HP Asset Manager

ソフトウェアバージョン：5.20

高度な使い方

ドキュメントリリース日：01 October 2009
ソフトウェアリリース日：October 2009

法的制限事項

著作権

(c) Copyright 1994-2009 Hewlett-Packard Development Company, L.P.

限定保証条項

機密コンピュータソフトウェア

所有、使用、コピーには、HPによる有効なライセンスが必要です。

FAR 12.211および12.212準拠。商用コンピュータソフトウェア、コンピュータソフトウェアマニュアル、技術データは、ベンダの標準商用ライセンスに基づき、 米国政府にライセンス供与されています。

保証

HP製品およびサービスに対する保証は、当該製品またはサービスに付帯する明示的保証条項でのみ規定されます。

本規定のいかなる部分も、他の保証を構成すると解釈されるものではありません。

HPは本書の技術上または編集上の誤謬、欠落についての責任は負わないものとします。

本書に含まれる内容は、予告なく変更される場合があります。

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

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

商標

- Adobe®, Adobe logo®, Acrobat® and Acrobat Logo® are trademarks of Adobe Systems Incorporated.
- Corel® and Corel logo® are trademarks or registered trademarks of Corel Corporation or Corel Corporation Limited.
- Java™ is a US trademark of Sun Microsystems, Inc.
- Microsoft®, Windows®, Windows NT®, Windows® XP, Windows Mobile® and Windows Vista® are U.S. registered trademarks of Microsoft Corporation.
- Oracle® is a registered trademark of Oracle Corporation and/or its affiliates.
- UNIX® is a registered trademark of The Open Group.

目次

1. リストデータ 13

カスタムリストデータ 14
システムリストデータ 15

2. 履歴 17

レコードを作成する 18
テーブル内のフィールドまたは1リンク（単純リンク）を変更する（例：資産のユーザ） 18
別のテーブルへのnリンクを追加する（例：契約が伴う資産） 19
リンク（別のテーブルへのリンク）を削除する 19
リンク（別のテーブルへのnリンク）を変更する 20
レコードに関連付けられている任意管理項目の履歴を保持する 20

3. AQLクエリ 23

AQLツール 23
はじめに 23
クエリエディタ 26
AQLクエリの記述に関する推奨事項 33
ソートとインデックス 42
AQLシンタックス 44
AQL関数 56
クエリの例 62

4. Crystal Reports 69

制限 69
概要 69
Asset Manager WindowsクライアントがAsset Managerデータベース内に格納されたレポートにアクセスできるようにするための インストールと設定 71
Asset Manager WebクライアントがCrystal Reports Serverデータベースに格納されているレポートにアクセスできるようにするためのインストールと設定 72
レポートの表示 86
レポートを利用可能にする 90
Seagate Crystal Reportsのレポートを変更する 95
詳細レポートを作成する 95
レポートを印刷する 97
モジュールに関連するCrystalレポートの識別 99
画面上のボタン（Windowsクライアント）または画面のアクション（Webクライアント）へのレポートの関連付け 100

5. コントロールパネル 101

コントロールパネルの作成 101
コントロールパネルの例 103

6. 統計 105

統計の作成 105
クエリの代用としてのスクリプト 107
パフォーマンスの最適化 108
統計詳細の拡大 109
個別に表示するグループの最大数の定義 109
統計を見やすくする 110
統計の表示 111
統計の例 111
グラフ統計用のカスタムカラーの定義 122

7. アクション 127

開発のベストプラクティス 127
アクションの定義 128
アクションの作成 130
アクションの例 139
変数を使う 142
アクションをテストする 142
アクションを実行する 143

画面上のボタンへのアクションの関連付け 146

8. メッセージ 147

メッセージシステムの仕組み 147

9. ワークフロー 151

定義 152
概要 153
ワークフローの導入方法 155
ワークフローのグラフィカルエディタの使用方法 155
依頼の承認手続きで用いられるワークフローの例 158
ワークフローのコンテキスト 175
ワークフローの役割 177
ワークフローのアクティビティ 178
タスク 184
イベント 187
ワークフローの遷移 198
ワークフローのアラームと制限時間 198
ワークフローの実行グループ 200
ワークフローのトラッキング 201
終了済みワークフローインスタンスの削除 203
技術情報：データモデル 207

10. データの出力およびSQLビューの作成 209

定義 209
Asset Managerデータベースからデータを出力する 210
Asset ManagerデータベースでSQLビューを管理する 211
推奨事項 212
出力スクリプトを定義する 212
出力スクリプトを実行する 217

11. スクリプト 219

スクリプトの定義 219
スクリプトの適用 221
関数の概要 221
Basic関数を分類する 225
スクリプト作成の基本 225
スクリプトライブラリ 228
ヒントと注意事項 230
スクリプト例1 234
スクリプト例2 237

12. カレンダ 239

カレンダの概要 239
カレンダに影響される機能 240
カレンダの作成方法 240
カレンダの作成手順 240

13. タイムゾーン 247

タイムゾーンを管理する理由 247
タイムゾーン機能を実装する 248
タイムゾーンを作成する 248
タイムゾーンを管理する 249
Asset Manager Automated Process Managerでタイムゾーンを管理する 254
操作に与える影響 255

14. 特殊フィールド 259

特殊フィールドの定義 259
特殊フィールドの用途 260
特殊フィールドを作成する 260
特殊フィールドの使用方法 265

15. ウィザード 267

表記法 267
定義 268
構造モデル 270
ウィザードページのモデル 272
ウィザードに関する一般情報 272
一般的な構造とシンタックス 272
ノードのプロパティ 273
ウィザードