース構造をインポートする

！ 警告:

一旦インポートを実行すると、取り消すことはできません。テスト用データベースおよび本番データベースのバックアップコピーを、作成しておくようお願いします。

テスト用データベースの構造を本番データベースにインポートすると、以下のことが可能になります。

- テスト用データベースの構造に基づいて、本番データベースを更新する。
- テスト用データベースの補足データに基づいて、本番データベースを更新する。

テスト用データベースの構造をインポートするには、以下の手順に従います。

- 1 AssetCenter Database Administratorを起動します。

- 2 本番データベースに接続します（[ファイル／開く／既存のデータベースを開く]メニュー）。
- 3 構造の更新を選択します([アクション／カスタマイズの統合])。
- 4 ウィザード画面に表示される指示に従います。

Connect-Itを使ってデータを移行する

移行するデータのタイプに対応するConnect-Itシナリオを起動します。この際、変更されたデータと元のデータを区別するために、WHERE句を含めます。

ソフトウェアカウンタに注意

【カウンタの作成】 ウィザードで資格カウンタを作成する場合、AssetCenterは一部のレコードを主キーで識別するクエリを作成します。

カウンタの**ユーザ権限範囲**（LicQuery）に関連するクエリの例:

```
(bDepartment = 0) AND (lEmplDeptId = 164)
```

テスト用データベースでカウンタを作成したあと、プロダクション用データベースに転送（エクスポートしてインポート）した場合、クエリ中の識別子はおそらく間違ったレコードを参照しています。

例えば、識別子「164」は、テスト用データベースでは「Mr. Doe」だが、プロダクション用データベースでは「Mrs Dour」かもしれません。

ソフトウェアカウンタとそのクエリをプロダクション用データベースにインポートしたあとは、カウンタに関連するクエリを確認する必要があります。



ヒント:

もう1つの方法としては、クエリでレコードを識別する主キーとしてもっと安定したフィールドを使用します。例えば、名前やコードなどです。

複数のオペレータによって同時に行なわれたカスタマイズ

警告:

ここで説明するプロセスを実行するには、以下の領域の詳細や知識に留意する必要があります。

- AssetCenter データベースの構造
- データの出力
- データのインポート
- データベース管理
- AssetCenterの管理

プロダクション用データベースのデータの整合性を保とうとする場合、時間のかかる困難な作業となる可能性もあります。

問題の説明

ここでは、AssetCenterデータベースをチームとしてカスタマイズする方法について説明します。

ファイルに保存したカスタマイズを共有できないため、これには制限があります。

このため、各開発者は以下の作業を行う必要があります。

- プロダクション用データベースのコピー（開発用データベースと呼びます）で作業を行います。
- 独自のカスタマイズは、自分のファイルに別個に保存します。

次に以下の操作を実行することができます。

- 1 個別のカスタマイズファイルをマージします。
- 2 マージしたカスタマイズをプロダクション用データベースに伝達します。

可能なカスタマイズの種類

AssetCenterデータベースは、さまざまなレベルでカスタマイズできます。

- データベースの構造パラメータ
- データベースのデータ

データベースのカスタマイズ可能な構造パラメータ

- テーブル（ラベル、説明など）
 - フィールド（ラベル、説明、サイズなど）
 - リンク（ラベル、説明、デフォルト値など）
 - インデックス（ラベル、説明、フィールドリストなど）
 - ページ（ラベル、説明、内容など）
 - 画面（ラベル、説明、ボタンなど）
 - フィールドとリンクのコンテキストヘルプ
-

注意:

これらのパラメータでスクリプトが使用されることがあります（フィールドの必須ステータスなど）。

データベースのカスタマイズ可能なデータ

- 任意管理項目および任意管理項目クラス
- 特殊フィールド
- リストデータ
- 単位
- 機能ドメイン
- アクション
- ワークフローチャート
- スクリプト
- クエリ
- レポート
- 書式
- ビュー
- ユーザ権限およびプロファイル、アクセス制限
- 機能権限
- データベースオプションおよびワンポイント
- 設定およびオプション
- ワークフローおよび配備サーバ

**注意:**

カスタマイズするデータは、ニーズに合わせて選択します。

選択を行ったら、AssetCenter ユーザがこのデータに対応するテーブルを許可なく変更できないようにします。

このためにはユーザプロファイルでユーザを判断します。

例えば、ワークフローにリンクされたテーブルを変更するのをやめ、リストデータの値の変更だけが続けることができます。

アクセス制限を使用すると、可能な設定の範囲を定義する際により明確な指定が行なえます。

必要に応じて、対象のテーブルの識別フィールドに接頭コードを使用すると（amWfScheme.Refなど）、設定で対象となるレコードの識別を簡単に行うことができます。

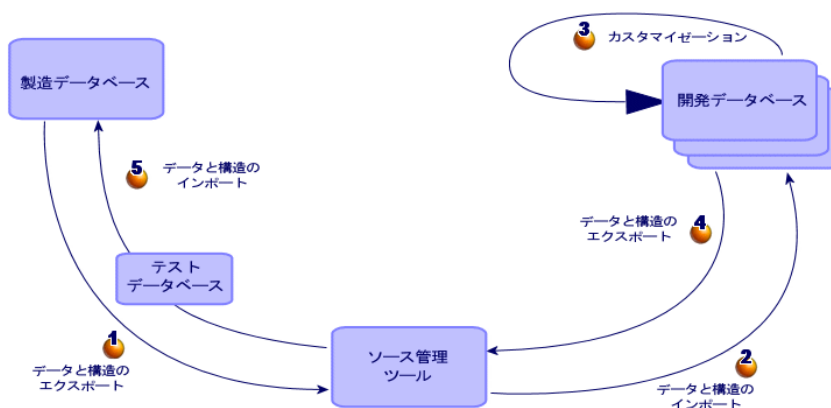
リストデータの場合、独立したリストを使用することも1つの簡単な解決方法です。

より柔軟な解決方法が必要な場合、独自に選択した基準に基づいてリストデータのアクセス制限を定義することができます。

推奨方法

グローバルプロセス

図 8.2. 複数のオペレータによるカスタマイズ - 単純なプロセス



AssetCenterをチームとしてカスタマイズするには:

- 1 プロダクション用データベースのデータと構造を出力する [修献 149]
 - 1 プロダクション用データベースの構造を出力する [修献 150]
- 2 カスタマイズするデータをプロダクション用データベースから出力する [修献 150]
- 3 出力の結果をソースコントロールツールに保存する [修献 154]
- 4 テストデータの作成 [修献 155]

注意:

この一連のステップは、プロジェクトを初めて実行するときに一度だけ実行します。

- 2 開発用データベースのデータと構造をインポートする [修献 155]
 - 1 開発用コンピュータでソースコントロールからのファイルを同期する [修献 155]
 - 2 ソースコントロールからのファイルを確認します。 [修献 155]

- 3 開発用データベースを作成し、カスタマイズするデータとテストデータをインポートする [修献 156]

 注意:

これらのステップを初めに一度実行したら、あとは必要に応じて実行します。

- 3  開発用データベースをカスタマイズする [修献 159]
 - 1 開発用データベースの構造をカスタマイズする [修献 159]
 - 2 開発用データベースのデータをカスタマイズする [修献 159]
- 4  開発用データベースのデータと構造を出力する [修献 160]
 - 1 カスタマイズした開発用データベースのデータを出力する [修献 160]
 - 2 開発用データベースからカスタマイズしたデータを出力する [修献 160]
 - 3 ソースコントロールツールのファイルを更新する [修献 161]
- 5  データと構造をプロダクション用データベースにインポートする [修献 161]
 - 1 テスト用データベースを作成し、マージしたカスタマイズデータをインポートする [修献 161]
 - 2 テスト用データベースをテストする [修献 162]
 - 3 カスタマイズした構造をプロダクション用データベースに適用する [修献 162]
 - 4 カスタマイズしたデータをプロダクション用データベースにインポートする [修献 162]

プロダクション用データベースのデータと構造を出力する

プロダクション用データベースを直接変更しないので、データベースのデータと構造をテキストファイルの形式で出力する必要があります。

テキスト形式を使用すると、得られたファイルをリビジョンコントロール（またはソースコントロール）ツールで管理することができます。

出力中に得られたファイルによって、以下の操作が可能です。

- カスタマイズの前にプロダクション用データベースの構造をトラッキングする
- カスタマイズとカスタマイズのテストのための開発用データベースを作成する
- 開発用データベースのカスタマイズ後に生成されたファイルを元のファイルと比較する

プロダクション用データベースの構造を出力する

プロダクション用データベースの構造を出力するには:

- 1 出力の結果を保存するフォルダを作成します。
- 2 AssetCenter Database Administratorを起動します。
- 3 プロダクション用データベースに接続します（[ファイル/開く/既存のデータベースを開く] メニュー）。
- 4 データベースの構造を出力します（[ファイル/データベース構造の出力] メニュー）。

データベースの構造を出力すると、以下のファイルが作成されます。

- <接続名>.bin: 一時ユーザの定義（変更不可）
- lstpage_xx.cfg: ページの説明
- lstdet_xx.cfg: 画面の説明
- <接続名>.xml: データベースの構造
- <接続名>_xx.xml: データベースのディクショナリ
- dawkfl.dsd: 自動ワークフローの定義（変更不可）
- tips.str: ワンポイント
- <接続名>_xx.str: フィールドとリンクのコンテキストヘルプ
- <接続名>.wiz: ウィザードの機能ライブラリ

注意:

xx は、（使用可能な）AssetCenterの言語バージョンに対応します。

- en: 英語
- fr: フランス語
- de: ドイツ語
- it: イタリア語
- sp: スペイン語
- ja: 日本語

カスタマイズするデータをプロダクション用データベースから出力する

注意:

プロダクション用データベースにデータが入っていない場合でも、ここで、プロセス全体を通して用いるインポートスクリプトを作成します。

カスタマイズするデータを出力するには:

- 1 データベースからのデータを保持するフォルダを作成します。
- 2 AssetCenter Exportを起動します。
 - ▶ マニュアル『高度な使い方』の「データの出力とSQLビューの作成」の章
- 3 プロダクション用データベースに接続します。
- 4 **【編集／オプション】** メニューを選択します。
- 5 ブランチ **【編集】** を展開します。
- 6 **【データベースオブジェクトの表示】** をSQL名（説明）に設定します。
- 7 この設定を確定します（**【OK】** ボタン）。
- 8 新しい出力スクリプトを表示します（**【ファイル／新規スクリプト】** メニュー）。

ヒント:

サンプルデータに含まれる「custom.scx」テンプレートを使用して出力スクリプトを作成することもできます。

このスクリプトはC:\Program Files\Peregrine\AssetCenter\datakit\customのサブフォルダにあります。

▶ 「custom.scx」スクリプトの使用に際しては「▶ 「custom.scx」スクリプトは「データベースのカスタマイズ - 考慮するテーブル [参考文献 163]」セクションの指示に従っています。

- 9 このスクリプトを選択した名前で作成します（**【ファイル／スクリプトの保存】** メニュー）。

重要項目:

スクリプトに加えた変更を定期的に保存します。

- 10 出力する各データセットのクエリを追加します。通常、1テーブル当たりのデータセットは1つです（**【新規】** ボタン）。
考慮するテーブルのリスト: ▶ データベースのカスタマイズ - 考慮するテーブル [参考文献 163]
- 11 各クエリに対して以下を実行します。
 - 1 クエリに名前をつけます（**【名前】** フィールド）。

 ヒント:

クエリの名前として、出力したテーブルのSQL名を使用することをお勧めします。

例: amAction

- 2 出力したデータの保存に使用するファイルのフルパスを定義します（【ファイル】フィールド）。

 ヒント:

ファイル名として、出力したテーブルのSQL名を使用することをお勧めします。

例: amAction.txt

- 3 【クエリ】フィールドの横にある拡大鏡をクリックします。
- 4 左のリストから右のリストに以下の要素を追加します。
 - 以下を除くすべてのフィールド
 - **【変更日】** (dtLastModif)
 - **階層レベル** (sLvl)
 これらのフィールドは、 アイコンによって識別されます。
 - すべての【メモ】フィールド
 これらのフィールドにはリンクの一覧からアクセスできます。フィールドは、 アイコンによって識別されます。
 - 便利な1リンク（nリンクはなし）
 これらは、 または  アイコンによって識別されます。リンクに対して、関連識別キーを構成するフィールドを選択します（複数選択可）。
 - ▶ **データベースのカスタマイズ - 考慮するテーブル** [修献 163]
 主キーを使用しないでください。
- 5 識別キー（1個の数字を持つ赤い矢印）を構成するフィールドの並べ替えを可能にします。

 警告:

これは、出力したファイルをソースコントロールツールで管理しやすくするために必要です。

ファイルを適切に並べ替えないと、ソースコントロールアプリケーションが対象となるファイル間の違いを解析できません。

 重要項目:

例外: テーブルに **【完全名】** フィールド (FullName) がある場合、**【完全名】** での並べ替えだけをオンにし、**【表示】** チェックボックスをクリアします。

【完全名】 フィールドでの並べ替えを有効にすると、下位レコードの前に上位レコードをインポートすることができます。この方法を用いることで、必ず上位リンクが入力されるようになります。

【表示】 チェックボックスのチェックをクリアすると、**【完全名】** フィールドがエクスポートされません。後で完全名フィールドをインポートできないおそれがあります。

このフィールドはインポート時に親／子リンクを元に自動生成されます。

- 6 **【OK】** をクリックして選択を確定します。
- 7 テーブルの認識キーを構成する各フィールドの後の **【クエリ】** フィールドに、**【キー値】** を追加します。

例

```
SELECT sX, sY, seType, Comment.memComment, WfTransition.Ref keyvalue, WfScheme.Ref keyvalue FROM amWfSysActiv ORDER BY lSysActivId
```

この例では、**【WfTransition.Ref】** フィールドと **【WfScheme.Ref】** フィールドの後に **【キー値】** を追加しています。これらのフィールドが **【amWfSysActiv】** テーブルのレコードの識別キーを構成しているからです。

 注意:

AssetCenter Export 出力中に作成された特定ファイルに「keyvalue」フィールドの値を使用します。

- 8 **【クエリ】** フィールドで、画像に対するリンクがある場合（**【画像】** テーブル (amImage) ）、画像識別子を<リンク>.Nameと置き換えます。

例

```
SELECT Name, SQLName, Comment.memComment, WfActivity.Ref keyvalue, WfEvent.Ref, lIconId FROM amWfUserChoice ORDER BY lUserChoiceId
```

結果

```
SELECT Name, SQLName, Comment.memComment, WfActivity.Ref keyvalue, WfEvent.Ref, Icon.Name FROM amWfUserChoice ORDER BY IUserChoiceld
```

この例では、IIconId がIcon.Nameに置き換えられています。

- 12 **【書式】** タブを表示します。
 - 13 **【列タイトル】** に値SQL名を入力します。
 - 14 スクリプトを保存します（**【ファイル／スクリプトの保存】** メニュー）。
 - 15 スクリプトを実行します（**【アクション／スクリプトの実行】**）。
 - 16 出力の結果を表示します。1クエリあたり1ファイルと、**【メモ】** フィールドと出力レコードあたり1ファイルが得られます。
- 出力の結果に一貫性があることを確認します。

出力の結果をソースコントロールツールに保存する

これにより、以下を実行することができます。

- 変更を加えるたびにデータベース構造とカスタマイズしたデータをトラッキングする。
- 必要に応じて前のステータスに戻り、異なるバージョンを比較することができます。
- 別の人が変更したファイルをマージする

注意:

異なるファイルバージョンのマージは、ソースコントロールツールの基本機能の1つです。

出力プロセス中に取得したファイルをソースコントロールツールに追加します。

重要項目:

これ以降、データベース構造に変更を加えるときや、データベースをカスタマイズするときには、これらのファイルを出発点として使用する必要があります。

出力の結果をソースコントロールツールに保存するには:

使用するソースコントロールツールのマニュアルを参照してください。

テストデータの作成

カスタマイズの際、開発用データベースの処理にテストデータを使用する必要があるものの、それをソースコントロールに保存するのは不便な場合があります。

例えば、アクションを作成するとき、作成者をアクションと関連付ける必要があります。作成者は【従業員】テーブル (amEmplDept) から選択します。このため、開発用データベースに従業員レコードが必要となります。しかしこれらの従業員レコードは、カスタマイズの必要がありません。こうした理由から、従業員はソースコントロールに保存されません。

以下からテストデータセットを作成することをお勧めします。

- プロダクション用データベースからのデータ:
 - ターゲットテーブルのレコード (例えば【従業員】) およびカスタマイズするテーブル(例えば【アクション】)のリンク (例えば【作成者】)
 - テストに必要なテーブルのすべてのレコードまたは選択したレコード
- その他のデータ (必要な場合)

🔥 開発用データベースのデータと構造をインポートする

開発用コンピュータでソースコントロールからのファイルを同期する

開発者は、各自のコンピュータでデータベースをカスタマイズする前に、データがソースコントロールのデータと同期しているデータベースを作成する必要があります。

開発用コンピュータでソースコントロールからのファイルを同期するには:

- 1 ソースコントロールツールをインストールして、設定を行ないます。
- 2 開発用コンピュータでデータベースの構造ファイルを同期します。

ソースコントロールからのファイルを確認します。

各開発者は、変更しようとすることを識別し、対応するファイルを確認する必要があります。

ソースコントロールからのファイルを確認するには:

使用するソースコントロールツールのマニュアルを参照してください。

開発用データベースを作成し、カスタマイズするデータとテストデータをインポートする

各開発者がデータベースを独立してカスタマイズできるように、それぞれの開発者に対して開発用データベースを作成します。

開発用データベースを作成するには:

- 1 SGBDで空のシェルを作成します。
 - ▶ DBMSを使って空のAssetCenterデータベースを作成する [修献 48]
- 2 DBMSログインを作成します。
 - ▶ DBMSログインを作成する [修献 59]
- 3 空のシェルへの接続を作成します。
 - ▶ 空のシェルへの接続を指定する [修献 61]
- 4 AssetCenter Database Administratorを起動します。
- 5 開発用コンピュータのソースコントロールから同期した<接続名>.xmlファイルを開きます（[ファイル/開く] メニュー、[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）。

このファイルは、ステッププロダクション用データベースの構造を出力する [修献 150]で取得します。
- 6 データベース構造を空のシェルに適用します（[アクション/データベースの作成]）。
 - 7 以下のオプションを選択します。
 - ・ [データベースの作成]
 - ・ [システムデータの作成]
 - ・ [タイムゾーンの使用]（このオプションが関係する場合）
 - ・ [フィールドのヘルプの使用]
 - ・ [AutoCADの統合の使用]（このオプションが関係する場合）
- 8 オプション [補足データのインポート] をクリアします。
- 9 [作成] をクリックします。

開発用データベースにカスタマイズするデータをインポートするには:

- 1 AssetCenterを起動します。
- 2 開発用データベースに接続します。

- 3 インポート機能を開始します（[ファイル/インポート]、[データベースのインポート] ボタン、[テキスト] タブ、[開く] ボタン）。
- 4 [編集/オプション] メニューを選択します。
- 5 オプション [Do not import empty fields from the source database（ソースデータベースから空のフィールドをインポートしない）] をクリアします。
- 6 この設定を確定します（[OK] ボタン）。
- 7 選択した名前を使用してブランクのインポートスクリプトを保存します（[ファイル/保存] メニュー）。

重要項目:

スクリプトに加えた変更を定期的に保存します。

ヒント:

サンプルデータに含まれる「custom.scx」テンプレートを使用して出力スクリプトを作成することもできます(ファイル/ スクリプトを開く)。

このスクリプトはC:\Program Files\Peregrine\AssetCenter\datakit\customのサブフォルダにあります。

▶ 「custom.scr」スクリプトの使用に際しては「データベースのカスタマイズ - 考慮するテーブル [修献 163]」セクションに記載された指示をよくお読みください。

- 8 出力したデータファイルをインポートするようにスクリプトを設定します。
 - ▶ インポート [修献 189]
- 9 インポートする各ファイル（各ファイルの出力に対して1つ; ▶ データベースのカスタマイズ - 考慮するテーブル [修献 163]）に対して以下を実行します。
 - 1 ファイルを追加します（[ファイル/追加]）。
 - 2 ウィザードによって表示されるページ（特に以下のページ）に入力します。
 - ページ1: オプション [最初の項目に列名を含める] と [区切り文字] を選択します。
 - ページ2: オプション [タブ] を選択します。
 - ページ3: [日付] または [日付と時刻] データを含む列を選択し、[タイプ] フィールドに入力します。
 - 3 ターゲットテーブル、インポートしたファイルに一致するメインテーブルを選択します。
 - 4 ファイルをテーブルにマッピングします（[編集/マッピング]）。
 - 5 マッピング（ファイル、ターゲットテーブル）を選択します。

- 6 ファイルの列をターゲットテーブルのフィールドにマッピングします（**【編集／プロパティ】**メニュー）。

 ヒント:

 ボタンを使用して、SQL名に基づきフィールドを自動的にマッピングすることをお勧めします。

特定のフィールドに対しては、**【ファイル】** タイプの計算式を作成し、そのフィールドではなく、ターゲットフィールドにマッピングする必要があります。

これは、特に以下のフィールドにあてはまります。

- **【メモ】** フィールド
- **【データ】** フィールド (blbData)
- **【スクリプト】** フィールド (memScript)
- **【機能権限】** テーブルの **【定義】** フィールド
- **【ワークフローイベント】** テーブル (amWfEvent) の **【期間】** フィールド

 ボタンを使用すると、式が自動的に作成され、適切なフィールドに関連付けられます。

これらの式のいずれかの詳細を見るには、以下を表示します。

- **【フィールドのタイプ】** フィールド: **【ファイル】**
- **「メインフィールド」** フィールド: メモを保存しているファイル名を示した、出力したファイルの列名

【期間】 フィールド (timespan の場合、SQL 名が ts で開始します) の場合、次の任意管理項目を持つ公式も作成する必要があります。

- **フィールドタイプ:** スクリプト
- **スクリプト:** RetVal = [<フィールドのSQL名>] & "s"

例: RetVal = [tsDelay] & "s"

作成した特殊フィールドを、(デフォルト設定のソースフィールドではなく) ターゲットフィールドに関連付けます。

- 7 テーブルデータベースのカスタマイズ-考慮するテーブル [参考文献 163] に推奨されたキーをメインテーブルおよびリンクされたテーブルの識別キーとして定義します。

 警告:

 ボタンによって適切なキーが自動的に選択されることはありません。

それらを系統的に検証する必要があります。

- 10 スクリプトを保存します（[ファイル/保存] メニュー）。
- 11 重要な設定を検証します（[編集/オプション] メニュー）。
- 12 スクリプトを実行します（[インポータ] ボタン）。
- 13 インポートの結果に一貫性があることを確認します。

重要項目:

ワークフローチャート以外のテーブルには、これらの手順で十分です。ワークフローチャートの場合は、スクリプトが正しく動作するよう、徹底したスクリプトのテストを行なう必要があります。

開発用データベースにテストデータをインポートするには:

カスタマイズするデータをインポートする場合と同じ方法で新しいインポートスクリプトを作成します。

開発用データベースをカスタマイズする

開発用データベースの構造をカスタマイズする

データベースの構造を複数の開発者が同時に変更することができます。

個々の開発者が開発用データベースの構造をカスタマイズするには:

- 1 AssetCenter Database Administratorを起動します。
- 2 開発用データベースのインスタンスに接続します（[ファイル/開く/既存のデータベースを開く] メニュー）。
- 3 データベースの構造を変更します。
- 4 変更を保存します（[ファイル/保存] メニュー）。

開発用データベースのデータをカスタマイズする

データベースのデータを複数の開発者が同時に変更することができます。

開発用データベースのデータをカスタマイズするには:

- 1 AssetCenterを起動します。
- 2 開発用データベースに接続します（[ファイル/開く/既存のデータベースを開く] メニュー）。
- 3 開発用データベースのデータを変更します。

 ヒント:

レコードを作成または変更した開発者を識別できるように、テーブルデータベースのカスタマイズ - 考慮するテーブル [修献 163]に推奨されたキーを入力します。

例えば各識別子を、それぞれの開発者の接頭コードで始めることができます。

 ヒント:

テストを実行する際、一時レコードの作成が必要となる場合があります。

これらの一時レコードは忘れずに削除してください。

レコードの数が多くなり過ぎた場合は、別のテストおよび開発用データベースの作成が必要です。

4 開発用データベースのデータと構造を出力する

カスタマイズした開発用データベースのデータを出力する

この操作は、カスタマイズした開発用データベースごとに行なう必要があります。

開発用データベースの構造を出力するには:

- 1 AssetCenter Database Administratorを起動します。
- 2 開発用データベースに接続します（[ファイル／開く／既存のデータベースを開く]メニュー）。
- 3 構造ファイルを同期した場所にデータベース構造を出力します（[ファイル／データベース構造の出力]）。

開発用データベースからカスタマイズしたデータを出力する

開発用データベースのデータを出力するには:

- 1 AssetCenter Exportを起動します。
- 2 開発用データベースのインスタンスに接続します。
- 3 出力するデータに対応する各スクリプトに対して、以下を実行します。

- 1 スクリプトを開きます（[ファイル/スクリプトを開く]）。
- 2 スクリプトを実行します（[アクション/スクリプトの実行]）。
 - ▶ カスタマイズするデータをプロダクション用データベースから出力する [修献 150]

ソースコントロールツールのファイルを更新する

開発者は、変更を終了したら、出力したファイルをソースコントロールに戻します。

開発者がファイルを処理しているあいだに、別のユーザがソースコントロールの中央に保存しているファイルを更新した場合、他のユーザによる変更を上書きしないよう、ファイルを戻す前に変更をマージする必要があります。

ソースコントロールツールは、後続バージョン間で行なわれた変更を強調表示するdiff and mergeツールといっしょに使用することができます。

5 データと構造をプロダクション用データベースにインポートする

テスト用データベースを作成し、マージしたカスタマイズデータをインポートする

すべての開発者が自分の開発用データベースをカスタマイズし、結果をソースコントロールに戻したら、テスト用データベースで新しい構造の実用性とカスタマイズデータのインポートを検証する必要があります。

最新の構造を持つテスト用データベースを作成するには:

ステップ開発用データベースを作成し、カスタマイズするデータとテストデータをインポートする [修献 156]と同じ手順に従います。

カスタマイズしたデータをテスト用データベースにインポートするには:

ステップ開発用データベースを作成し、カスタマイズするデータとテストデータをインポートする [修献 156]と同じ手順に従います。

テストデータをテスト用データベースにインポートするには:

ステップ開発用データベースを作成し、カスタマイズするデータとテストデータをインポートする [修献 156]と同じ手順に従います。

テスト用データベースをテストする

データベースの構造が、開発用データベースで行なったカスタマイズと一致することを確認します。

また、開発用データベースでカスタマイズしたオブジェクトを1つまたは複数テストします。

例えばアクションの場合、1～2つのアクションを実行して正しく機能するかをチェックします。

カスタマイズした構造をプロダクション用データベースに適用する

テスト用データベースでのテストに合格したら、プロダクション用データベースを更新することができます。

カスタマイズした構造をプロダクション用データベースに適用するには:

- 1 ソースコントロールツールを起動します。
- 2 構造ファイルとデータを含むフォルダを同期します。
- 3 AssetCenter Database Administratorを起動します。
- 4 プロダクション用データベースを開きます。
- 5 **【アクション／カスタマイズの統合】**を選択します。
- 6 **【データベース記述ファイルから統合】**オプションを選択します。
- 7 **作業フォルダ:** 選択した一時フォルダ
- 8 **ソースファイル:** 同期した構造ファイル<データベース名>.xmlのフルパス（ファイル<データベース名>_xx.xmlを選択しないでください）
- 9 **【統合】**をクリックします
- 10 統合を実行したら、プロダクション用データベースでカスタマイズが正しく行なわれたかを確認します。

カスタマイズしたデータをプロダクション用データベースにインポートする

カスタマイズしたデータをテスト用データベースにインポートするには:

ステップ開発用データベースを作成し、カスタマイズするデータとテストデータをインポートする [参考文献 156]と同じ手順に従います。

参照: カスタマイズ可能なテーブル

次に、カスタマイズ可能なオブジェクトの一覧を示します。

これらのオブジェクトのそれぞれに対して、関連テーブルが示されます。

これらの各テーブルの推奨識別キーは、便利な1リンクとして指定されます（メモリンクと画像リンクは指定されません）。

ヒント:

インポートでは、テーブルの順序が重要となります。インポートした特定のレコードが、すでに存在するはずの別のテーブルのレコードを参照するからです。以下に示すテーブルの順序は、テーブル間の依存状態に従っている必要があります。

表 8.1. データベースのカスタマイズ - 考慮するテーブル

テーブル	推奨キー	便利な1リンク
画像		
画像 (amImage)	名前 (Name)	
特殊フィールド		
特殊フィールド (amCalcField)	SQL名 (SQLName) + テーブル (TableName)	
スクリプト		
スクリプト (amScriptLibrary)	SQL名 (SQLName)	
単位		
単位 (amUnit)	シンボル (Symbol)	
配備		
配備ワークフロー (amDaWfScheme)	参照 (Ref)	
配備サーバ (amDaServer)	名前 (Name)	コンピュータ資産タグ (Computer.AssetTag)
機能ドメイン		
機能ドメイン (amFuncDomain)	SQL名 (SQLName)	親ドメイン (Parent.Code)
クエリ		
クエリ (amQuery)	SQL名 (SQLName)	作成者.IDNo (Author.IDNo)
		ドメイン.SQL名 (Domain.SQLName)
ビュー		

テーブル	推奨キー	便利な1リンク
ビュー (amViewDef)	SQL名 (SQLName)	作成者.IDNo (Author.IDNo)
		ドメイン.SQL名 (Domain.SQLName)
アプリケーションのオプション		
アプリケーションのオプション (amOption)	- セクション(OptSection) - エントリ(OptEntry)	従業員.ID番号 (EmplDept.IDNo)
設定およびオプション		
設定およびオプション (amSysConfig)	- エントリ(CfgEntry) - セクション(CfgSection) - 従業員(EmplDept.IDNo)	従業員.ID番号 (EmplDept.IDNo)
レポート		
レポート (amReport)	SQL名 (SQLName)	作成者.ID番号 (Author.IDNo)
		ドメイン.SQL名 (Domain.SQLName)
形式		
内部形式 (amForm)	SQL名 (SQLName)	作成者.ID番号 (Author.IDNo)
		ドメイン.SQL名 (Domain.SQLName)
リストデータ		
リストデータ (amItemizedList)	識別子 (Identifier)	
リストデータ値 (amItemListVal)	- 値(Value) - リストデータ.識別子 (ItemizedList.Identifier)	リストデータ.識別子 (ItemizedList.Identifier)
任意管理項目		
任意管理項目クラス (amFeatureClass)	コード (Code)	親クラス(Parent.Code)
任意管理項目 (amFeature)	SQL名 (SQLName)	クラス.コード (FeatureClass.Code)
		リストデータ.識別子 (ItemizedList.Identifier)
		単位.シンボル (Unit.Symbol)

テーブル	推奨キー	便利な1リンク
任意管理項目 (amFeature)	SQL名 (SQLName) とソース任意管理項目.SQL名 (SrcFeatures.SQLName) だけを使ってレコードの2度目の出力を行ないます。	
	注意:	
	この2番目のインポートにより、インポートフェーズ中に統合する任意管理項目と統合される任意管理項目との間にリンクを作成することができます。	
	これを実現できるのは、これらの任意管理項目がすべて存在している場合だけです。	
	この条件は、最初のファイルをインポートすることにより保証されます。	
	AssetCenter Export 1 リンクあたり 1 行を生成します。	
任意管理項目パラメータ (amFeatParam)	- テーブル(TableName) - 任意管理項目.SQL名 (Feature.SQLName)	このパラメータのヘルプ、説明 (ShortHelp.Comment) このパラメータのヘルプ、例 (ShortHelp.Example) このパラメータのヘルプ、注 (ShortHelp.Warning) 任意管理項目.SQL名 (Feature.SQLName)
アクション		
アクション (amAction)	SQL名 (SQLName)	作成者.ID番号 (Author.IDNo) 配備サーバ.名前 (DaServer.Name) 配備ワークフロー.参照 (DaWfScheme.Ref) ドメイン.SQL名 (Domain.SQLName) 実行するアクション.SQL名 (ExecAction.SQLName) 形式.SQL名 (Form.SQLName) レポート.SQL名 (Report.SQLName)
ユーザプロファイル		
ユーザプロファイル (amProfile)	SQL名 (SQLName)	デフォルトアクション.SQL名 (DefLoginAction.SQLName)

テーブル	推奨キー	便利な1リンク
ユーザ権限 (amUserRight)	SQL名 (SQLName)	
アクセス制限 (amAccessRestr)	SQL名 (SQLName)	
機能権限 (amFuncRight)	SQL名 (SQLName)	
ユーザ権限 (amUserRight)	SQL名 (SQLName) とプロフィール.SQL名 (Profiles.SQLName) だけを使ってレコードの2度目の出力を行ないます。	
	注意: この2番目のインポートにより、インポートフェーズ中にユーザプロフィールとユーザ権限との間にリンクを作成することができます。 AssetCenter Export 1 リンクあたり 1 行を生成します。	
アクセス制限 (amAccessRestr)	SQL名 (SQLName) とプロフィール.SQL名 (Profiles.SQLName) だけを使ってレコードの2度目の出力を行ないます。	
	注意: この2番目のインポートにより、インポートフェーズ中にユーザプロフィールとアクセス制限との間にリンクを作成することができます。 AssetCenter Export 1 リンクあたり 1 行を生成します。	
機能権限 (amFuncRight)	SQL名 (SQL名) とプロフィール.SQL名 (Profiles.SQLName) だけを使ってレコードの2度目の出力を行います。	
	注意: この2番目のインポートにより、インポートフェーズ中にユーザプロフィールと機能権限との間にリンクを作成することができます。 AssetCenter Export 1 リンクあたり 1 行を生成します。	
ワークフローチャート		
従業員グループ (amEmplGroup)	バーコード (BarCode)	管理者.ID番号 (Supervisor.IDNo) バーコードのサブグループ (Parent.BarCode)

テーブル	推奨キー	便利な1リンク
従業員グループ (amEmplGroup)	バーコード (BarCode) とメンバ.ID番号 (Members.IDNo) だけを使ってレコードの2度目の出力を行ないます。 注意: この2番目のインポートにより、インポートフェーズ中にワークフローグループとメンバとの間にリンクを作成することができます。 AssetCenter Export グループとメンバ間で 1 行ずつエクスポートします。	
ワークフローロール (amWfOrgRole)	参照 (Ref)	指定代理人.ID番号 (Assignee.IDNo) グループ.バーコード (EmplGroup.BarCode)
ワークフローチャート (amWfScheme)	参照 (Ref)	責任者.ID番号 (Supervisor)
システムアクティビティ (amWfSysActiv)	- ワークフロー.参照 (WfScheme.Ref) - タイプ(seType)	ワークフロー.参照 (WfScheme.Ref)
	注意:	
	この最初のファイルでは、WHERE句を追加してSTARTアクティビティを抽出します(seType = 0)。	
ワークフローアクティビティ (amWfActivity)	- ワークフロー.参照 (WfScheme.Ref) - 参照 (Ref)	指定代理人.参照 (Assignee.Ref) カレンダー.SQL名 (Calendar.SQLName) アクション.SQL名 (NotifAssignAct.SQLName) モデル.参照 (Template.Ref) モデル.ワークフロー.参照 (Template.WfScheme.Ref) ワークフロー.参照 (WfScheme.Ref) ウィザード.SQL名 (Wizard.SQLName)

テーブル	推奨キー	便利な1リンク
ユーザ選択 (amWfUserChoice)	- SQL名 (SQLName) - アクティビティ.参照 (WfActivity.Ref) - アクティビティ.ワークフロー.参照 (WfActivity.WfScheme.Ref)	アクティビティ.参照 (WfActivity.Ref) アクティビティ.ワークフロー.参照 (WfActivity.WfScheme.Ref)
ワークフローイベント (amWfEvent)	- ワークフロー.参照 (WfScheme.Ref) - ソースアクティビティ.参照 (SrcActivity.Ref) - 参照 (Ref)	ソースアクティビティ.参照 (SrcActivity.Ref) ワークフロー.参照 (WfScheme.Ref)
ワークフロートランジション (amWfTransition)	- 参照 (Ref) - ソースイベント.参照 (SrcEvent.Ref) - ソースイベント.ソースアクティビティ.参照 (SrcEvent.SrcActivity.Ref) - ワークフロー.参照 (WfScheme.Ref)	ソースイベント.参照 (SrcEvent.Ref) ソースイベント.ソースアクティビティ.参照 (SrcEvent.SrcActivity.Ref) ターゲットアクティビティ.参照 (TargetActiv.Ref) ワークフロー.参照 (WfScheme.Ref)
システムアクティビティ (amWfSysActiv)	- ワークフロー.参照 (WfScheme.Ref) - トランジション.参照 (WfTransition.Ref) - トランジション.ソースイベント.参照 (WfTransition.SrcEvent.Ref) - トランジション.ソースイベント.ソースアクティビティ.参照 (WfTransition.SrcEvent.SrcActivity.Ref)	トランジション.参照 (WfTransition.Ref) トランジション.ソースイベント.参照 (WfTransition.SrcEvent.Ref) トランジション.ソースイベント.ソースアクティビティ.参照 (WfTransition.SrcEvent.SrcActivity.Ref) ワークフロー.参照 (WfScheme.Ref)

注意:

この2番目のファイルでは、WHERE句を追加してENDアクティビティを抽出します(seType = 1)。

テーブル	推奨キー	便利な1リンク
ワークフローアクティビティ (amWfActivity)	ワークフロー.Reference (WfScheme.Ref)、参照 (Ref) とアクション.SQL名 (Actions.SQL name)だけを使ってレコードの2度目の出力を行ないます。 注意: この2番目のインポートにより、インポートフェーズ中にワークフローグループとアクションとの間にリンクを作成することができます。 AssetCenter Export 1 リンクあたり 1 行を生成します。	
ワークフローアラーム (amWfActivAlarm)	- 参照 (Ref) - アクティビティ.参照 (WfActivity.Ref) - アクティビティ.ワークフロー.参照 (WfActivity.WfScheme.Ref)	アクション.SQL名 (Action.SQLName) アクティビティ.参照 (WfActivity.Ref) アクティビティ.ワークフロー.参照 (WfActivity.WfScheme.Ref)
カレンダー		
タイムゾーン (amTimeZone)	コード (Code)	
営業日のカレンダー (amWorkCalendar)	SQL名 (SQLName)	タイムゾーン.コード (TimeZone.Code)
通貨		
通貨 (amCurrency)	名前 (Name)	
為替レート (amCurRate)	- ソース通貨.名前 (SrcCurrency.Name) - ターゲット通貨.名前 (TargetCurrency.Name) レートの日付(dtStart)	ソース通貨.名前 (SrcCurrency.Name) ターゲット通貨.名前 (TargetCurrency.Name)

コメント: 識別キーの選択

識別キーの選択には以下の注意が必要です。

- 主キーを識別キーとして使用することはできません。事実上、各開発者は自分の開発用データベースに固有の主キーを生成します。ただし、これらの識別子は、いったんプロダクション用データベースにインポートすると別のものになります（主識別子をインポートすることはできません。AssetCenterが、インポートプロセス中に自動的に主識別子を生成します）。
- 将来、プロダクション用データベースをカスタマイズし直したときに、レコードを更新するための識別キーが必要です。

例

- 1 カスタマイズの最初の一巡で、開発者は開発用データベースに新規アクションA1を作成します。
このアクションの主キーは、PK1とそのSQL名SQL1です。
- 2 プロダクション用データベースにインポートすると、アクションA1の主キーは自動的に変化します（PK2）。ただし、そのSQL名は変わりません（SQL1）。
- 3 カスタマイズの後続の一巡で、開発者は開発用データベースのアクションA1を変更します。
アクションの主キーはPK1で、そのSQL名はSQL1です。
- 4 プロダクション用データベースにインポートすると、アクションA1がSQL名（SQL1）によってプロダクション用データベースで識別されます。SQL名は識別キーとして使用されます。アクションA1を記述するレコードが更新されます。

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ったが、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作成する必要があります。

▶ [DB2 UDB \[修献 55\]](#)

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。

AssetCenter Database Administrator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

AssetCenter Database Administrator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、タイプ識別子は以下のうちのどれかです。

- 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。
- 有効化が終了したら、データベースのアーカイブを実行できます。

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。以前にアーカイブしたデータがある場合、新たにアーカイブしたデータによって上書きされます。
- 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、AssetCenter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 <Clause F ROM> WHERE <Clause W HERE>)	標準テーブル
バックアップ	BACKUP FROM <Clause F ROM> WHERE <Clause W HERE>)	標準テーブル
Purge	PURGE FROM <Clause FR OM> WHERE <Clause WH ERE>)	標準テーブルとアーカイブ テーブル
復元	RESTORE FROM <Clause F ROM> WHERE <Clause W HERE>)	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、マニュアル『AssetCenterの高度な使い方』の「AQLクエリ」の章を参照してください。

アーカイブデータの表示

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、**【クローズド】** に対応する値、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、**【amRequest】** 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。

- 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、『**プログラマーズリファレンス**』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 **【調達／購入依頼】** メニューを使って、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。



注意:

この例はデータキットにあります。

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実行する場合には、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、『**プログラマーズリファレンス**』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、プロダクション用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いっしょ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いっしょに消去されます。 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。
定義または普通	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（arl<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。

任意管理項目の場合

任意管理項目が関係している場合、アーカイブ機能の動作を指定することが役立ちます。

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。

アーカイブの無効化

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。

！ 警告:

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 | データベースオプション

一部のオプションはデータベースにのみ関連します。
オプションは、データベースへ接続する全クライアントコンピュータに共通しています。
本章ではオプションの内容と設定方法を説明します。

データベースオプションを設定する

1 AssetCenterを起動します。

2

⚠ **警告:**

Adminユーザと【管理者権限】オプションが選択されているユーザのみが、データベースオプション設定へアクセスできます。

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）。

3 データベースオプションのリストを表示します（【管理／データベースオプション】メニュー）。

4 変更するオプションを選択します。

5 以下のいずれかの方法を使用して続行します。

- **【現在の値】** 列内のオプションの値をダブルクリックします。
 - **[Shift]** と **[スペースバー]** を同時に押します。↵
- 6 値を変更するには以下の方法があります。
- 値がリストの一部である場合は、値をリストから選択します。
 - 1行分の値を自由に入力できる場合は、**【現在の値】** 列に値を入力します。
 - 複数行の値を自由に入力できる場合は、オプションウィンドウに下部に表示される編集ゾーンを使って値を記入します。
-

 ヒント:

このウィンドウが表示されない場合は、オプションウィンドウの下枠部分にマウスポインタを置き、画面分割用ポインタが表示されたらポインタを上部ヘドラッグします。

- 7 オプションを確定します（**[OK]** ボタン）。
- 8 AssetCenter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接続してもらいます。

データベースオプションの設定ウィンドウ

 注意:

黒で表示される値は編集不可能です。

青で表示される値は編集可能です。

【デフォルト値】 列では、オプションの標準値を必要に応じて確認できます。

表 10.1. データベースオプション - 使用可能なオプション

セクション	オプション名	説明	値の例
調達	AssetCenter Serverで受領品をポートフォリオに自動的に作成	はい：発注された品目を受領すると、【受領】（SQL名：amReceipt）テーブルにレコードが作成されます。 ただし、受領品は他のテーブル（資産、コンピュータ、ポートフォリオ品目のテーブルなど）には作成されません。 一時レコードは【受領品】（SQL名：amItemsReceived）テーブルに作成され、AssetCenter Serverの【受領品に対応する資産、消耗品などの作成】モジュールにより処理されます。AssetCenter Serverが最終的なテーブルにレコードを作成します。 この機能の利点：AssetCenterではなくAssetCenter Serverを使ってこのタスクを実行することにより、注文品の受領を行うクライアントのパフォーマンスの向上を図ることができます。 いいえ：発注された品目を受領すると、【受領】（SQL名：amReceipt）テーブルにレコードが作成されます。 受領品に対応するレコードは他のテーブル（資産、コンピュータ、ポートフォリオ品目のテーブルなど）にも即時に作成されます。	はい

セクション	オプション名	説明	値の例
調達	受領時に消耗品の経費明細を作成	- はい：消耗品が受領されると経費明細が作成されます。これは消耗品の購入に対応します。 消耗品の消費時（消耗品が親資産に割り当てられる時）には、追加の経費明細は作成されません。 この「消費」も考慮に入れた上で経費明細を作成する必要がある場合は、経費付替え規則を作成しなければなりません。 - いいえ：消耗品を受領後に作成しても、経費明細は作成されません。 消耗品の消費時（消耗品が親資産に割り当てられる時）に、経費明細が作成されます。	はい
インポート	インポート中の履歴を無効にする	- はい：データのインポートにより発生した変更事項に関するレコードは、【履歴】（SQL名：amHistory）テーブルには作成されません。 - いいえ：データのインポートにより発生した変更事項が、履歴を保持するフィールドまたはリンクに関する場合、1レコードが、【履歴】（SQL名：amHistory）テーブルに作成されます。	はい
ウィザード	Basic関数	このオプションは、ウィザードタイプのアクションで呼び出されるBASIC関数を含んでいます。	

セクション	オプション名	説明	値の例
ワンポイント	ワンポイントのテキスト	AssetCenterの起動時に表示されるワンポイント「ご存知でしたか？」のリスト。	ヒント2： [Shift+F9] キーを押すと、ウィザードデバッグが起動されます。
LDAP認証	認証にLDAPサーバを使用する	- はい：データベースへの接続時に入力されるパスワードは、外部LDAPディレクトリ内で検索されます。 - いいえ：データベースへの接続時に入力されるパスワードは、【部署と従業員】（SQL名：amEmplDept）テーブルの【パスワード】（SQL名：LoginPassword）フィールド内で検索されます。	はい
LDAP認証	LDAPサーバ名	AssetCenterユーザのログインとパスワードを保存するディレクトリを表示するサーバ名 このオプションは、 【認証にLDAPサーバを使用する】 オプションが「はい」の場合にのみ使用されます。	ServerLdap
LDAP認証	LDAPサーバポート	ディレクトリに関するデータの要求をリッスンするLDAPサーバのポート このオプションは、 【認証にLDAPサーバを使用する】 オプションが「はい」の場合にのみ使用されます。	389

セクション	オプション名	説明	値の例
LDAP認証	LDAPディレクトリのログインを格納する [amEmplDept] テーブルのフィールドのSQL名	このオプションはデフォルトで、AssetCenterデータベースとLDAPディレクトリで同じログインを使用するものと仮定しています。このログインは、【部署と従業員】 (SQL名: amEmplDept) テーブルの 【ログイン】 (SQL名: UserLogin) フィールドに格納されています。 AssetCenterデータベースとLDAPディレクトリのログインを別のものにすることも可能です。この場合は、LDAPディレクトリのログインを格納するフィールドを選択し、このフィールドのSQL名をオプションで指定する必要があります。 このオプションは、【認証にLDAPサーバを使用する】 オプションが「はい」の場合にのみ使用されません。	Field1
認証	ライセンスファイル	AssetCenterデータベースに挿入したライセンスファイル (【ツール／ファイセンスファイルの編集】) 。 このオプションを現ウィンドウで変更することはできません。	
認証	AssetCenter Server署名	AssetCenter Serverが定期的にデータベースに接続していることを確認するために、AssetCenterにより使用されます。 このオプションを現ウィンドウで変更することはできません。	AssetCenter Serverにより更新されます。
認証	Desktop Administrationサーバ署名	Desktop Administrationサーバが定期的にデータベースに接続していることを確認するために、Desktop AdministrationのServer Administration Consoleにより使用されます。 このオプションを現ウィンドウで変更することはできません。	Desktop Administration Serverにより更新されます。

セクション	オプション名	説明	値の例
認証	Remote Controlサーバ署名	Remote Controlサーバが定期的にデータベースに接続していることを確認するために、Remote ControlのServer Administration Consoleにより使用されます。 このオプションを現ウィンドウで変更することはできません。	Remote Control Serverにより更新されます。
任意管理項目	統合された任意管理項目の再計算を無効にする	• はい：統合された任意管理項目の値が変更する際に、値は再計算されません。 • いいえ：統合された任意管理項目の値が1つでも変更すると、値は再計算されます。	はい
AssetCenter データ ベースへの アクセスの 制御	スロットの更新間隔 (分)	クライアントがログインスロットを更新する間隔	10
アクセス コントロール	自動切断を有効にする	非活動時にデータベースからクライアントを自動的に切断します。	はい
アクセス コントロール	スロットの自動タイムアウト (秒)	AssetCenterが自動的にデータベースから切断する際のタイムアウト時間 このオプションは、 【自動切断を有効にする】 オプションが「はい」の場合にのみ使用されます。	1800
文書	データベースに挿入可能な文書の最大サイズ (バイト)	【文書】 (SQL名：amDocument) テーブルに挿入されるファイルの最大サイズです。	5 242 880
システム データ	ビルド	データベース内にインポートされるシステムデータのビルド番号 システムデータは、データベースの作成時に自動的にデータベース内にインポートされます。 このオプションを現ウィンドウで変更することはできません。	

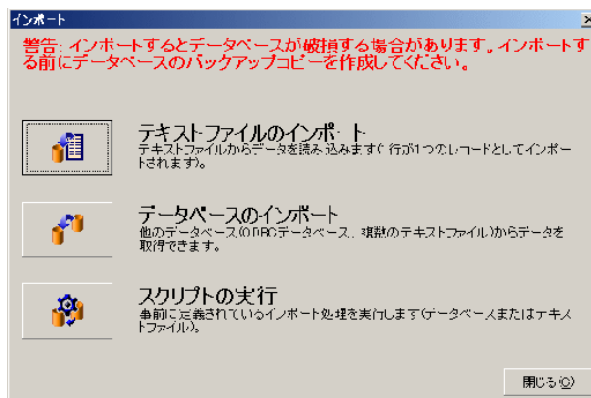
セクション	オプション名	説明	値の例
タイムゾーン	データ：基準となるGMTタイムゾーンとの時間差（時間）	データベース内のデータのタイムゾーン。GMTとの時差を指定します。 値を指定しないとタイムゾーンを使わないことになります。 「0」を指定するとGMTと同じタイムゾーンになります。	-2（GMTのタイムゾーン - 2）
タイムゾーン	サーバ：基準となるGMTタイムゾーンとの時間差（時間）	サーバのタイムゾーン。GMTとの時差を入力します。 値を指定しないとタイムゾーンを使わないことになります。 「0」を指定するとGMTと同じタイムゾーンになります。	4（GMTのタイムゾーン + 4）
イベント管理	出力イベントの有効時間（時間）	出力イベントは、外部アプリケーションにエクスポートするデータを受け取る待ち行列（キュー）です。期限はイベントの解除のために待つ時間の値を指定します。	336
イベント管理	入力イベントの有効時間（時間）	入力イベントは、外部アプリケーションからインポートするデータを受け取る待ち行列（キュー）です。期限は、イベントの解除のために待つ時間の値を指定します。	336
リース	資産リース料率の自動計算の承認	- はい：資産の【料率】（SQL名：pLRF）フィールドは、他の使用可能な値に応じて自動的にAssetCenterによって計算されます。 - いいえ：資産の【料率】（SQL名：pLRF）フィールドは、自動的に計算されません。このフィールド値は、この値に依存する他の値の計算に使用されません。	はい

セクション	オプション名	説明	値の例
全文検索	全文検索を有効にする	シンプルフィルタでは、デフォルトで演算子LIKEを使用できます。 この演算子は、1フィールドの値内でテキストを検索します。 この検索は直接AssetCenterによって実行されます。 一部のDBMSでは更に高性能な検索機能が提供されています。 この場合、この種の機能を使用可能にするには、DBMSで機能を有効にする必要があります。 この種の検索機能では、LIKEの代わりにCONTAINS演算子が使用されます。 【全文検索を有効にする】 オプションをオンにすると、シンプルフィルタのエディタで、CONTAINS演算子を使用できるようになります。	はい

11 | インポート

本章では、AssetCenter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

AssetCenter管理者は、次のファイルやデータベースのデータをAssetCenter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、AssetCenter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、AssetCenter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、AssetCenter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。

AssetCenter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す。

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、AssetCenter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

AssetCenter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」フィールドを調整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、調整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。

AssetCenterデータベースのバックアップ

インポートすると、AssetCenter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。

AssetCenter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AssetCenter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、AssetCenter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]（入力タイプ）プロパティの値

値	結果
Default	「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。
数値または金額	ソースフィールドは数値でなければなりません。 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。 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。
Boolean (Yes/No)	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。
Date	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。 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。 - 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。

値	結果
System itemized list	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。 例 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。 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。 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、AssetCenter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。
Custom itemized list	フィールドの値は、ユーザが変更できるリストデータから選択します。インポートモジュールは、リストデータの値の1つを認識できます。さらに、リストデータが「オープン」タイプの場合は、好きな値を指定できます。この値はリストデータに追加されます。
Percentage	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%」)。
Duration	[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。
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、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。

[dtLastModif] 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 AssetCenter 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

ここでは、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。

複数のテキストファイルを一度に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】オプションの【テキスト】タブページ**を使います。

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】メニューの【データベースのインポート】オプションの【ODBC】タブページ**を使います。

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めることができる。これらのコメント行を特殊文字で始める必要はありません。ファイルのインポート時に行数を指定します。
- 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。この場合、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。
- フィールドタイプは、「数値」、「テキスト」または「日付」である。
- インポートする値は、AssetCenter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。AssetCenter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

次の手順に従って、インポートするテキストファイルを選択します。

- 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するかを指定します：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。

文書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。

AssetCenter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。

最初のインポート項目に列名を含める

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。

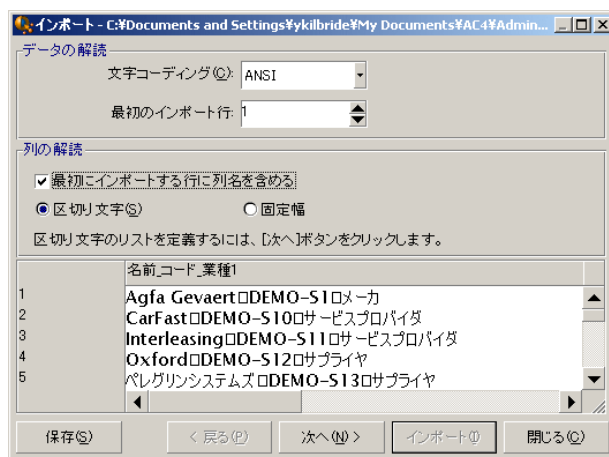
次の画面でこの文字を指定します。

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。

AssetCenter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コロン (;) を使うことをお勧めします。

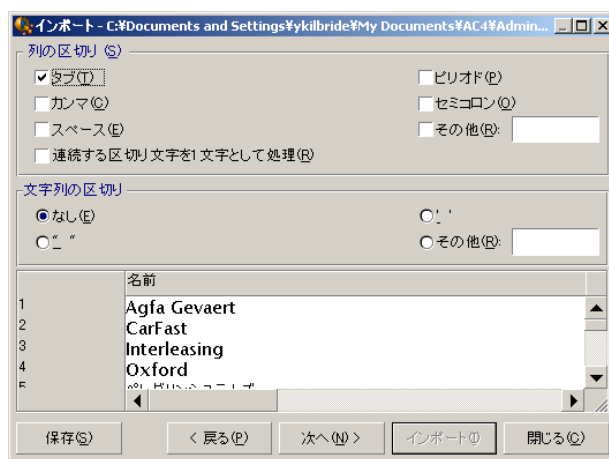
【連続する区切り文字を1文字として処理】 オプションをオンにすると、連続する同じ区切り文字は1つとして扱われ、空の列は作成されません。空の列を作成する場合は、このチェックボックスをオフにして、テキストファイルで2つの区切り文字を続けて使います。

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。AssetCenter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釈を行ったら、【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釈を行ったら、【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釈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【最初の項目に列名を含める】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【最初の項目に列名を含める】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

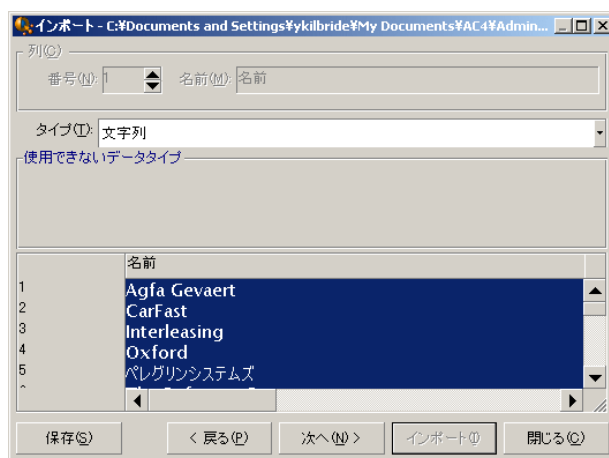
インポートするフィールドのデータ型を指定します。次の選択肢から選択できます。

- 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」オプション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、AssetCenter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AssetCenter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AssetCenter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AssetCenter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】** テーブルのリストに表示される）をAssetCenter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AssetCenterテーブルにマップします。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 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、テキストフィールドをAssetCenter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。

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:

この節は、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。
- [編集/名前でマッピング] メニューを使う。同じ名前のファイルまたはテーブルが自動的にマップされます。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。

AssetCenter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AssetCenter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（AssetCenter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。この場合は、フィールドの技術名を使います。

ソースファイルに特殊フィールドを追加する

AssetCenter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。

AssetCenter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、調整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。

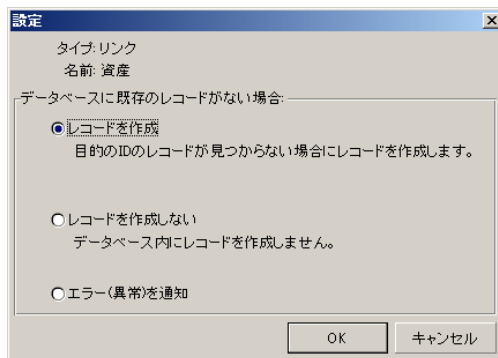
リンク先レコードの作成を設定する

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】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】テーブル）のどこにインポートするかを指定します。

AssetCenter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どう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、次の手順に従います。

- 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、AssetCenter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。

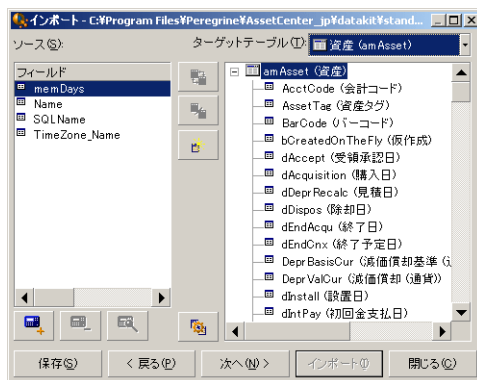
画鋏の影響は、リンクのタイプによって異なります。

- 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」リンクに画鋏を関連付けた場合、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」リンクに画鋏を関連付けた場合、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、画鋏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:

AssetCenter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、カタログは複数サプライヤにリンクでき、サプライヤも複数のカタログにリンクできます。



ターゲットテーブルのツリー構造で使われるシンボル

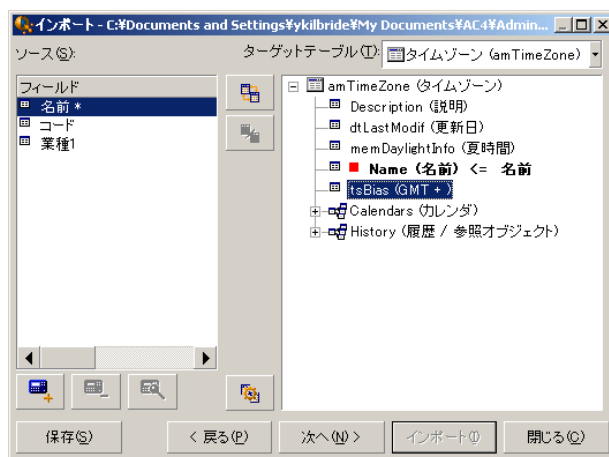
■ オブジェクトがテーブル（大きな記号）またはフィールド（小さな記号）のどちらであるかを示します。

□ 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「画鋏」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。画鋏を設定すると、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。画紙の有無は、 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

AssetCenter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たことができます。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。

- Long text field
- Binary field

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（AssetCenter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す。

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】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。

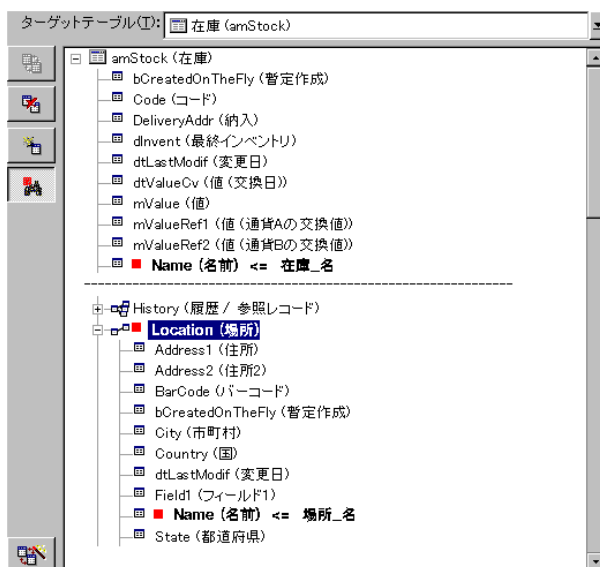
書類をインポートする

書類をインポートする場合は、インポートするフィールドを、**【書類】**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、書類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

AssetCenter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



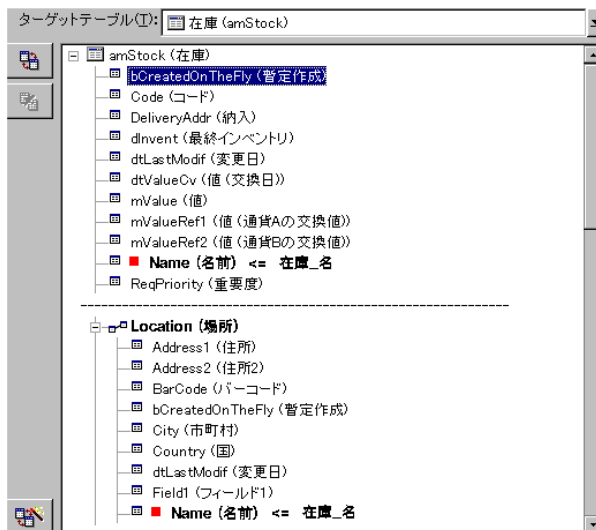
この例では、在庫は2つのメインキーで識別されます。

- Stock.Name：「Name (名前) <= 在庫_名」キーは識別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]** テーブルで識別キーとして宣言されており、「□■ Location (場所)」リンクは識別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]** テーブルで識別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)」リンクは識別子キーではない。

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

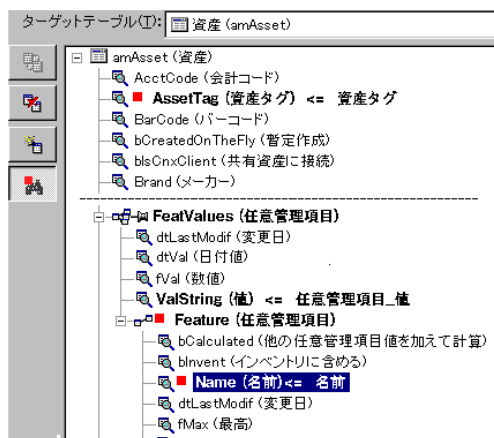
- メインテーブル用にキーを定義できるだけでなく、メインテーブルにリンクしているテーブルにも他の独立したキーを定義できます。これにより、1つのテキストファイルから複数のテーブルにデータをインポートできます。
- リンク先のテーブルのフィールドを、メインテーブルのキーの1つとして宣言するには、リンク先のテーブルとリンクを識別キーとして宣言する必要があります。リンクだけを選択すると、キーはリンク先テーブルのキーとしてのみ使われます。

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。AssetCenter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。

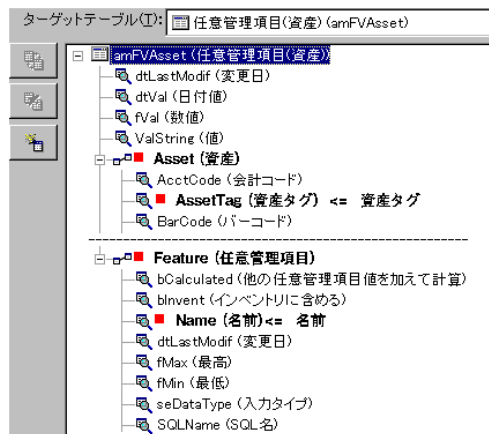
次の2つの方法のどちらかを使います。

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 アイコン  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 アイコン  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 アイコン  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）

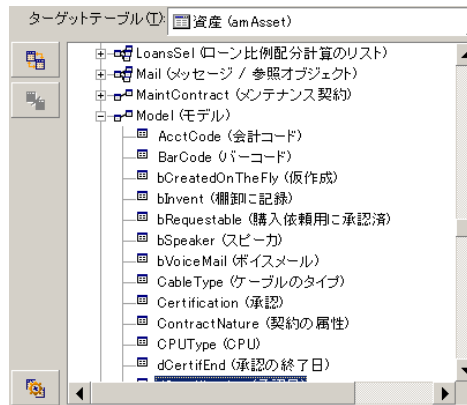


- 1 (■ Feature (任意管理項目)、 ■ Asset (資産)) の組合せにより、資産に関連付けられている任意管理項目が識別されます。
- 2 「 ■ Name (名前) 」 のキーは、「 ■ Feature (任意管理項目) 」 リンクのキーフィールドを示します。
- 3 「 ■ AssetTag (資産タグ) 」 のキーは、「 ■ Asset (資産) 」 リンクのキーフィールドを示します。
- 4 新しい値が [ValString (値) <= 任意管理項目_値] フィールドに表示されます。

例4：識別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 -> 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】 オプション

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】 オプション

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AssetCenter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

- AssetCenter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。
- AssetCenter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。
- 識別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。
- 識別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示さ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

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従業員テーブルを更新できます）。

AssetCenter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

- 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

- 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ファイル*.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選択します。
- 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】ボタンをクリックします。
- 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、次の手順に従います。

- 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完全名を指定します。
- 4 ソースODBC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たときに、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。

amimpl.exe（AssetCenter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で実行したり、バッチファイルを使うなどして自動的に実行することもできます。

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AssetCenterデータベースの接続名（【ファイル／データベース接続の管理】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AssetCenter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	Sybase SQL Anywhere	Sybase SQL Server
エンジン名	Oracle	ODBC	ODBC	Sybase
データベースの場所	サーバ名	データソース名	データソース名	サーバ名：データベース名
ユーザ	アカウント名	MS SQL Server ユーザ名	Sybase SQL Anywhereユーザ名	アカウント名
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	Sybase SQL Anywhereユーザパスワード	アカウントパスワード

-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。AssetCenter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AssetCenterデータベースの接続名（【ファイル／データベース接続の管理】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AssetCenter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AssetCenter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

本章では、AssetCenter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。

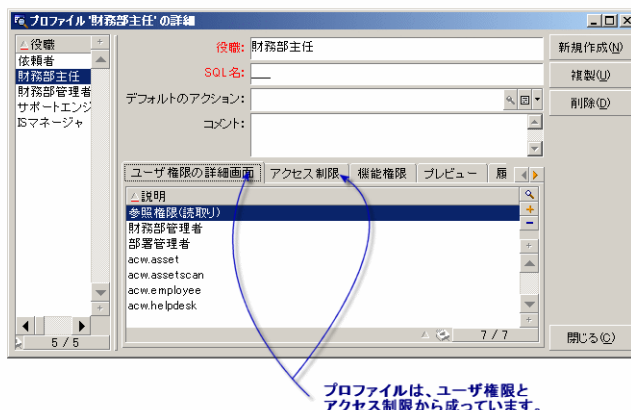
【管理／ユーザプロフィール】 メニューを使ってユーザプロフィールのリストを表示します。

【管理／ユーザ権限】 メニューを使ってユーザ権限のリストを表示します。

【管理／アクセス制限】 メニューを使ってアクセス制限のリストを表示します。

【管理／機能権限】 メニューを使って機能権限のリストを表示します。

図 12.1. ユーザプロフィールの詳細画面



アクセス権限の管理の重要性と概要

AssetCenter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。

AssetCenter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、取得したAssetCenterライセンスのタイプに従って、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

必ずしもすべてのユーザが、データベース内の同じデータを参照したり変更したりできるわけではありません。これは、ユーザの役職や企業の構造によって異なります。例えば、ある従業員には資産を作成する権限、また別の従業員には在庫へのアクセス権、さらに別の従業員には作業指示伝票へのアクセス権が必要になります。

従業員にAssetCenter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。

- **【ログイン】フィールド**（**【従業員と部署】**テーブルの**【プロフィール】**タブ）を入力します。
- データベース管理者は、従業員にユーザプロフィールを割り当てるか、管理者権限を割り当てる必要があります。

ユーザプロフィールには、ユーザ権限、機能権限とアクセス制限があります。

1つのユーザ権限、機能権限またはアクセス制限が、複数のユーザに割り当てられることがあります。また、1つのユーザプロフィールまたは機能権限を複数のユーザにリンクすることもできます。ただし、1人のユーザには、1つのユーザプロフィールしか割り当てられません。

データベースへの接続数を管理する

取得したAssetCenterライセンスによって、データベースへの接続数が制限されます。ライセンスは、同時接続数または宣言ユーザ数で付与されます。

AssetCenterは、接続した各ユーザに接続スロットを割り当てます。

AssetCenterは、データベースの接続スロットを自動的に管理します。ただし、AssetCenterの管理者が管理することもできます。

データのセキュリティと機密性を確実に保護する方法

AssetCenterでは、データベースへのアクセスを次のように3段階で制御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
- AssetCenterユーザのプロファイルの定義
- 定期的なデータベースのバックアップ

アクセス権限の定義

ここでは、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。

- ユーザプロフィールの定義
- ユーザ権限の定義
- 機能権限の定義

- アクセス制限の定義

ユーザプロファイルの定義

ユーザプロファイルは、特定のテーブルとフィールドへのアクセス権限と、その権限の使用対象となるレコードを制限するアクセス制限から成り立っています。

プロファイルは、AssetCenterユーザに割り当てられます。

例えば、次のようにアクセスを制御できます。

- 「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。

ユーザ権限の定義

ユーザ権限は、ユーザプロファイルの1つです。AssetCenterの特定のレコードだけでなくテーブルとフィールドにもその権限が適用されます。管理者は、AssetCenterデータベースのユーザ1人1人に対して、それぞれのユーザが使用するテーブルの読取り／書込み権限を、オペレーティングシステムの権限を割り当てると同じ要領で割り当てることができます。

機能権限の定義

機能権限は、ユーザプロファイルの1つです。機能権限は機能カテゴリ（例：調達、ケーブル）に関連し、機能ドメインに基づいています。管理者は、各ユーザに対してユーザの役職に応じて権限を割り当てます。ユーザに直接関連する機能の権限のみを割り当てます。

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つです。テーブルのレコードのフィルタに相当します。例えば、技術者は技術者の部署の資産にしかアクセスできないようにできます。アクセス可能なレコードで、読取り／書込み（追加または変更）を行うことができます。

アクセス条件を定義する

ここでは、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。

- ユーザプロファイルを定義する
- ユーザ権限を定義する
- 機能権限を定義する
- アクセス制限を定義する

アクセス条件の定義が完了すると、AssetCenterユーザに関連付けられます。

ユーザプロファイルを定義する

ユーザプロファイルを定義するには、**【管理／ユーザプロファイル】**メニューを選択します。

ユーザプロファイルは、次のプロファイルで構成されます。

- データベースのテーブルまたはフィールドの情報の読取り、書込み、作成、削除を行うユーザ権限
- 特定のテーブル内のレコードに対する読取り／書込み条件を定義するアクセス制限。例えば技術者は、自分の現場またはユーザグループの現場に配置された設備のデータしか閲覧できないように、制限されます。

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／ユーザ権限】**メニューを使う。
- ユーザプロファイルの詳細画面（**【管理／ユーザプロファイル】**メニュー）で、**【ユーザ権限】**タブページのリストの右側にある  ボタンをクリックする。

ユーザ権限の詳細画面で、データベースのテーブルとフィールドへのアクセス権限を定義します。

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

ユーザ権限の詳細画面では、データベース構造がテーブルのツリーとして表示されます。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。

ユーザ権限画面は、AssetCenterデータベースへのアクセスを管理するために用いられ、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 (R/U/W)
+ AQLタイプの特殊フィールドスクリプト (amCFAg)	---/--
+ Automated Desktop Administrationトラッキング (amD...)	---/--
+ Documents de reconciliation (amReconcDoc)	---/--
+ Scriptタイプの特殊フィールドスクリプト (amCFScript)	---/--
+ アクション (amAction)	r--/--
+ アクションのメモ (amActionMemo)	R--/--
+ アクセス制限 (amAccessRestr)	---/--
+ アプリケーションのオプション (amOption)	---/--
+ アラーム (amDateAlarm)	r--/--
+ インストール済ソフトウェア (amSoftInstall)	---/--
+ エスカレーションレベル (amEscSchLevel)	r--/--
+ エスカレーション処理 (amEscalScheme)	r--/--
+ エージェント (amIfAgent)	---/--

- **【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、AssetCenter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。

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、【賃貸料】と【ローン】タブページを表示できません。

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／機能権限】メニューを使う。
- ユーザプロファイルの詳細画面（【管理／ユーザプロファイル】メニュー）で、【機能権限】タブページのリストの右側にある  ボタンをクリックする。

機能権限の編集画面には、ユーザが定義するAssetCenterの機能ドメインの完全な階層リストが表示されます。

機能権限はアプリケーションの機能ドメインを定義し（【アイテム】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。

機能権限の論理

各機能ドメイン（【アイテム】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権利（【許可】、【拒否】、【親の値】）を割り当てます。

全機能ドメインは、権限のデフォルト値（【デフォルト値】）を継承します。デフォルト値は、アイテム全体（ドメインとその子アイテム）用に定義されています。

機能ドメイン内の各アイテムは親の権限値を継承します（**【親の値】**が選択されている場合）。

各機能ドメインに、**【デフォルト値】**フィールドとは異なる権限が指定されている場合もあります。この場合、このドメイン内の子アイテムは、機能ドメイン値を継承します（このドメイン内の子アイテムの**【親の値】**が選択されている場合）。

ドメイン内の各アイテムに、ドメインの値と異なる値がある場合もあります。この場合、各子アイテムの値は**【許可】**と**【拒否】**フィールドで定義されます。

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。

- **【デフォルト値】**フィールドが「拒否」である。
- 機能ドメインの値が「親の値」である。
- ドメイン内の子アイテム全体の値が「親の値」である。

または

- **【デフォルト値】**フィールドが「許可」である。
- 機能ドメインの値が「拒否」である。
- ドメイン内の子アイテム全体の値が「親の値」である。

または

- **【デフォルト値】**フィールドが「拒否」である。
- 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。

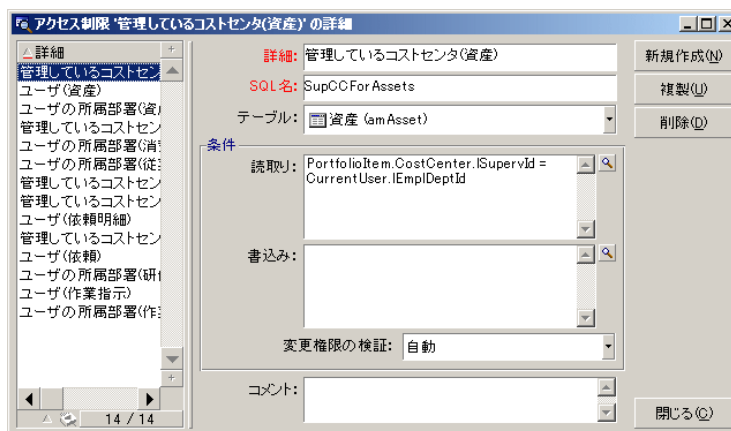
アクセス制限を定義する

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／アクセス制限】**メニューを使う。
- ユーザプロファイルの詳細画面（**【管理／ユーザプロファイル】**メニュー）で、**【アクセス制限】**タブページのリストの右側にある  ボタンをクリックする。

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。

図 12.2. ユーザプロファイル - アクセス制限



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、メーカ、資産
- 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、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

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、読取りまたは書き込みのアクセス制限を定義します。これらの条件は、AssetCenter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。

書き込み条件

既存のレコード内のフィールドの書き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。
AssetCenterでは自動的に処理されません。

変更権限の検証

このフィールドは、変更権限の検証の条件を指定します。【書き込み条件】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

【書き込み】（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】 変更権限の検証を選択する場合は、在庫に入っている品目を変更または削除できます。

【変更後】 変更権限の検証を選択する場合は、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の計算では、アクセス制限が無視されます。このため、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計算に使うように定義する必要があります。

AssetCenterのユーザを定義する

新しいユーザを定義するには、次の手順に従います。

- 1 【部署と従業員】 テーブルでユーザを作成します。
- 2 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。
- 3 従業員の【ログイン】 名とパスワードを入力します。【ログイン】 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。【ログイン】名だけ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:

LDAPサーバがパスワードを管理する場合、AssetCenterデータベースに入力されるAdminユーザ以外のパスワードは、考慮されません。

- 4 【ログインのタイプ】（SQL名：seLoginClass）フィールドに入力します。この後の手順は、その従業員に管理者権限を割り当てるかどうかによって異なります。

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】タブページで、【管理者権限】（SQL名：bAdminRight）チェックボックスをオンにするだけで、使用しているAssetCenterデータベースのすべての権限がその従業員に付与されます。

 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AssetCenterをインストールしたときは、「Admin」だけがAssetCenter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。

管理者でないユーザを定義する

この場合は、従業員の詳細画面の【プロフィール】タブページ／【プロフィール】（SQL名：Profile）フィールドで、ドロップダウンリストからユーザプロフィールを選択します。

データベースへの接続数を管理する

ここでは、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

AssetCenter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。

各ユーザのアクセスタイプを定義するには、従業員の詳細画面の【プロフィール】タブページでログインタイプを指定します。

ログインタイプによって、Windows版AssetCenterとAssetCenter WebのユーザインタフェースまたはAssetCenter APIを使ってデータベースにアクセスするユーザ数を管理します。



注意:

AssetCenter Export、AssetCenter ServerおよびAssetCenter Database Administratorでデータベースに接続する場合は、ログインタイプは無視されます。

不特定アクセス

不特定アクセスでは、ライセンスで決められている最大同時接続数のユーザがデータベースにアクセスできます。

【ログイン】名は必要なだけ定義できますが、同時に接続するユーザ数が最大数に達すると、他のユーザは接続できなくなります。

同じ【ログイン】で複数のユーザが接続することも可能ですが、接続数は実際に接続しているユーザ数でカウントされます。

固定アクセス

固定アクセスでは、ライセンスで決められている最大【ログイン】名数のユーザがデータベースにアクセスできます。

不特定アクセスとは異なり、登録されているユーザだけがいつでもデータベースに接続できます。

ライセンスで許可されている【ログイン】名以外を入力すると、AssetCenterからエラーメッセージが表示されます。

また、複数のユーザが同じ【ログイン】名を使って接続することはできません。

**注意:**

「Admin」 **【ログイン】** のアクセスタイプは、**特定**です。「Admin」 **【ログイン】** は、許可されたユーザ数には入りません。

一時アクセス

このログインタイプは、ほとんどデータベースにアクセスしないユーザに適しています。一時ユーザは、標準のログインとパスワードを使うことができますが、使用できるアクセス権限は限られています。

このタイプの一時アクセスは、AssetCenter Web経由でデータベースにアクセスするユーザに向いています。

例えば、一時ユーザはデータベースで次の作業を行うことができます。

- 自分が資産を表示する。
- 購入依頼を作成する。
- 作成した購入依頼の処理経過をトラッキングする。

一時ユーザのアクセス権限は、次の方法で制限されます。

- ユーザプロファイルの割り当て
- AssetCenterで定義したアクセス制限の組み合わせ

上記の2つの制限はAND句でリンクされます。

接続できる一時ユーザ数に制限はありません。

複数のユーザが同じ**ログイン**を使ってデータベースに同時に接続できます。

接続スロットの機能

ユーザがAssetCenterまたはAssetCenter Webからデータベースに接続すると、AssetCenterは接続スロットを割り当てます。

ユーザがデータベースに接続している間、AssetCenterは接続スロットを定期的に更新します。更新の周期は、**【管理／データベースオプション】**メニューを選択すると表示されるテーブルの、「アクセスコントロール」セクションの**【接続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分に設定されます。

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

AssetCenterの管理者は、**【管理／接続状況】**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AssetCenterなど）
- ユーザの【ログイン】
- ユーザの【ログインタイプ】（SQL名：seLoginClass）

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

「Admin」ログインを使う管理者は、強制的にユーザを切断できます。手順は以下のとおりです。

- 1 【管理／接続状況】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。



注意:

上記の作業を実行できるのは「Admin」ログインだけです。管理者権限のある他のログインでは実行できません。

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（【ファイル／データベースの接続解除】メニューを選択するなど）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になります。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、実際に接続しているユーザ数が1人減ります。

AssetCenter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。AssetCenter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 AssetCenter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、AssetCenterはスロットが無効であり、再利用できると見なします。

「Admin」ログインを使う管理者は、使わなくなった接続スロットを手動で削除して問題を解決することもできます。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 クエリフィルタを使って、接続スロットのテーブル内で **【変更日】** 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。
- 2 これらのレコードのスロットを選択して削除します。

無効なユーザを切断する

AssetCenterでは、タイムアウトを設定して不特定ユーザを自動的に切断できます。

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 **【自動切断を有効にする】** オプションを使って、自動切断機能を有効にするかどうかを指定します。
- 2 自動切断機能を有効にしたら、**【アクセス制御】** セクションの **【スロットのタイムアウト（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3.0.1以前のAssetCenterでは、この自動切断機能が、全てのユーザに適用されます。

パスワードを管理する

ここでは、AssetCenterでパスワードを管理する方法について説明します。

AssetCenter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。

- AssetCenterデータベースに直接格納される
- 外部LDAPディレクトリ内に格納される

「Admin」 ログインの場合

ユーザのパスワードがAssetCenter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AssetCenter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AssetCenterをインストールした時に、AssetCenter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、AssetCenter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AssetCenter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。

Adminログインのパスワードを変更する

「Admin」ログインでデータベースを開き、**【ツール／パスワードの変更】**メニューを選択すれば、「Admin」ログインのレコードのパスワードを変更できます。

「Admin」ログインのパスワードを忘れた場合

上記の注意にも係らずこのパスワードを忘れた場合、AssetCenterによってパスワードを回復する手段はありません。Peregrine Systemsサポートにお問い合わせください。

管理方法1：AssetCenterデータベース内にパスワードを格納する

パスワードの作成

パスワードは**【パスワード】**（SQL名：LoginPassword）フィールドに格納されます。

パスワードの変更

ユーザ自身を変更する

AssetCenterユーザは、それぞれのログイン名でデータベースを開きます。**【ツール／パスワードの変更】**メニューを選択すると、パスワードを変更できます。

管理者が変更する

管理者は、従業員の**【プロフィール】**タブページでユーザのパスワードを変更できます。古いパスワードが新しいパスワードで上書きされます。

パスワードを忘れた場合

AssetCenterでは、他のユーザ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。AssetCenterのユーザ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。

管理方法2：LDAPディレクトリ内にパスワードを格納する

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、ディレクトリを表示するため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、AssetCenter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。

- 義務化されたパスワードの定期変更
- 文字数とフォーマットの制御
- その他

LDAPディレクトリ使用の導入

- 1 AssetCenter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 AssetCenter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 AssetCenterを起動します。
- 5 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）。
- 6 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）。
- 7 LDAP認証項目のオプションにデータを入力します。
- 8 [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。

特に以下のフィールドに値を入力します。

- [ログイン]（SQL名：UserLogin）：AssetCenter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：<名前> <姓>/<ドメイン名>

ヒント:

Connect-Itを使用するとLDAPディレクトリをインポートできます。

- 9 LDAP DLLがAssetCenter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。
AssetCenter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。
- 10 AssetCenter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接続してもらいます。

AssetCenterデータベースへのLDAP接続

AssetCenter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 AssetCenterを起動します。
- 2 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）。
 - ログイン： **[部署と従業員]**（SQL名：amEmplDept）テーブルの **[ログイン]**（SQL名：UserLogin）フィールドのログイン
AssetCenter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 NT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サーバ経由で管理されます。

13 | AssetCenter Server

本章では、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、AssetCenterではなくAssetCenter Server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。

AssetCenter Serverの概要

AssetCenter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AssetCenter Serverと呼ばれ、AssetCenterから独立して機能します。

AssetCenter Server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）
- 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、AssetCenter Serverは内部メッセージシステムを介して、AssetCenter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:**

AssetCenter Server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。

AssetCenter Server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、モニタ対象の期限や実行するタスクをそれぞれのAssetCenter Serverで分担できるため、AssetCenter Serverの性能が向上します。

**注意:**

1つのタスクを、複数のAssetCenter Serverで分担しないでください。

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。

AssetCenter Serverを実行する

推奨事項

AssetCenter Serverは、頻繁にデータベースにアクセスします。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AssetCenter Serverを実行します。すべてのユーザー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AssetCenter Serverを実行できます。

データベースの構造を変更する場合

AssetCenter Database Administrator、またはAssetCenterの【オブジェクトの設定】ポップアップ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AssetCenter Serverをデータベースから切断し、作業が完了してから再接続する必要があります。

AssetCenter Serverを起動する

AssetCenter Server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AssetCenterプログラムグループからAssetCenter Serverプログラムを起動します。



警告:

AssetCenter Server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。AssetCenterからデータベースに接続すると、AssetCenter Server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。

AssetCenter Serverをサービスモードで自動的に起動する

次の手順に従って、AssetCenter Serverをサービスモードで起動します。

- 1 AssetCenter Serverを手動で起動します。
- 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 AssetCenter Serverを終了します。
- 4 Windowsのコントロールパネルで、AssetCenter Serverを選択し、スタートアップの種類を「自動」に設定します。

**警告:**

AssetCenter Serverサービスの接続先のデータベースのDBMSクライアントレイヤをインストールした後、AssetCenter Serverがインストールされたコンピュータを、再起動しなければなりません。これは、Windowsのサービスコントロールマネージャが、コンピュータの起動時に環境変数PATHを読み取るためです。最新のPATH環境変数は、AssetCenter ServerがDBMSのDLLを検索するために必要になっています。これは特にSQL Anywhereランタイムをインストールし、AssetCenter Serverをデモ用データベース（SQL Anywhereを使用）上で実行する場合に注意が必要です。

DOSコマンドプロンプトからAssetCenter Serverを起動する

次のコマンドを使って、AssetCenter Server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、AssetCenter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。

<>の間にスペースは入れません。

例

```
amsrv -cnx:BasePeregrine -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AssetCenter Serverを手動で実行する

AssetCenter Serverをデータベースに接続する

AssetCenter Serverからデータベースに接続できるのは管理者だけです。管理者は、「Admin」ユーザまたは管理者権限を有するユーザです。

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。AssetCenter Serverをサービスモード（Windows 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。

AssetCenter Server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。

- 起動時に表示されるダイアログボックスを使う。
- **【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。

AssetCenter Serverをデータベースから切断する

AssetCenter Server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。

Windows NT、2000、XPの統合セキュリティを使用する

Windowsの**統合セキュリティ**（または**統合ログイン**）を使用することは、AssetCenterのセキュリティ情報をWindowsユーザマネージャのセキュリティ情報と同期することを意味します。

同期を実行すると、以下のことが可能になります。

- ユーザマネージャで宣言されているユーザのリストをAssetCenterデータベースに自動的にインポートできる。
- AssetCenter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る。

概要

- AssetCenterとユーザマネージャとの間での同期は、Connect-Itのシナリオを使って実行します。
- AssetCenter ServerがConnect-Itのシナリオをスケジュール設定し、トリガします。
- Connect-Itは以下のフィールドを使ってAssetCenterデータベースのユーザを識別します。
 - **【姓】** (Name)
 - **【名前】** (FirstName)
 - **【ログイン】** (UserLogin)
- 以下の2値が同一である場合、ユーザはログインと共にNTセキュリティを用いてAssetCenterデータベースへ接続することを許可されます。

- AssetCenterユーザの【SID】（SQL名：Identifier）：Windowsのログインに対応します。
- ログインに関連付けられたWindows SID

導入の手順

導入手順は以下のとおりです。

- 1 Windows NTのユーザマネージャでユーザとグループを作成する。
目的：AssetCenterに転送する情報を準備する。
- 2 AssetCenter ServerとConnect-Itを設定する。
目的：対象となるNTドメインを定義し、データ転送を計画する。
- 3 AssetCenter Serverの【データベースにNTユーザを追加】モジュール（および必要に応じてその他のモジュール）を起動する。
目的：データ転送を実行し、ログインをテストする。
- 4 AssetCenter データベースでユーザの情報を補足する。
- 5 AssetCenter Serverモジュールの自動トリガを有効にする。

手順1：Windows NTのユーザマネージャ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:

AssetCenterデータベースで従業員（ユーザ）を作成する際に、AssetCenter Serverは【フルネーム】フィールド（ユーザマネージャ）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、【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、【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、AssetCenter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：【名前】フィールド

このフィールドは、AssetCenterデータベースの【姓】（SQL名：Name）に入力するために使用されます。

手順2：AssetCenter ServerとConnect-Itを設定する

- ▶ 【データベースにNTユーザを追加（AddUser）】 モジュール [修献 264]

手順3：更新プロセスを始めてトリガする

- 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。

AssetCenterは更新を開始します。



注意:

このタスクに關与するユーザ数が、ライセンスで許可された**固定ユーザ数**を超える場合、AssetCenter Serverはユーザを**不特定ユーザ**として指定します。

手順4：AssetCenter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人を含むことになります。ログインは次のとおりです。

【ドメイン】【ユーザ】

作成された従業員は、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AssetCenter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：AssetCenter Server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】 枠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。

AssetCenterへの接続

前述の操作が終了すると、NTユーザは、AssetCenter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。

基本的な情報

サポートされる環境

- Windowsドメイン：Windows NT 4がサポート可能です。Windows 2000 Active Directoryはサポートされません。
- Windowsクライアント：Windows NT、2000、XPがサポート可能です。Windows ME、95、98はサポートされていません。

ユーザの作成／変更時にAssetCenterが適用する規則

- AssetCenterログインは、Windowsの**【ドメイン】**名と**【ユーザ名】**を連結して作成されます（例：<ドメイン名><ユーザ名>）
- 以下のフィールドが、AssetCenterデータベースのユーザの識別キーとして使用されます。
 - **【姓】**（SQL名：Name）
 - **【名前】**（SQL名：FirstName）
 - **【ログイン】**（SQL名：UserLogin）
- あるユーザをインポートすると、識別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、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。

識別キーに一致するレコードが既に存在する場合は、Connect-Itのシナリオに含まれている、識別キーを除く全フィールドが更新されます。

**注意:**

【パスワード】フィールドは特殊文字を使って入力されており、ユーザが自動ログイン以外の方法でAssetCenter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）。

- AssetCenter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、AssetCenter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。

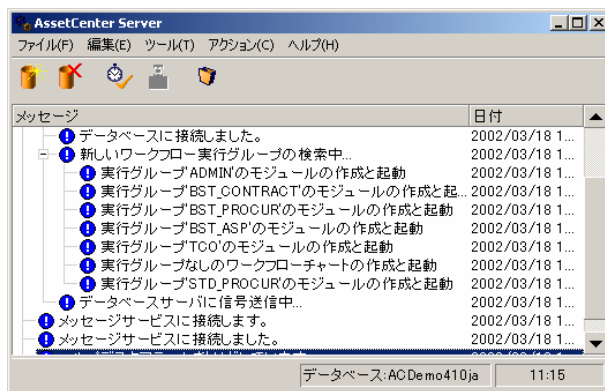
AssetCenter Serverを終了する

AssetCenter Server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。

AssetCenter Serverとメッセージシステム

AssetCenter Server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。

AssetCenter Serverのメイン画面



メイン画面には、AssetCenter Serverが処理するイベントの一覧が表示されます。

AssetCenter Serverの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. プログラムのアイコンとメニュー

アイコン	メニュー	機能
	ファイル / 接続	データベースに接続します。
	ファイル / 切断	データベースとの接続を切断します。
	アクション / 起動	起動する期限のモニターエージェントを選択します。
	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。

アイコン	メニュー	機能
	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します。 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テストします。

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】コマンドを使ってウィンドウを空にすることができます。

AssetCenter Serverの全般オプション

全般的なモニタオプションは、【オプション／設定】メニューを選択すると表示される画面の【全般】タブページで指定します。

これらのオプションで、AssetCenter Serverの全般的な操作を管理します。

【データベース】への再接続の間隔

このオプションは、AssetCenter Server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、AssetCenter Server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、AssetCenter Serverが、【再接続の間隔】枠内の【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。

AssetCenter Server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。

このフィールドには、「時間」を指定します。

【メッセージ】システムへの再接続の間隔

外部メッセージシステムに問題が発生した場合、AssetCenter Server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。

AssetCenter Server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。

このフィールドには、「時間」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、AssetCenter Server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、AssetCenter Server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指定します。

この保存容量が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 [全般] タブページで、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AssetCenter Server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、AssetCenter Server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AssetCenter Serverをインストールしたマシンの時刻のいずれかを変更して、2つの時刻の差を5分以内になると、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、AssetCenter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、AssetCenter Serverが稼動しているマシンで、時刻と夏時間の変更に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

AssetCenter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:

このオプションをオンにすると、DBMSサーバの時刻をAssetCenter Serverが稼動しているいずれかのコンピュータの時刻と比較します。必要に応じてAssetCenter Serverのタイムゾーン情報が調節され、AssetCenterクライアントがGMT時刻で時刻を保存できるようになります。

AssetCenter Serverでモニタするモジュールを設定する

はじめに

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】メニューを使います。

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:

AssetCenter Serverのセッションを複数のマシンで起動して、各マシンのAssetCenter Serverで異なるモジュールを選択して、モニタするモジュールを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、あるマシンで、同時に複数のモジュールを起動しないでください。

以下では、AssetCenter Serverが管理する各モジュールについて説明します。

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、次の手順に従います。

- 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】セクション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。
曜日指定	【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。 例 毎週月曜日
日付指定	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。 例 7月20日

【実行日】フィールドの値

第1
第2
第3
第4
最後から2番目
最後

モニタの実行間隔

【曜日指定】で定義した曜日に、【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。

例

毎月「第1」金曜日
9月の「第2」月曜日
11月の「最後から2番目」の水曜日
2000年の毎月「最終」火曜日

時刻をモニタする

定期的

1日に定期的に行うモジュールの検証を2種類定義できます。

1つめの実行間隔は、設定した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候補値を入力します。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>

AssetCenter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【期間外の実行間隔】は、前の手順で定義した期間外に適用されません。

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入力できます。

- AssetCenterで定義した時刻形式に従う必要があります。
- 複数の時刻を指定する場合は、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;
------	--------	--------------------------------------

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、【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 (AddCpu)】モジュール

AssetCenter Server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、Connect-Itの「addcpu.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。

- AssetCenter Server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
- Connect-Itの「addcpu.scn」シナリオのパラメータ（Connect-Itのインストール先フォルダの「scenario\ntsec\ntac40」サブフォルダ内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【ユーザデータ】フィールド

デフォルトのパラメータは以下のとおりです。

```
"$connectit_exedir$/conitsvc.exe" -once '$connectit_exedir$/../scenario/ntsec/ntac40/addcpu.scn'
-d:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$ -d:AssetCenter.EXTPASSWORD=$pwd$
```

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（Connect-Itの「Once」スケジューラを使用する）。
このパラメータは、変更しないでください。AssetCenter Serverでスケジュールを定義します。
- `$connectit_exedir$../scenario/ntsec/ntac40/addcpu.scn`は、使用するConnect-Itシナリオへのアクセスパスです。
別の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:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$ -d:AssetCenter.TEXTPASSWORD=$pwd$`は、AssetCenter Serverが開くAssetCenter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AssetCenter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。

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、次の手順に従います。

- 1 Connect-Itのシナリオビルダを実行します。
- 2 「addcpu.scn」シナリオを開きます（Connect-Itのインストール先フォルダの「scenario\ntsec\ntac40」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。

- AssetCenter Server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
- AssetCenter Server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、次の操作を実行します。

- 1 AssetCenter ServerのマシンでWindowsエクスプ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 AssetCenter Server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。

AssetCenterデータベースにどのデータが入力されるかを知るには、「addcpu.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、Connect-It付属マニュアル『ユーザガイド』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」の節を参照してください。

AssetCenterコネクタの詳細については、Connect-It付属マニュアル『ユーザガイド』の「Peregrine Systemsコネクタ」の章、「Asset Managementコネクタ」の節を参照してください。

【データベースにNTユーザを追加 (AddUser)】モジュール

AssetCenter Server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、AssetCenter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、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。

- AssetCenter Server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
- Connect-Itの「adduser.scn」シナリオのパラメータ（Connect-Itのインストール先フォルダの「scenario\ntsec\ntac40」サブフォルダ内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以下のとおりです。

```
"$connectit_exedir$/conitsvc.exe" -once '$connectit_exedir$/../scenario/ntsec/ntac40/adduser.scn'
-d:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$ -d: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（Connect-Itの「Once」スケジューラを使用する）。
このパラメータは、変更しないでください。AssetCenter Serverでスケジュールを定義します。
- \$connectit_exedir\$/../scenario/ntsec/ntac40/adduser.scnは、使用するConnect-Itシナリオへのアクセスパスです。
別の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:AssetCenter.SERVER=\$cnx\$ -d:AssetCenter.LOGIN=\$login\$
-d:AssetCenter.TEXTPASSWORD=\$pwd\$は、AssetCenter Serverが開くAssetCenter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AssetCenter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。

Connect-Itの「addcpu.scn」シナリオのパラメータ

「adduser.scn」シナリオを変更するには、次の手順に従います。

- 1 Connect-Itのシナリオビルダを実行します。
- 2 「adduser.scn」シナリオを開きます（Connect-Itのインストール先フォルダの「scenario\ntsec\ntac40」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Connect-ItシナリオとAssetCenter Server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。

- AssetCenter Server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
- AssetCenter Server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、次の操作を実行します。

- 1 AssetCenter Serverのマシンで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 表示されるユーザから、AssetCenter Serverはデータを取得することができます。

AssetCenter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、Connect-It付属マニュアル『ユーザガイド』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」の節を参照してください。

AssetCenterコネクタの詳細については、Connect-It付属マニュアル『ユーザガイド』の「Peregrine Systemsコネクタ」の章、「Asset Managementコネクタ」の節を参照してください。

【賃貸料の計算（Rent）】モジュール

AssetCenter Serverは、契約と資産の賃貸料の定期的な支払いをモニタし、支払額を計算および必要に応じて再計算します。

【賃貸料の計算】モジュールは次のオプションを定義します。

- 契約賃貸料と資産賃貸料の経費明細の作成に関連する特定のパラメータ
- 更新頻度

概要

AssetCenter Serverは、経費明細の作成が必要かどうかを定期的に検証します。必要な場合は経費明細を作成します。

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ると、AssetCenter Server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する場合は、資産の詳細画面の【取得】タブページ／【賃貸料】サブタブページ／【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しない場合は、契約の詳細画面の【賃貸料】タブページ／【賃貸料】サブタブページ／【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。

作成パラメータ

【ユーザーデータ】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：▶ マニュアル『AssetCenter - インストール』の「.ini および .cfg ファイル」の章

賃貸料の見積

【ユーザデータ】フィールドには、支払う賃貸料の見積を計算する日数を指定できます。

AssetCenter Server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、2001年7月1日から2004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。
- AssetCenter Serverでは、賃貸料の支払を2ヶ月ごとに検証し、向こう12ヶ月に支払う見積賃貸料を作成します。

AssetCenter Serverを2002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。

- 2001年7月1日から2002年7月1日までの過去の賃貸料
- 2002年7月1日現在の賃貸料
- 2002年8月1日から2003年7月1日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現在の経費明細の日付、つまり2002年7月1が表示されます。

AssetCenter Serverはバックグラウンドで動作します。AssetCenter Serverは、2ヶ月後の2002年9月1日に次のデータを作成します。

- 2002年10月1日から2003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、2002年8月1日から2002年10月1日までの賃貸料

【規定の損失額の計算（LostVal）】モジュール

AssetCenter Serverは、計算方法が【全期間で計算】（リース契約の詳細画面の【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契約の損失額を定期的に再計算します。このように、AssetCenter Server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、AssetCenter Serverによって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

必要条件

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。

- AssetCenterを起動します。
- 【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／AssetCenter Server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

AssetCenterではなくAssetCenter Serverを使ってこのタスクを実行すると、注文品の受領を行うクライアント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、本モジュールを毎日、数回実行することをお勧めします。

【実行グループにワークフロー規則を適用】 モジュール

AssetCenter Server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

AssetCenter Server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AssetCenter Server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／【制限時間】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

AssetCenter Serverは、イベントの詳細画面の【パラメータ】タブページで定義した周期と起動条件で**定期的なイベント**をトリガします。

AssetCenter Server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、AssetCenter ServerはこれをamWfOccurEvent (SQL名) テーブルに保存します。
その後、AssetCenter Server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、AssetCenter Server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、AssetCenter Serverはトランジションを起動します。

トランジションの起動

AssetCenter Serverは、設定画面で定義した周期に従ってイベントのトランジションを起動します。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

AssetCenter Server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。

AssetCenter Serverがタスクを検証して実行する周期は、AssetCenter Server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。

- AssetCenter Server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、AssetCenter Serverがそのタスクを実行します。
- それ以外の場合は、AssetCenterクライアントマシンでタスクを実行します。

【スキャナの結果を基にデータベースを更新 (IddAc)】 モジュール

AssetCenter Serverでは、InfraTools Desktop Discoveryが生成する「.fsf」ファイルを取得するようプログラムできます。「.fsf」ファイルは、コンピュータのスキャン結果を格納するファイルです。

「.fsf」ファイルを格納するフォルダは、Connect-Itの「iddac.scn」シナリオで指定されています。

注意:

このモジュールは、コンピュータのスキャンが実行済みであることを前提にしています。

【スキャナの結果を基にデータベースを更新】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。

- AssetCenter Serverの**【スキャナの結果を基にデータベースを更新】**モジュールのパラメータ
- Connect-Itの「iddac.scn」シナリオのパラメータ（Connect-Itのインストール先フォルダの「scenario\idd\iddac40」サブフォルダ内に格納されています）

【スキャナの結果を基に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以下のとおりです。


```
"$connectit_exedir$/conitsvc.exe" -once '$connectit_exedir$../scenario/idd/iddac40/iddac.scn'
-d:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$ -d: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ことはお控えください。
- -onceは、「conitsvc.exe」を1度のみ実行することを意味します（Connect-Itの「Once」スケジューラを使用する）。
このパラメータは、変更しないでください。AssetCenter Serverでスケジューラを定義します。
- \$connectit_exedir\$../scenario/idd/iddac40/iddac.scnは、使用するConnect-Itシナリオへのアクセスパスです。
別の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:AssetCenter.SERVER=\$cnx\$ -d:AssetCenter.LOGIN=\$login\$
-d:AssetCenter.TEXTPASSWORD=\$pwd\$は、AssetCenter Serverが開くAssetCenter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、「iddac.scn」シナリオのAssetCenter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。

Connect-Itの「iddac.scn」シナリオのパラメータ

「iddac.scn」シナリオを変更するには、次の手順に従います。

- 1 Connect-Itのシナリオビルダを実行します。
- 2 「iddac.scn」シナリオを開きます（Connect-Itのインストール先フォルダの「scenario\idd\iddac40」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InfraTools Desktop Discoveryボックスのタイトルバーをクリックして、InfraTools Desktop 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 [次へ] をクリックします。
- 6 以下のフィールドに入力します。
 - **[FSFファイルフォルダ]**：取得する「.fsf」ファイル（コンピュータのスキャンファイル）を格納するフォルダのパス
このフォルダは、「getfsf.scn」シナリオのマッピングで指定されたフォルダと同一でなければなりません。
 - **[SAIファイルフォルダ]**：「.sai」ファイル（スキャナがコンピュータ上で検索し識別するアプリケーションをリストにして記述ファイル）を格納するフォルダのパス

このフォルダは、「getfsf.scn」シナリオのマッピングで指定されたフォルダと同一でなければなりません。

- **【ユーザ追加項目ファイル】**：「.cdt」ファイル（InfraTools Desktop Discoveryのスキャンフォームに含まれているフィールドのリストで、これらのフィールドはConnect-Itのマッピングに含まれます）の完全パス

AssetCenterデータベースにどのデータが入力されるかを知るには、「iddac.scn」シナリオの詳細を確認します。

モジュールは特に、**【コンピュータ】**（SQL名：amComputer）テーブルの**【前回のハードウェアスキャン】**（SQL名：dtHardScan）フィールド、**【前回のネットワークスキャン】**（SQL名：dtNetworkScan）フィールド、**【前回のソフトウェアスキャン】**（SQL名：dtSoftScan）フィールドに入力します。

InfraTools Desktop Discoveryコネクタの詳細については、Connect-It付属マニュアル『ユーザガイド』の「Peregrine Systemsコネクタ」の章、「InfraTools Desktop Discoveryコネクタ」の節を参照してください。

AssetCenterコネクタの詳細については、Connect-It付属マニュアル『ユーザガイド』の「Peregrine Systemsコネクタ」の章、「Asset Managementコネクタ」の節を参照してください。

シナリオ「iddac.scn」に関する文書を作成するには、Connect-Itの『ユーザガイド』マニュアルの「シナリオ文書」の章、「シナリオ文書の作成」の節を参照してください。

【スキャナの結果を基にデータベースを更新 (PdiAc)】モジュール

AssetCenter Serverでは、Desktop Inventoryが生成する「.xml.gz」ファイルを取得するようプログラムできます。「.xml.gz」ファイルは、コンピュータのスキャン結果を格納するファイルです。

「.xml.gz」ファイルを格納するフォルダは、Connect-Itの「addcpu.scn」シナリオで指定されています。

【スキャナの結果を基にデータベースを更新】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。

- AssetCenter Serverの**【スキャナの結果を基にデータベースを更新 (PdiAc)】**モジュールのパラメータ
- Connect-Itの「addcpu.scn」シナリオのパラメータ（Connect-Itのインストール先フォルダの「scenario\pdi\pdiac41」サブフォルダ内に格納されています）

[スキャナの結果を基にデータベースを更新 (PdiAc)] 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以下のとおりです。

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/pdi/PdiAc41/PdiAc.scn'
-d:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$ -d: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ことはお控えください。
- `-once`は、「conitsvc.exe」を1度のみ実行することを意味します（Connect-Itの「Once」スケジューラを使用する）。
このパラメータは、変更しないでください。AssetCenter Serverでスケジュールを定義します。
- `$connectit_exedir$/../scenario/pdi/PdiAc41/PdiAc.scn`は、使用するConnect-Itシナリオへのアクセスパスです。
別の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:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$ -d:AssetCenter.TEXTPASSWORD=$pwd$`は、AssetCenter Serverが開くAssetCenter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、「PdiAc.scn」シナリオのAssetCenter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ことはお控えください。

Connect-Itの「PdiAc.scn」シナリオのパラメータ

「PdiAc.scn」シナリオを変更するには、次の手順に従います。

- Connect-Itのシナリオビルダを実行します。
- 「PdiAc.scn」シナリオを開きます（Connect-Itのインストール先フォルダの「scenario\pdi\pdiac41\PdiAc.scn」サブフォルダ内に格納されています）。
- シナリオ図の画面でPDIボックスのタイトルバーをクリックして、PDIコネクタを選択します。
- **[ツール／設定]** メニューを選択します。
- **[次へ]** をクリックします。
- 処理モードを選択します（読み取りまたは書き込み）。

- **【次へ】** をクリックします。
- 接続プロトコルを選択します。デフォルトのプロトコルは**【ローカルファイル／ネットワークファイル】** です。
- **【次へ】** をクリックします。
- ファイルの場所を指定するか、または**【フォルダ名】** フィールドでスキャンフォルダの場所を指定します。

AssetCenterデータベースにどのデータが入力されるかを知るには、**「PdiAc.scn」** シナリオの詳細を確認します。

【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。

AssetCenter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 (AssetCenter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（AssetCenter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

AssetCenter Server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、AssetCenter Server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AssetCenter Server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

AssetCenter Serverは、データベースサーバにAssetCenter Serverが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AssetCenter Serverから1時間以上信号を受信しない場合は、ユーザがAssetCenter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、AssetCenter Server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AssetCenter Server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

AssetCenter Serverで、経費明細を分割することができます。

概要

AssetCenter Server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。

AssetCenter Server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

AssetCenter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、【ツール／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します。
- 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、AssetCenterは次の項目を削除します

- 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、AssetCenter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AssetCenter Serverが検出した場合は、AssetCenter Server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AssetCenter Serverは分割前の元の経費明細も削除します。

次に、AssetCenter Server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、クライアント用のタイムゾーンを指定した場合（[管理/タイムゾーン]）に使用されます。

[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]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】セクション）にリンクできます。

消耗品のアラーム

AssetCenter Server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、次の手順に従います。

- 1 AssetCenterを起動します。
 - 2 【調達／購入依頼】メニューを選択します。
 - 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

AssetCenter Server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】の場合：アラームはリース終了の通知日にリンクできます。これらの日付は【オプション】タブページの期間満了オプション（更新、購入、返却）の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：アラームは個々の【賃貸料】タブのサブタブページに表示される【契約期間】の最終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／【終了】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、AssetCenter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AssetCenter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。

【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。

AssetCenter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。AssetCenter Server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

AssetCenter Serverは、発注点（再発注時の在庫数）をモニタします。

AssetCenter Server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。

- AssetCenter Serverは、ポートフォリオ品目の詳細画面の【割当】フィールドの値を元にして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、AssetCenter Server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。
- 購入依頼した物件が完全に受領されるまでは、AssetCenter Server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、AssetCenter Server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。

手動で検証モジュールを起動する

定義した時間を待たずに、手動でAssetCenter Serverの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、AssetCenter Server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。

AssetCenter ServerをWebブラウザで管理する

Windows NT用のAssetCenterインストールプログラムは、AssetCenter Server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、AssetCenter Serverを次の2つの方法で起動できます。

- AssetCenter Serverのグラフィカルユーザインタフェースを使って起動する。
- NTサービスとしてAssetCenter Serverを起動する。

AssetCenter Server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。

ここでは、次の内容について説明します。

- AssetCenter ServerをNTサービスとして起動する
- Webブラウザ経由でAssetCenter Serverサービスにアクセスする
- WebからAssetCenter Serverサービスを制御する

AssetCenter Serverをサービスとして起動する

Windows NTにAssetCenterをインストールすると、次のようになります。

- AssetCenter Server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。
- AssetCenter ServerサービスへのWebアクセスは無効です。

注意:

AssetCenter Server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 サービスのインストール先コンピュータ上に、Windows NTのユーザアカウントを作成します。

このアカウントには、AssetCenter Server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、AssetCenter Server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 AssetCenter Serverサービスをこのアカウント上にインストールします。
-

**警告:**

AssetCenter Serverサービスの接続先のデータベースのDBMSクライアントレイヤをインストールした後、AssetCenter Serverがインストールされたコンピュータを、再起動しなければなりません。これは、Windowsのサービスコントロールマネージャが、コンピュータの起動時に環境変数PATHを読み取るためです。最新のPATH環境変数は、AssetCenter ServerがDBMSのDLLを検索するために必要になっています。これは特にSQL Anywhereランタイムをインストールし、AssetCenter Serverをデモ用データベース（SQL Anywhereを使用）上で実行する場合に注意が必要です。

AssetCenter Server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、次の手順に従います。

- 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：▶ マニュアル『AssetCenter - インスタレーション』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、AssetCenter Server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。

AssetCenter Serverサービスを起動する

AssetCenter Server NTサービスを起動するには、次の手順に従います。

- 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 AssetCenter Server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。AssetCenter Server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。

AssetCenter Server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 NTを起動すると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AssetCenter Serverサービスにアクセスする



注意:

注意: AssetCenter Server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。

AssetCenter Serverサービスにアクセスするには、次の手順に従います。

- 1 普段使用しているWebブラウザを起動します。
- 2 AssetCenter Server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、AssetCenter Server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"

AssetCenter Server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 AssetCenter Server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。このパスワードのデフォルトは空です。

WebからAssetCenter Serverサービスを制御する

ここでは、AssetCenter Server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。

- AssetCenter Serverデータベースに手動で接続します。
 - AssetCenter Serverの起動時に自動的にAssetCenterデータベースに接続するように設定します。この手順を次に示します。
 - 1 **【起動時に再接続】** オプションをチェックします。
 - 2 AssetCenter Server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、AssetCenter Server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：▶ マニュアル『AssetCenter - インスタレーション』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、AssetCenter Server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AssetCenter Server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を変更できません。

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、AssetCenter Server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。

起動

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、AssetCenter Server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分に設定されます。

このファイルの場所：▶ マニュアル『AssetCenter - インストール』の「.ini および .cfg ファイル」の章

14 | メッセージシステム

AssetCenterでは、次の2種類のメッセージを管理できます。

- AssetCenterから発信され、内部メッセージシステムを介してAssetCenterデータベースに送信されたメッセージ
- AssetCenter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

本章では、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。

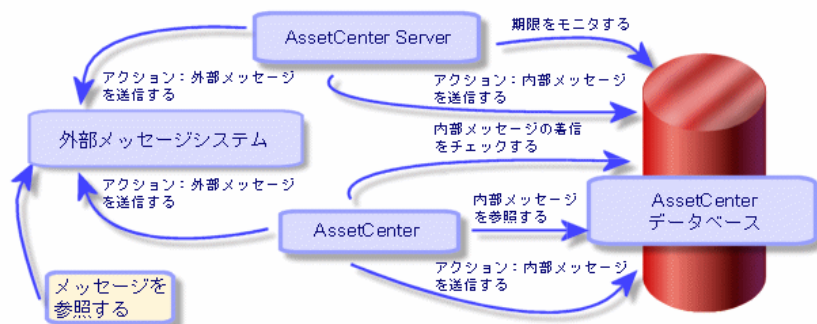
メッセージの送受信

AssetCenter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 (AssetCenter)
- SMTP
- MAPI
- VIM

受信では、AssetCenterはAM（AssetCenter）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、マニュアル『AssetCenterの高度な使い方』の「メッセージ」の章を参照してください。

AssetCenter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AssetCenterの設定を行います。

AssetCenterとAssetCenter Server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。
- AssetCenter Server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。

AssetCenterの【ツール／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:

AssetCenter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「AssetCenter (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:[andreoucassidou@peregrine.com], SMTP:[administrator@prgn.com]

SMTPプロトコル

従業員の詳細

AssetCenterで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AssetCenterで指定しておく必要があります。

- 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。

The screenshot shows a user profile form with the following details:

- 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

The 'Message' tab is selected, showing the following fields:

- アカウント:** SMTP:honda
- パスワード:** *****

- 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]

- 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）。

SMTP:[name@address.domain]

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。メッセージのコピーが管理者に送信されます。

win.iniファイル(Windows)またはmail.ini (Unix)

AssetCenterでメッセージングシステムを使用するには、以下のコマンドラインを追加してください。

- Windowsの場合、「win.ini」ファイルで以下を実行します。
- UNIXの場合、「mail.ini」ファイルを作成し、以下を実行します。

```
[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
```

上記のファイルの場所: マニュアル▶『AssetCenter - インスタレーション』の「.ini および .cfg ファイル」の章

MAPIプロトコル

従業員の詳細

AssetCenterで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AssetCenterで指定しておく必要があります。

- 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]
```

ユーザプロファイルを確認するには、Windowsのコントロールパネルで、【メール】セクションを開き、【プロファイルの表示】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]

- 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。

win.iniファイル(Windows)またはmail.ini (Unix)

[mail]セクションに以下の行を追加します。

- Windowsの場合、「win.ini」ファイルで以下を実行します。
- UNIXの場合、「mail.ini」ファイルを作成し、以下を実行します。

```
MAPI=1
MAPIX=1
```

上記のファイルの場所: マニュアル▶『AssetCenter - インストール』の「.ini および .cfg ファイル」の章

VIMプロトコル

従業員の詳細

AssetCenterで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AssetCenterで指定しておく必要があります。

- 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

メッセージ
 アカウント: VIM:kanada/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]

全般 研修 ポートフォリオ コスト プロジェクト プロファイル メッセージ
 住所
 所在地: /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: VIM:kanada/TALTEK
 フィールド1:
 フィールド2:
 ID番号: DEMO-M041 コストセンタ: 統括部
 雇用日: 1995/12/21 バーコード: DEMO-U005
 退職日:
 フィールド3:
 コメント:

- 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

AssetCenter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 2000

- 「vim32.dll」ファイルへのアクセスパスを書き留めます。
- マイコンピュータのアイコンを右クリックして、コンピュータのプロパティを編集します。
- 【詳細設定】タブで【環境変数】ボタンをクリックします。
- 【システム環境変数】枠で、変数「Path」を編集し、Lotus Notesのパスを追加します。

Windows NT4

- Lotusフォルダの「vim32.dll」ファイルへのアクセスパスを書き留めます。
- コンピュータのアイコンを右クリックして、コンピュータのプロパティを編集します。

- **【環境】** タブで、変数「Path」を編集し、Lotus Notesのパスを追加します。

Windows 95, 98, ME

- Lotusフォルダの「vim32.dll」ファイルへのアクセスパスを書き留めます。
- ハードドライブのルートに位置している「autoexec.bat」ファイルを編集します。
- 例

```
SET PATH=[Lotus Notesのパス]
```

Lotus Notesフォルダの「vim32」ファイルへのアクセスパスを指定します。

win.iniファイル(Windows)またはmail.ini (Unix)

[mail]セクションに以下の行を追加します。

- Windowsの場合、「win.ini」ファイルで以下を実行します。
- UNIXの場合、「mail.ini」ファイルを作成し、以下を実行します。

```
SMI=1
```

上記のファイルの場所: マニュアル▶『AssetCenter - インストール』の「.ini および .cfg ファイル」の章

AMプロトコル

- 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。

- **【プロフィール】 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]

- 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

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。

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 AssetCenter Server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と、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「win.ini」ファイルの問題が明らかになります。

AssetCenterでMAPIメッセージシステムを正しく使うには、「win.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

AssetCenterでVIMメッセージシステムを正しく使うには、「win.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

AssetCenterでSMTPメッセージシステムを正しく使うには、「win.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、「win.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、AssetCenter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AssetCenter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。

「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（AssetCenter Database Administratorを使って追加する）
- ウィザード（AssetCenter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

AssetCenter Database Administrator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 AssetCenter Database Administrator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、編集ゾーン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]|CompleteDate
s=[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 [修献 309]

- 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 テーブルの詳細画面（**【表示／詳細】**メニュー）を表示します。
- 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 AssetCenter Database Administrator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】** オプション）。
- 3 **【作業指示】**（SQL名：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】** タブのフィールドに以下のデータを入力します。

フィールド	値
SQL名	PgIn_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、編集ゾーン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	結果
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	【満了予定日】 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。

パラメータ	値	結果
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□に、色が自動的に循環して割り当てられます。
ItemColor	lTechId	
CompleteDates	true	【作業開始日】（SQL名：dtActualFixStart）フィールドまたは【作業完了日】（SQL名：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】（SQL名：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。
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□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 テーブルの詳細画面（【表示／詳細】メニュー）を表示します。
- 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「PgIn_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 AssetCenter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】オプション）。

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（【ポートフォリオ／作業指示】）。

20 作業指示を2つ作成し、以下のフィールドにデータを入力します。

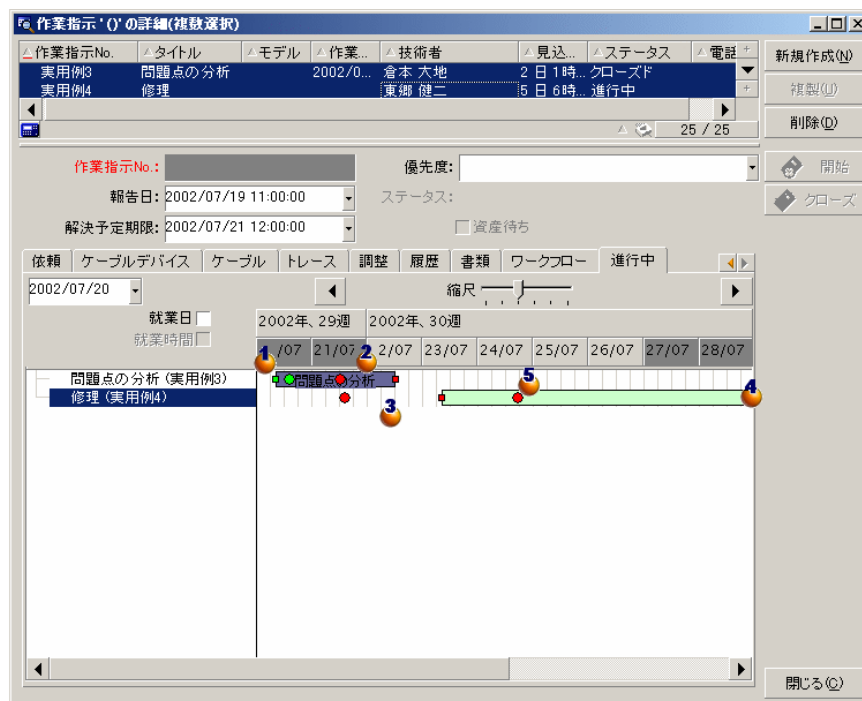
タブ	フィールドの 名前	フィールドの SQL名	作業指示1の値	作業指示2の 値
	作業指示No.	WONo	CP3	CP4
	報告日	dtNotif	2002/07/19 11:00:00	2002/07/19 11:00:00
全般	タイプ	seType	内部メンテナンス	内部メンテナ ンス
全般	タイトル	Title	問題点の分析	修理
トラッキング ／技術者	技術者	Contact	技術者を1人選択してくださ い。	技術者をもう 1人選択して ください。
トラッキング ／スケジュール	開始予定日	dtSchedFixStart	2002/07/20 14:00:00	2002/07/21 14:00:00
トラッキング ／スケジュール	終了予定日	dtSchedFixed	2002/07/21 12:00:00	2002/07/24 17:00:00
トラッキング ／スケジュール	作業開始日	dtActualFixStart	2002/07/20 08:00:00	2002/07/23 08:00:00
トラッキング ／スケジュール	作業完了日	dtActualFixed	2002/07/22 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

AssetCenter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 AssetCenter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】オプション）。
- 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】）。
- 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】（SQL名：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、AssetCenter 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 [修献 309]

- `strIds`は、ウィザードが戻すレコードのイベントを表示する場合、これらのレコード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 AssetCenter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] オプション）。
- 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]）。
- 4 新規ウィザードを作成します。
- 5 以下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] ポップアップ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=IWorkOrderId|StartDate=dtActualFixStart|EndDate=dtActualFixed|EndBefore=dtSchedFixed|StartBefore=dtSchedFixStart|ItemColor=ITechId|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

本節では、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（AssetCenter Database Administrator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](#) [参考文献 302]

- ウィザードのページ（AssetCenter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](#) [参考文献 302]

PlannerViewer-[A]（テーブル）またはMainField=[A]（ウィザード）

用途
値

表示するイベントを指定します。

- イベントが直接、テーブルの一部を成す場合

<テーブルの主キー>

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以下の通りです。

- 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、

<リンク1のSQL名>.<リンク2のSQL名>...<リンクNのSQL名>

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、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、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	以下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 ・ <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】（SQL名：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	以下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 ・ <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】（SQL名：amWorkOrder）テーブルの【満了予定日】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	以下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 ・ <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】（SQL名：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。
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	以下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 ・ <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】（SQL名：amWorkOrder）テーブルの【満了予定日】フィールドでは、[E] はdtSchedFixedになります。
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に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	以下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 ・ <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】（SQL名：amWorkOrder）テーブルの【報告日】フィールドでは、[F] はdtNotifになります。
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	以下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 ・ <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 ☐ 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	• 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 ☐ 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] がtrueである場合の値	[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。 [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> • <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 [H] の形式は次のうちの1つになります。
 値

- ・ <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、本マニュアルの「AssetCenterを初めて使用する」の章の「レコードリスト」の節、「メインリストを設定する」／「[リストの設定] ポップアップメニューを使う」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使用されるタイトルを識別します。
----	--

- 値
- **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、リンクN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。

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

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
- **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。

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 | <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:

AssetCenter Database Administrator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	・ <テーブルの直接フィールドのSQL> ・ <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】 (SQL名: 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 フィールドは以下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> リンクは以下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】（SQL名：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 | AssetCenterをDDEサーバとして使用する

ここでは、AssetCenterをDDEサーバとして使う場合に、AssetCenterによって認識される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AssetCenter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。
ここでは、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。AssetCenter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。AssetCenterの場合、このトピックは「AssetCenter」です。

DDEコマンド

DDEコマンドは、AssetCenter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。

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）：AssetCenterでタスクを実行できます。
- Request（要求）：AssetCenter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、ポップアップ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。AssetCenterをDDEサーバとして使う場合、「トピック」は常に「AssetCenter」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。

- AssetCenterが受信できるDDEコマンドの一覧。これらのコマンドのシンタックスについて詳しく説明します。
- DDEメカニズムによるAssetCenterの駆動例。これらのコマンドの使い方を説明します。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。

本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

説明

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。

- AssetCenter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：▶ マニュアル『AssetCenter - インストール』の「.ini および .cfg ファイル」の章
- 下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AssetCenter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、AssetCenter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、AssetCenter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

説明

AssetCenter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、AssetCenter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

説明

<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、AssetCenter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

Request

説明

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>

このパラメータでは、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

Request

説明

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

説明

AssetCenterで定義されたビューを開きます。

<ViewName>

このパラメータには、AssetCenter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。

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

説明

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

説明

現在AssetCenter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

Request

説明

<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

説明

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

説明

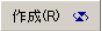
開いているテーブルのレコードとの対話モードを定義します。

<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：テーブル<Table>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：テーブル<Table>で選択したレコードを破棄します。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  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)
Execute:amAsset.SetRecordMode(Cancel)
```

<Table>.ListAllFields([Mask])

アクションのタイプ

Request

説明

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

Request

説明

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】** 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

説明

<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

説明

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

Request

説明

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。

次のコマンドについて説明します。

- Table-Objet.AddLink()
- Table-Objet.GetValue()
- Table-Objet.Highlight()
- Table-Objet.RemoveLink()
- Table-Objet.SetFocus()
- Table-Objet.SetValue(Value)
- Table-Link.SetValueWhere(Condition)
- Table-Objet.Show()

<Table>:<Objet>.AddLink()

アクションのタイプ

Execute

説明

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。

<Objet>

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>:<Objet>.GetValue()

アクションのタイプ

Request

説明

選択したレコードについて、テーブル<Table>の<Objet>（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。

<Objet>

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>:<Objet>.Hilight()

アクションのタイプ

Execute

説明

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。

<Objet>

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。このコマンドはフィールド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.Highlight()
```

<Table>:<Objet>.RemoveLink()

アクションのタイプ

Execute

説明

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。

<Objet>

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>:<Objet>.SetFocus()

アクションのタイプ

Execute

説明

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<Objet>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。

<Objet>

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>:<Objet>.SetValue(<Value>)

アクションのタイプ

Execute

説明

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【受領承認日】（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

説明

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、値を入力するリンクを含むテーブルの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>:<Objet>.Show()

アクションのタイプ

Execute

説明

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。

<Objet>

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()
```

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します。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: AssetCenterの内部DDEコール

この例の目的は、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。

コールに応答する

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AssetCenter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： **AssetCenter**

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)
```

AssetCenter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。

- Commande : **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、AssetCenter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。次のコマンドとパラメータを使います。

- コマンド：<テーブル>:<オブジェクト>.SetValue(<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>.SetValue(<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>.SetValue(<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2：Microsoft ExcelからのDDEコール



注意:

このシナリオはMicrosoft Excel 2002でテストされています。

この例では、Excelのワークシート「TestDDE.xls」を使います。このワークシートは、AssetCenterの「Samples\DDE\Excel」フォルダにあります。このファイルには、この例に必要なマクロが含まれています。

この例のDDEコールはVBA（Visual Basic for Applications）に準拠しています。それ以外は前の例と同じです。

ここでは、次の内容について説明します。

- マクロの指定
- マクロのソースコード

マクロの指定

TestDDE.xlsファイルの「Data_entry」ワークシートに移動します。このワークシートには、それぞれ [Name]、[City]、および [ZIP] という名前の3つの列を含む表とボタン（「Create location」というラベル付き）があり、ボタンには「Create location」と呼ばれるマクロが割り当てられています。

[Name]（この例ではセルB6）、[City]（セルC6）、[ZIP]（セルD6）フィールドに適切な値を入力し、「Create location」をクリックすると、ExcelからAssetCenterに対して次のアクションを実行する命令が発信されます。

- 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。

マクロのソースコード

参考のため、DDEコールを実行するマクロのソースコードを以下に示します。ExcelまたはWordでのDDEコールおよびVBA言語のシンタックスの詳細については、それぞれのオンラインマニュアルを参照してください。

```
Sub CreateLocation()  
  
Dim Name As String
```

```

Dim City As String
Dim ZIP As String

Name = Worksheets("Data_entry").Range("B6")
City = Worksheets("Data_entry").Range("C6")
ZIP = Worksheets("Data_entry").Range("D6")

Contexte = Application.DDEInitiate(app:="am", topic:="AssetCenter")
Application.DDEExecute Contexte, "OpenTable(amLocation)"
Application.DDEExecute Contexte, "amLocation.SetRecordMode(New)"
Application.DDEExecute Contexte, "amLocation.Name.SetValue(" + Name + ")"
Application.DDEExecute Contexte, "amLocation.City.SetValue(" + City + ")"
Application.DDEExecute Contexte, "amLocation.ZIP.SetValue(" + ZIP + ")"
Application.DDEExecute Contexte, "amLocation.SetRecordMode(Create)"
Application.DDETerminate Contexte

End Sub

```

例3：Visual BasicからのDDEコール

この例では、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、AssetCenterを起動する必要があります。

ここでは、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のマニュアルを参照してください。

推奨事項

本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 AssetCenter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AssetCenterウィンドウの横に移動する。



注意:

これにより、AssetCenter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、**[場所]** 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]**（SQL名：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 AssetCenter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AssetCenter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、AssetCenterの特定の機能が利用できなくなります。

本章では、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ming】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かどうかを基準にして設定します。
- **【メッセージ】項目の【新規メッセージのチェック】**：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、AssetCenter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（**【ポートフォリオ／資産とロット】**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】**ポップアップメニュー）
- 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、AssetCenter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。

以下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

以下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。

以下の手順でこのようなオプションを表示します。

- 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] オプション
- ・ [リスト／その他のリスト／読み込む項目数の限度] オプション
- ・ [リスト／メインリスト／読み込み時間の限度] オプション
- ・ [リスト／メインリスト／読み込む項目数の限度] オプション

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。

インタフェースオプションに関する詳細は、マニュアル『はじめに』の「ユーザのコンピュータでAssetCenterをカスタマイズする」の章を参照してください。

AssetCenterクライアントでリストのパラメータを設定する

「FetchingArraySize」パラメータが、AssetCenterのサポートするすべてのDBMSにより使用されます。

DBMSは、AssetCenter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AssetCenterクライアントで、AssetCenter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：▶ マニュアル『AssetCenter - インストール』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。

「FetchingArraySize」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] オプション
- ・ [リスト／メインリスト／読み込む項目数の限度] オプション

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」としてします。
- この場合、AssetCenter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。

以下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 pingを使ってこのシナリオを試行したところ、レコード200個の表示時間が1.5秒短縮されました。

アプリケーション例

- **「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：▶ マニュアル『AssetCenter - インストール』の「.ini および .cfg ファイル」の章
- 2 [<最適化するAssetCenter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、セクションに次のように完全項目を追加します：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：▶ マニュアル『AssetCenter - インスタレーション』の「.ini および .cfg ファイル」の章

18 | AutoCADの統合

AutoCAD／AssetCenter統合の概要

AssetCenterのデータとAutoCADのデータを相互に動的にリンク、トラッキングすることが可能です。

Connect-Itシナリオを用いて、データのマイグレーション／トラッキングを行います。



注意:

Connect-Itシナリオはデータ構成の1例であり、データを適切に複製するためにはシナリオのパラメータを設定する必要があります。

AssetCenterのAutoCAD機能によって、AutoCADに新たにメニューが2つ追加されました。新しいメニューを使って、AssetCenter: **Draw Tools**およびAssetCenterの場所、従業員および関連資産をインポート／表示できます。

インストール

AssetCenter AutoCADモジュールのインストール方法についてはAssetCenter マニュアル『インストール』の「Windowsでの設定」の章の「AutoCADとの統合」のセクションを参照してください。

設定

AssetCenterに付属のConnect-Itシナリオは、
「Peregrine/AssetCenter/acadi/scenario」フォルダに格納されています。
「acadi.scn」シナリオを使用するには、AutoCADコネクタとAssetCenterを設定する必要があります。

AssetCenterコネクタを設定する

AssetCenterコネクタを設定するは、以下の手順に従います。

- 1 「acadi.scn」シナリオを開きます。
- 2 【シナリオ図】の画面で【AssetCenter】コネクタを右クリックして、表示されるショートカットメニューから【コネクタを設定する】を選択します。
- 3 表示されるウィザードで、必要に応じて【名前】フィールドと【説明】フィールドに入力し、【次へ】をクリックします。
- 4 AssetCenterデータベースを選択し（【接続】フィールド）、ODBC接続の【ログイン】フィールドと【パスワード】フィールドに入力してから、【次へ】をクリックします。
- 5 【終了】をクリックします。

AutoCADコネクタを設定する

AutoCADコネクタを設定するには、次の手順に従います。

- 1 【シナリオ図】の画面でAutoCADコネクタを右クリックして、表示されるショートカットメニューから【コネクタを設定する】を選択します。
- 2 表示されるウィザードで、必要に応じて【名前】フィールドと【説明】フィールドに入力し、【次へ】をクリックします。
- 3 AutoCADと互換性のあるAssetCenterデータベース接続タイプに対してAssetCenterを選択し、【次へ】をクリックします。
- 4 AutoCADと互換性のあるAssetCenterデータベースを選択し（【接続】フィールド）、【ログイン】フィールドと【パスワード】フィールドに入力してから、【次へ】をクリックします。

- 5 **【終了】** をクリックします。

AutoCAD統合を使用する

インタフェース

AssetCenterがAutoCADモジュールをインストールすると、AutoCADに**【DrawTools】** メニューと **【AssetCenter】** メニューが追加されます。

概要

AssetCenterまたはAutoCADを使用する前に、Connect-Itを使ってデータ同期の設定を実行する必要があります。

Connect-Itを使ってデータを同期する

Connect-Itの「acadi.scn」シナリオを使用すると、AssetCenterとAutoCAD間におけるデータの同期／マイグレーションを実行できます。

このシナリオでは、AssetCenter／AutoCAD間でデータの複製を行います。Connect-Itのスケジュールを使って、このシナリオを定期的に起動します。例えば、毎週金曜日にAssetCenter／AutoCAD間でデータの複製を行います。詳しくは、Connect-It付属の『**ユーザガイド**』を参照してください。

AssetCenterのデータをAutoCADに複製して初めて、AutoCADでAssetCenterのデータを使用できるようになります（AssetCenterコネクタの**【生成する】**ポップアップメニュー）。

AutoCADでAssetCenterのデータを使用する

AutoCADと互換性のあるAssetCenterのデータベースにアクセスするには、以下の手順に従います。

- 1 続行する前にグラフィックを保存します（**【ファイル／名前を付けて保存】**）。
- 2 **【AssetCenter／Property Portfolio】** メニューを選択します。
- 3 **【グラフィック管理】** から **【プロパティナビゲータ】** を選択します。
- 4 **【プロファイル】** ドロップダウンリストからAssetCenterデータベースを選択します。

 ヒント:

AutoCADでエラー37000が表示されても、無視してかまいません。

- 5 場所を選択してから、**【Attach】** をクリックします。

 ヒント:

場所のリストはツリー構造になっています。ただし、直接には表示されていない場合もあります。場所をダブルクリックすると下のレベルが展開されます。

【フロア】 レベルを1つだけ選択して **【関連付け】** をクリックします。

- 6 AutoCADが、データベースに接続されたことを通知してきたら、選択されたマップで操作を開始できます。

AutoCADの使用時に変更を行い、これを保存した場合、AssetCenterを使用する前にConnect-Itシナリオを起動してデータを同期します（AutoCADコネクタの**【生成する】** ポップアップメニュー）。

Conenct-Itシナリオ用のデータ構成

Conenct-Itシナリオを正しく機能させるためには、

- 以下のように場所を分類する必要があります。
 - Campus
 - Building
 - Floor
 - Room/ Sublocation
- **【LocationType】** フィールドに、次のキーワードを入力します：Campus、Building、Floor、Room/Sublocation
- 資産と従業員は、**Room/Sublocation**のレベルで入力されている値に該当します。
- 資産の条件は以下のとおりです。
 - Nature.seBasis = portfolio item
 - Nature.seMgtConstraint = unique asset tag
 - Nature.bSoftLicense = 0

 注意:

シナリオは、完全にカスタマイズ可能です。

シナリオをカスタマイズする場合は、ペレグリンシステムズのカスタマサポートまでお問い合わせください。

AssetCenterでAutoCADのデータを使用する

Connect-ItへのAutoCADデータのマイグレーションを行うと、【従業員】テーブル、【場所】テーブル、および【資産】テーブルが、AutoCAD起源のデータによって記入／更新されます。

以下のデータが入力されます。

- 場所の詳細画面の【全般】タブページと【ポートフォリオ】タブページ
各階のAutoCAD図面は、場所の階の【書類】タブに表示されます。
- 従業員の詳細画面の【全般】タブページと【ポートフォリオ】タブページ
- 資産の詳細画面の【全般】タブページと【ポートフォリオ】タブページ



注意:

AutoCADモジュールは資産と従業員に関するデータを視覚化します。これ以外のデータを視覚化するには、専用Connect-Itシナリオを作成するか、またはペレグリンシステムズのカスタマサポートへ連絡する必要があります。

インデックス

目次

- & (特殊文字) , 46
- < (特殊文字) , 46
- アイコン, 349
- アクション
 - アクションボタン - 作成, 135
 - 自動トリガ (参考 AssetCenter Server)
- アクションボタン, 135
- 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, 227-245
 - (参考 パスワード)
 - オプション, 185
 - ユーザ, 236
 - ユーザ接続, 238
 - 概要, 228
 - 制限, 227
 - 編集, 231
 - 有効性, 228
- アクセス - 固定, 238
- アクセス制限, 234
 - リンク - デフォルト値, 236 , 236
 - 低速ネットワーク - チューニング, 352
 - 定義, 230
 - 例外, 43
- アラームの検証 (モジュール) , 278
- インデックス
 - SQL名, 40
 - カスタマイズ, 126 , 42
 - ラベル, 40
 - 数値, 50
 - 説明, 40 , 40
 - 挿入, 132
- インデックス (インスタンス) , 100
- インポート, 189-226
 - ODBC - データベースの選択, 197
 - インポートするフィールド
 - マッピング, 205 , 203
 - 説明, 201
 - キー, 206
 - スクリプト, 224 , 222
 - ソーステーブル - マッピング, 204
 - テーブル - 分析の説明, 197
 - テキストファイル
 - マッピング, 204
 - 構造, 196
 - テキストファイル - 選択, 197
 - データベース, 190
 - データベース構造とデータ, 71
 - パフォーマンス, 219
 - ファイル, 197
 - フォルダ - 作成, 220
 - プロセス, 195
 - リンク - 作成パラメータ, 207
 - 概要, 190
 - 推奨事項, 191
 - 数値データ, 191
 - 制限, 192
 - 単一のテキストファイル, 190
 - 転送 - 設定, 219

- 特殊フィールド, 210
- 部署と従業員, 213
- 複数のテキストファイル, 190
- 文字 - ISOコード, 190
- 文書, 214
- インポート - オプション, 182
- インポート中の履歴を無効にする (オプション), 182
- ウィザード - オプション, 182
- OWNコピー (リンクタイプ), 33
- OWN (リンクタイプ), 33
- オーバーフロー, 36
 - テーブル, 36
 - リンクの動作, 37
- オーバーフローテーブル, 36
 - 実装, 38
- オーバーフロー (リンクタイプ)
 - リンクタイプ, 33
- オブジェクトの設定 (メニュー), 41
- オブジェクト (入力タイプ), 30
- カウンタ
 - デフォルト値, 44
 - 名前 - 特殊文字, 44
- カウンタ名, 44
- カスタムリストデータ (入力タイプ), 28, 28
- キー
 - インポート, 206
 - 例, 214
- キャッシュ, 352
- キャッシュ (Oracle), 52
- コストセンタの経費を分割する (モジュール), 276
- コピー (リンクタイプ), 33
- コメント, 79
- コメント (リンクが記憶する情報のタイプ), 34, 34
- この接続をサービスモードで使用 (オプション), 250, 249
- サーバとローカルマシンの時刻の比較 (オプション), 258
- サーバ: 基準となるGMTタイムゾーンとの時間差 (オプション), 186
- サイズ (パラメータ), 126
- サイズ (プロパティ)
 - 制限, 195
- システムデータ
 - オプション, 185
 - データベース - 作成, 63
- システムリストデータ (入力タイプ), 30, 28, 28
- スキャナの結果を基にデータベースを更新 (シナリオ), 274
- スキャナの結果を基にデータベースを更新 (モジュール), 273, 271, 271
- スクリプト
 - オーバーフローリンク, 37
- スクリプトのインポート
 - コマンドラインからの実行, 224
 - 保存と実行, 222
- 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)
- スケジュールのグラフィック表示, 301-317
 - オプション, 301
 - 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, 307
 - テーブルに追加, 302
 - 実用例, 308, 303
 - 設定, 309
 - 概要, 301
 - 休日, 302
- スロットの更新間隔 (オプション), 185
- スロットの自動タイムアウト (オプション), 241, 185
- セキュリティ, 229
 - (参考 アクセスコントロール)
- タイプ (プロパティ)
 - データをインポートする際に守るべき制限, 194
- タイムゾーン, 63
- タイムゾーン - オプション, 186
- タイムゾーンの使用 (オプション), 63
- ツリー構造 - インポート, 212
- テーブル
 - SQL名, 24
 - オーバーフローテーブル, 36

- カスタマイズ, 124, 41
- タイトル, 24
- 記述文字列, 24
- 作成, 131
- 参照テーブル, 36
- 数値, 50
- 説明, 24, 24
- 中間テーブル, 34
- テーブルスペース (Oracle), 52
- テーブルの統計情報の更新 (モジュール), 275
- テーブルまたはフィールドSQL名 (入力タイプ), 29
- テーブルまたはフィールド名 (入力タイプ), 29
- テーブルを追加 (メニュー), 131
- テキストファイル - インポート, 196
- テキスト (データ型), 27
- テスト (参考 テスト用データベース)
- テスト用データベース, 140-144
 - カスタマイズ, 140
- テンプレート (メニュー), 74
- デフォルト値
 - アクセス制限, 43
 - カウンタ, 44
 - コメントテーブルへのリンク, 43
 - 記述 - エラー, 44
 - 特殊フィールド, 43
- デフォルト (パラメータ), 128
- デフォルト (入力タイプ), 28, 28
- データの機密性, 229
- 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 データベースのカスタマイズ)
 - (参考 フィールド)
 - (参考 リンク)
 - (参考 接続)
- DBMS - 変更, 59
- Microsoft SQL Server, 52

- UDB DB2 - クライアント, 57
- アクセスタイプ, 238
 - 一時, 239
 - 固定, 238
 - 不特定, 238
- オプション, 179-187
 - 制限, 179
 - 設定, 179
 - 設定 - ウィンドウ, 180
- カスタマイズ, 41
- システムデータ, 63
- データ
 - インポート, 71
 - 出力, 71
- 開く, 17
- 記述 - テンプレート, 78-89
 - 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
 - 改行マーク - 削除
 - 関数
 - 固定テキスト
 - 参照したコンポーネント - 数
 - 抽出
 - 並べ替え
 - 別のテンプレートを挿入する, 79
- 記述 - ファイル, 73-112
 - グローバル変数
 - パラメータ
 - 概要
- 構造
 - インポート, 71
 - 更新, 117
 - 出力, 71
 - 表示, 21
- 作成, 47-71
 - AssetCenter Database Administrator, 61
 - DB2 UDB, 55
 - DBMS, 49
 - Sybase Adaptive Server, 54

- Sybase SQL Anywhere (runtime), 58
- 空のデータベース, 48
- 事前検証, 52
- 新規データベース, 17
- 削除, 59
- 自動切断 - オプション, 185
- 修理, 113-115
- 診断, 113
- 説明, 73, 21-46
- 定義, 22
- データベースオブジェクト, 137
- データベースオプション(参考 データベース)
- データベースオプション (メニュー), 179
- データベースサーバ, 49
- データベースサーバに信号送信 (モジュール), 276, 65
- データベースサーバのタイムゾーンの検証 (オプション), 258
- データベースサーバのタイムゾーンの検証 (モジュール), 278
- 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), 262, 262
- データベースにNTユーザを追加 (モジュール), 265, 264
- データベースに言語を挿入 (メニュー), 67
- データベースに挿入可能な文書の最大サイズ (オプション), 185
- データベースのカスタマイズ, 123-140
 - アクションボタン, 135
 - インデックス, 132
 - オブジェクト, 130, 123
 - テーブル, 131, 124
 - フィールド, 132
 - フィールド、リンク、インデックス, 126
 - ページ, 138, 136
 - ボタン, 129
 - リンク, 132
 - 重要, 140
 - 詳細, 134, 128
 - 変更
 - コントロール, 140
 - バックアップ, 139
 - 言語, 139

- データベースの作成 (メニュー), 62
- データベースの診断/修復 (メニュー), 113
- データベース管理者, 237
- 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), 17
- データベース構造とデータのインポート (メニュー), 71
- データベース構造とデータの出力 (メニュー), 71
- データベース再接続間隔 (オプション), 257
- データベース設定の保存 (メニュー), 41
- データ (Microsoft SQL Server設定), 53
- データ：基準となるGMTタイムゾーンとの時間差 (オプション), 186
- ドロップダウンリストをツリー構造で表示 (オプション), 348
- ニュートラル (リンクタイプ), 33
- バイナリフィールド (データ型), 27
- パスワード
 - (参考 LDAP)
 - Adminログイン, 242
 - アクセスコントロール, 241
 - データベース - 作成, 61
 - パスワードの忘失, 243
 - 作成, 243
 - 変更, 243
 - 保存, 243
- パスワード - オプション, 183
- パスワード (フィールド), 255
- 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), 28, 28
- ビルド (オプション), 185
- ファイル (オプション), 258
- フィールド
 - (参考 フィールドとリンクのヘルプ)
 - SQL名, 25
 - カスタマイズ, 126, 41
 - タイトル, 26
 - デフォルト値, 42
 - アクセス制限, 236, 236
 - データ型と形式, 26
 - 説明, 26, 25

- 挿入, 132
- 値の固有性 - インデックス, 40
- フィールド'XXX'が'XXX'に見つかりません (エラー), 44
- フィールドとリンクのヘルプ
 - 文字参照, 46
 - 編集, 45
- フィールドの追加 (メニュー), 132
- フィルタ
 - データベース記述, 79
 - 低速ネットワーク - チューニング, 349
- フィルタ (パラメータ), 127
- フォルダの選択 (メニュー), 74
- ブール (はい/いいえ) (入力タイプ), 28
- ブール (入力タイプ), 28
- ページ
 - 作成, 136
 - 詳細への追加, 138
- ページの追加 (メニュー), 136
- ボタン, 129
- マイグレーション, 117
- メッセージシステム
 - MAPI
 - mail.ini, 293
 - win.ini, 293
 - MAPI - 従業員の設定, 291
 - SMTP
 - mail.ini, 291
 - win.ini, 291
 - 従業員 - 設定, 289
 - 設定, 289
 - VIM
 - mail.ini, 296
 - win.ini, 296
 - 従業員 - 設定, 293
 - 設定, 296, 295
- インタフェース, 287-300
- テスト, 298
- 概要, 287
- 接続 - 問題, 298
 - エラー, 299
 - 問題が発生した場合に実行するテスト, 298
- 設定, 288
- 複数の受信者, 289
- 複数プロトコルの使用, 288
- メッセージシステム再接続間隔 (オプション), 257
- メモフィールド (データ型), 27
- ユーザタイプ (フィールド), 27
- ユーザプロフィール, 231
 - 定義, 230
- ユーザ権限の詳細画面, 231
 - 定義, 230
- ライセンス
 - データベースに保存されたファイル, 184
 - データベース - 作成, 65
 - 使用可能な権限と機能, 60
 - 変更, 66
- ライセンス - オプション, 184
- ライセンスファイル (オプション), 184
- ラベル
 - テーブル, 24
 - フィールド, 26
- リース - オプション, 186
- リース - 料率 - オプション, 186
- リストデータ (入力タイプ), 30
- リストの更新 (メニュー), 74
- リストの設定 (メニュー), 348
- リンク
 - (参考 フィールドとリンクのヘルプ)
 - SQL名, 32
 - インポート, 207
 - カスタマイズ, 126, 41
 - タイトル, 32
 - タイプ, 33
 - デフォルト値, 42
 - 基数, 34
 - 状況依存リンク, 39
 - 説明, 32, 32
 - 挿入, 132
- リンクタイプ
 - 外部キー, 37
- リンクの追加 (メニュー), 132
- ワンポイント - オプション, 183
- ワンポイントのテキスト (オプション), 183

- 一時 (アクセス), 239
- 画像 (リンクが記憶する情報のタイプ), 34
- 画面, 352
- 開いているオブジェクトの数 (パラメータ), 50, 50
- 階層構造のテーブル, 133
- 外部キー, 37
- 完全名 (フィールド), 31, 31, 25
- 管理者権限 (オプション), 237, 179
- 既存のデータベースを開く (オプション), 17
- 期間 (入力タイプ), 30, 28, 28
- 期限 - モニタ (参考 AssetCenter Server)
- 機能権限, 233
 - 定義, 230
- 規定の損失額の計算 (モジュール), 268
- 金額 (入力タイプ), 28, 28
- 経費明細 - オプション, 182
- 言語
 - AssetCenter Database Administrator, 14
- 固定テキスト, 78
- 固有のインデックス
 - 削除または変更に関する注意, 21
- 更新日 (フィールド), 30
- 構造とデータの出力, 71
- 合計した任意管理項目 - オプション, 185, 185
- 最大サイズ (オプション), 258
- 在庫の検証 (モジュール), 281
- 資産リース料率の自動計算の承認 (オプション), 186
- 時刻 (データ型), 27
- 時刻 (入力タイプ), 28, 28
- 次へ (メニュー), 19
- 自動切断を有効にする (オプション), 241, 185
- 実行エラー - タイプエラー (エラー), 44
- 実行グループにワークフロー規則を適用 (モジュール), 269
- 実用例
 - スケジュールのグラフィック表示, 308, 303
- 受領 - オプション, 181
- 受領時に消耗品の経費明細を作成 (オプション), 182
- 受領品に対応する資産、消耗品などの作成 (モジュール), 269
- 従業員 - インポート, 213
- 出力イベント - オプション, 186
- 出力イベントテーブルの消去 (モジュール), 275
- 出力イベントの有効時間 (オプション), 186
- 書式 (パラメータ), 127
- 消耗品 - オプション, 182
- 詳細
 - カスタマイズ, 128
 - ページ - 追加, 138
- 詳細ウィンドウ (参考 詳細)
- 詳細の追加 (メニュー), 134
- 詳細画面 (参考 詳細)
- 新しいワークフロー実行グループの検索 (モジュール), 276
- 親リンクの追加 (メニュー), 133
- 進行ステータス (パラメータ), 316
- 整数 (16ビット) (データ型), 27
- 整数 (32ビット) (データ型), 27
- 整数 (8ビット) (データ型), 26
- 接続
 - (参考 LDAP)
 - (参考 指定)
 - (参考 接続スロット)
 - Windows NT, 61
 - 管理, 238
 - 指定, 61
- 接続スロット
 - 異常終了, 240
 - 概要, 239
 - 更新 - 遅延, 239
 - 破壊, 240
 - 手動破棄, 240
 - 非アクティブユーザ, 241
- 接続 - スロット - オプション, 185
- 専門分野データ, 64
- 前へ (メニュー), 19
- 全般 (タブ), 257
- 全文検索 - オプション, 187
- 全文検索を有効にする (オプション), 187

帯域幅 (参考 低速ネットワーク - チューニング)
 調達 - オプション, 181

賃貸料の計算 (モジュール), 266

低速ネットワーク - チューニング, 347-353

アクセス制限, 352

オプション, 347

キャッシュ, 352

リスト, 348

画面, 352

定期チェック (オプション), 348

定義

DDEコマンドサーバ, 319

アクセス制限, 230

データベース, 22

ユーザプロファイル, 230

ユーザ権限, 230

機能権限, 230

定義 (リンクタイプ), 33

統合された任意管理項目の再計算を無効にする (オプション), 185

特殊フィールド

インポート, 206

デフォルト値, 43

特殊文字, 44

読み込み時間の限度 (オプション), 348

読み込む項目数の限度 (オプション), 348

読取り専用 (パラメータ), 127

読取り専用 (プロパティ)

制限, 195

日付+時刻 (データ型), 27

日付+時刻 (入力タイプ), 28, 28

日付 (データ型), 27

日付 (入力タイプ), 28, 28

入力イベント - オプション, 186

入力イベントテーブルの消去 (モジュール), 275

入力イベントの有効時間 (オプション), 186

任意管理項目の値 (入力タイプ), 29

任意管理項目の追加 (オプション), 131

任意管理項目値, 34

任意管理項目値 (リンクが記憶する情報のタイプ), 34

任意管理項目値 (入力タイプ), 29

認証 - オプション, 184

認証にLDAPサーバを使用する (オプション), 183

倍精度 (データ型), 27

半自動入力のタイミング (オプション), 347

番号 (入力タイプ), 28, 28

必須タイプ (プロパティ), 195

必須 (パラメータ), 128

表記文字列, 24

表示言語, 67

オブジェクトタイプ, 68

データベースへのオブジェクトの追加, 139

選択, 68

置換, 68

追加, 67

不特定 - アクセス, 238

普通 (リンクが記憶する情報のタイプ), 34

普通 (リンクタイプ), 33

浮動小数点 (データ型), 27

部署 - インポート, 213

文字セット (パラメータ), 50

文字列 - 区切り文字, 200

文書

AssetCenter Database Administrator - メモリ

中の最大数, 16

AssetCenter Database Administrator - 自動読

み込み, 14

インポート, 214

文書 - オプション, 185

並べ替え

データベース - 記述

フィルタ, 79

階層リスト, 25

低速ネットワーク - チューニング, 349

無関連 (パラメータ), 127

有効 (パラメータ), 125

履歴 - オプション, 182

履歴項目の検証 (モジュール), 281

履歴 (パラメータ), 126

履歴 (リンクが記憶する情報のタイプ), 34
 列

インポート - 区切り文字, 200

低速ネットワーク - チューニング, 349
 連結, 210
 連続するスペース (特殊文字), 46

A

acadi.scn, 357, 356
 adbc.dll, 119
 addcpu.scn (シナリオ), 265, 263
 additional netmem (Sybase Adaptive Server), 54
 AddLink (DDE), 334
 Admin (ログイン), 242, 13
 alias (キーワード), 81
 amdb.ini, 353, 350
 amimpl.exe, 224
 amsrvf.ini, 283
 AM (メッセージシステム), 296
 Anonymous (アクセス), 244
 API, 21
 APP_CTL_HEAP_SZ (UDB DB2), 56
 AppBuild (プロパティ), 90
 AppInfo (プロパティ), 90
 AppLanguage (プロパティ), 90
 APPLHEAPSZ (UDB DB2), 56
 AppVersion (プロパティ), 90
 AssetCenter Database Administrator (参考 データベース)
 (参考 データベースのカスタマイズ)
 インタフェース, 13-19, 14
 オブジェクトタイプ, 15
 オプション, 19
 テーブル, 15
 はじめに, 13
 フィールド, 41
 ライセンス, 66
 リンク, 41
 検索, 18
 言語, 14
 書類 - 自動読み込み, 14
 注, 13
 AssetCenter Server, 247-286
 Web - 制限, 285
 Windows統合セキュリティ (参考 Windows 統合セキュリティ)

アイコン, 256
 インストール, 260
 オプション, 257
 サービス - としての実行, 284
 サービスとして開始, 282
 データベース
 切断, 251, 250
 メイン画面, 256
 メッセージシステム, 255
 メニュー, 256
 モジュール - 設定, 259
 モニタ - タイムゾーン, 260
 概要, 247
 管理 - Web, 282
 AssetCenter Serverサービス, 284
 ドライブ, 284
 起動, 283, 283
 起動
 DOS, 250
 自動起動, 249
 手動起動, 249
 自動切断, 286
 実行
 Windows, 250
 推奨事項, 248
 終了 - 結果, 255
 制限, 250, 248
 即時トリガ, 281
 AssetCenter Serverで受領品をポートフォリオに自動的に作成 (オプション), 269, 181
 AssetCenter Server署名 (オプション), 184
 AssetCenterデータベース (ODBC接続), 121
 AutoCADコネクタ, 356
 AutoCAD - 統合, 355-359
 インストール, 356
 はじめに, 355
 使用, 357
 auto create statistics (Microsoft SQL Server), 52
 auto update statistics (Microsoft SQL Server), 53
 AvailableScript (オブジェクト), 105

B

BaseType (プロパティ), 110

Base (オブジェクト), 101, 99, 95, 93
 Basic - オプション, 182
 Basicスクリプト - インポート, 213
 Basicスクリプト (入力タイプ), 29, 29
 Basic関数 - オプション, 182
 Basic関数 (オプション), 182
 bin (ファイル), 143
 B (HTMLタグ), 46

C

cache (Sybase Adaptive Server), 55
 CalcFieldDesc (インスタンス), 106
 CalcField (オブジェクト), 110
 CalcMode (プロパティ), 101
 CaptionDetail (プロパティ), 108
 CaptionList (プロパティ), 108
 Cardinality (プロパティ), 98
 cfg (ファイル), 143
 Class (オブジェクト), 104
 CloseTable (DDE), 327
 CollectionCount (関数), 86
 CollectionCreate (関数), 84
 CollectionCurrentIndex (関数), 85
 CollectionIsFirst (関数), 85
 CollectionIsLast (関数), 85
 CollectionName (関数), 85
 CollectionNext (関数), 84
 Collection (インスタンス), 77
 CompleteDates (パラメータ), 314
 ComputeString (プロパティ), 91
 config (フォルダ), 17
 Connect (DDE), 322
 Connected (プロパティ), 90
 Connection (プロパティ), 90
 Connect-It - AutoCAD統合, 356
 Count (プロパティ), 92
 Create (プロパティ), 91
 CreationHistorized (プロパティ), 105
 Crystal Reports, 122 (参考 Crystal Reports)
 CurrentIndex (プロパティ), 92

D

DATA (Sybase Adaptive Server), 55

database.txt, 73, 26, 21
 構造, 23
 database device (Sybase Adaptive Server), 55
 Database Owner (Microsoft SQL Server), 53
 Database (インスタンス), 77
 Database (インスタンス), 90
 Data Only Locking (Sybase Adaptive Server), 55
 Date (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 db_block_buffers (パラメータ), 51
 db_block_size (パラメータ), 50
 DB2 UDB, 55, 55
 dbase.tpl, 23
 Dbb.Ext (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 Dbb.Fullname (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 Dbb.Name (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 Dbb.Path (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 Dbb.Shortname (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 DbbVersion (プロパティ), 90
 dbdict.tpl, 74, 22
 DBMS (参考 データベース)
 変更, 70, 59
 DBMSログイン, 59
 DDE, 319-345
 グローバルコマンド, 321
 Connect, 322
 Disconnect, 323
 ExecuteAction, 323
 ListAllTables, 324
 ListOpenTables, 325
 OpenView, 325
 コマンド, 320
 はじめに, 321
 コンテキスト, 321
 サービス, 320
 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, 333
 AddLink, 334

- GetValue, 334
- Hilight, 335
- RemoveLink, 336
- SetFocus, 336
- SetValue, 337
- SetValueWhere, 338
- Show, 339
- テーブルに関連するコマンド, 326
 - CloseTable, 327
 - GetRecordCount, 327
 - GetSelectionId, 333
 - ListAllFields, 330
 - ListAllLinks, 331
 - OpenTable, 326
 - SetFilter, 331
 - SetRecordMode, 329
 - SetSelection, 332
 - SetViewMode, 328
- テーマ, 320
- 呼び出し, 319
- 呼び出し - 例
 - Excel, 342
 - Visual Basic, 343
- はじめに, 339
- 内部呼び出し, 340
- 必要条件, 343
- DDE TestCenter.exe, 343
- DDEコマンド (参考 DDE)
- DDEコマンドサーバ, 319
- DDEサービス, 320
- DDEテーマ, 320
- default network packet size (Sybase Adaptive Server), 54
- DefaultScript (オブジェクト), 105, 95
- Delete (プロパティ), 91
- Desc (プロパティ), 109, 108, 107, 103, 101, 98, 94, 91
- Desktop Administrationサーバ署名 (オプション), 184
- Disconnect (DDE), 323
- dml_locks (パラメータ), 51
- dsd (ファイル), 143
- DstField (オブジェクト), 99

- DstTable (オブジェクト), 99
- dtLastModif (フィールド), 30
 - インポート, 195
- Duplicates (プロパティ), 100

E

- elseif (キーワード), 82
- else (キーワード), 82
- empty43.db, 58
- EmptyOnDup (プロパティ), 95
- EndAfter (パラメータ), 312
- EndBefore (パラメータ), 311
- EndDate (パラメータ), 310
- endfor (キーワード), 79
- endif (キーワード), 82
- endscript (関数), 86
- Enum (プロパティ), 103
- ExecuteAction (DDE), 323
- Exist (関数), 83
- extent i/o buffers (Sybase Adaptive Server), 54

F

- FeatClassDesc (インスタンス), 106
- FeatDesc (インスタンス), 103
- FeatParamDesc (インスタンス), 104
- FeatParam (オブジェクト), 110
- FeatureValueTable (オブジェクト), 92
- Feature (オブジェクト), 105
- FetchingArraySize (パラメータ), 350
- FieldCase (プロパティ), 95
- FieldCount (プロパティ), 91
- Field (インスタンス), 93, 77
- Field (オブジェクト), 110, 102
- filter (キーワード), 82
- FONT COLOR (HTMLタグ), 45
- FONT FACE (HTMLタグ), 45
- FONT SIZE=+n (HTMLタグ), 45
- FONT SIZE=-n (HTMLタグ), 45
- ForceDisplayScript (オブジェクト), 105
- ForeignKey (プロパティ), 94
- Formula (プロパティ), 107
- for (キーワード), 79
- FrameLabel (プロパティ), 109

FullName (フィールド)
 注, 31
 FullName (プロパティ), 106
 FuncDomain (プロパティ), 108
 function (関数), 86
 FVSourceTable (オブジェクト), 92

G

gbbase.xml, 61, 17
 gbbase*.*, 73, 22
 GenerateColors (パラメータ), 313
 GetRecordCount (DDE), 327
 GetSelectionId (DDE), 333
 GetValue (DDE), 334

H

HasClass (プロパティ), 103
 HasFeatureValueTable (プロパティ), 92
 HasMemoField (プロパティ), 92
 HasNotebook (プロパティ), 108
 HasParent (プロパティ), 106
 HasSystemPage (プロパティ), 108
 Hilight (DDE), 335
 Historized (プロパティ), 98, 94
 HistoryScript (オブジェクト), 105, 99, 95
 HR (HTMLタグ), 46
 HSplitControl (オブジェクト区切り), 137

I

iddac.scn (シナリオ), 272
 if (キーワード), 82
 IndexCount (プロパティ), 91
 Index (インスタンス), 77
 InternalName (プロパティ), 109, 107, 101, 98, 94, 91
 IsCalcField (プロパティ), 110
 IsConsolidated (プロパティ), 103
 IsFeatureValueTable (プロパティ), 92
 IsFeature (プロパティ), 110
 IsField (プロパティ), 110
 IsFirst (プロパティ), 91
 IsHorizontalSplit (プロパティ), 109
 IsInherited (プロパティ), 104

IsLast (プロパティ), 92
 IsLink (プロパティ), 110
 IsValidScript (オブジェクト), 92
 IsVerticalSplit (プロパティ), 109
 IsVisible (プロパティ), 107
 ItemColor (パラメータ), 313
 ItemLabel (パラメータ), 314
 I (HTMLタグ), 46

L

Label (プロパティ), 109, 108, 107, 103, 101, 98, 94, 91
 LDAP
 アクティブ化, 245
 パスワード, 244, 244
 リマインダ, 244
 実装, 244
 接続, 245
 LDAP - オプション, 183
 LDAPサーバポート (オプション), 183
 LDAPサーバ名 (オプション), 183
 LDAPディレクトリ (参考 LDAP)
 LDAPディレクトリのログインを格納する
 [amEmplDept] テーブルのフィールドのSQL
 名 (オプション), 184
 LDAP認証 (参考 LDAP)
 LineLabel (パラメータ), 315
 LinkCount (プロパティ), 91
 LinkFilter (プロパティ), 104
 Link (インスタンス), 97, 77
 Link (オブジェクト), 110
 ListAllFields (DDE), 330
 ListAllLinks (DDE), 331
 ListAllTables (DDE), 324
 ListOpenTables (DDE), 325
 LI (HTMLタグ), 46
 locking (Sybase Adaptive Server), 55
 log_buffer (パラメータ), 51
 LOG (Microsoft SQL Server), 53
 LOG (Sybase Adaptive Server), 55
 LogError (関数), 84
 LoginId (プロパティ), 90
 LoginName (プロパティ), 90

LongHelpCommentNoHTMLTag (プロパティ), 104, 98, 94
 LongHelpComment (プロパティ), 104, 98, 94
 LongHelpDescNoHTMLTag (プロパティ), 104, 99, 95
 LongHelpDesc (プロパティ), 104, 98, 94
 LongHelpSampleNoHTMLTag (プロパティ), 104, 98, 94
 LongHelpSample (プロパティ), 104, 98, 94
 LongHelpWarningNoHTMLTag (プロパティ), 104, 99, 94
 LongHelpWarning (プロパティ), 104, 98, 94

M

mail.ini, 296, 293, 291
 MainField (パラメータ), 309
 MainIndex (オブジェクト), 92
 MandatoryScript (オブジェクト), 105, 95
 MAPI (参考 メッセージシステム)
 maximum network packet size (Sybase Adaptive Server), 54
 max server memory (Microsoft SQL Server), 52
 MaxValue (プロパティ), 103
 memory (Sybase Adaptive Server), 54
 Microsoft SQL Server
 クライアント, 54
 サーバ, 52
 データベース - 作成, 52
 MinValue (プロパティ), 103

N

Name (プロパティ), 111, 106
 NTサービス, 282
 NullValues (プロパティ), 100
 null識別子レコードの検証 (モジュール), 280
 NumValue (プロパティ), 111

O

Object (インスタンス), 77
 ODBC, 119-122

Crystal Reports - 作成, 122
 DBMS, 120
 ドライバ
 インストール, 119
 有用性, 120
 概要, 119
 言語, 120
 使用する接続, 121
 制限, 119
 接続 - 作成, 122
 表示されたデータ, 121
 ODBCアクセス (参考 ODBC)
 ODBCドライバ (参考 ODBC)
 open_cursors (パラメータ), 51
 OpenTable (DDE), 326
 OpenView (DDE), 325
 optimizer_mode (パラメータ), 51
 Oracle, 50
 OrderBy (パラメータ), 316

P

PageItem (インスタンス), 109
 Page (インスタンス), 109
 ParentClass (オブジェクト), 106
 pdiac.scn (シナリオ), 274
 PlannerViewer (パラメータ), 309
 Positive (プロパティ), 95
 PrimaryKey (オブジェクト), 92
 PrimaryKey (プロパティ), 94
 procedure cache (Sybase Adaptive Server), 54
 processes (パラメータ), 51
 Property (インスタンス), 77

R

RAID, 49
 RawSource (プロパティ), 102
 ReadOnlyScript (オブジェクト), 95
 ReadOnly (プロパティ), 94
 Read (プロパティ), 105, 98, 93
 ReferencedStorageFieldCount (プロパティ), 102
 RelDstField (オブジェクト), 99
 RelevantScript (オブジェクト), 99, 96, 93

RelSrcField (オブジェクト), 99
 RelTable (オブジェクト), 99
 Remote Controlサーバ署名 (オプション), 185
 RemoveLink (DDE), 336
 ReverseLink (オブジェクト), 99
 rollback segments (Oracle), 52
 Row-Level Locking (Sybase Adaptive Server), 55

S

ScreenFilter (プロパティ), 108
 ScriptField (インスタンス), 111
 ScriptType (プロパティ), 102
 Script (AssetCenterデータベース記述インスタンスのオブジェクト), 107
 Script (インスタンス), 101
 Script (インスタンス), 77
 script (関数), 86
 SetFilter (DDE), 331
 SetFocus (DDE), 336
 SetOutput (関数), 84
 SetProperty (関数), 83
 SetRecordMode (DDE), 329
 SetSelection (DDE), 332
 SetValue (DDE), 337
 SetValueWhere (DDE), 338
 SetViewMode (DDE), 328
 shared_pool_size (パラメータ), 51
 Show (DDE), 339
 Size (プロパティ), 94
 SMTP (参考 メッセージシステム)
 sort_area_retained_size (パラメータ), 51
 sort_area_size (パラメータ), 51
 sort (キーワード), 80
 Source (プロパティ), 102
 SQLName (プロパティ), 109, 109, 108, 107, 103
 SqlName (プロパティ), 101, 98, 94, 91
 SQL名
 インデックス, 40
 テーブル, 24
 データベースオブジェクト - 競合, 140
 フィールド, 25

リンク, 32

用途, 24

SrcField (オブジェクト), 99
 SrcTable (オブジェクト), 99
 StartAfter (パラメータ), 312
 StartBefore (パラメータ), 311
 StartDate (パラメータ), 310
 StringValue (プロパティ), 111
 String (インスタンス), 111
 str (ファイル), 143
 stt (ファイル), 143
 Sybase Adaptive Server, 54, 54
 Sybase SQL Anywhere (runtime), 58
 SysEnumValue (インスタンス), 111
 SysPage (オブジェクト), 108
 system data source (UDB DB2), 57
 System (プロパティ), 109, 108, 103, 101, 98, 95, 92

T

TableCount (プロパティ), 90
 tables.txt, 73, 21
 構造, 23
 Table (AssetCenterデータベース記述インスタンスのオブジェクト), 107
 Table (インスタンス), 91, 77
 Table (オブジェクト), 108, 105, 102, 101, 95
 tempdb (Microsoft SQL Server), 53
 tempdb (Sybase Adaptive Server), 55, 54
 Template.Ext (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 Template.Fullname (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 Template.Name (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 Template.Path (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 Template.Shortname (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 testdde.xls, 342
 Time (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 tpl (ファイル), 78, 73, 22

Truncate Log on Checkpoint (Microsoft SQL Server), 53
 Truncate Log on Checkpoint (Sybase Adaptive Server), 55
 Typed (プロパティ), 98
 TypeField (オブジェクト), 99
 Type (インスタンス), 99
 Type (プロパティ), 107, 103, 98, 96, 93

U

UDB DB2
 アプリケーションカウンタサーバ, 57
 クライアント, 57
 Unit (プロパティ), 103
 Update (プロパティ), 105, 97, 93
 user data source (UDB DB2), 57
 Userlogin (AssetCenterデータベース記述インスタンスのグローバル変数), 112
 UserTypeFormat (プロパティ), 97, 93
 UserType (インスタンス), 100
 UserType (プロパティ), 107, 98, 96, 93
 UserType (プロパティ)
 インポート - 制限, 193
 usr (ファイル), 143

V

ValueField (オブジェクト), 105
 ValueOf (関数), 83
 Value (プロパティ), 111
 VbReturnType (プロパティ), 102
 View (インスタンス), 107
 VIM (参考 メッセージシステム)
 vim32.dll, 295
 VSplitControl (オブジェクト区切り), 137

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), 316
 win.ini, 296, 293, 291
 Windows統合セキュリティ
 AssetCenter Server, 251

サポートされる環境, 254
 概要, 251
 実装, 252
 従業員, 254
 wiz (ファイル), 143
 Write (プロパティ), 105, 97, 93

X

xml (ファイル), 143

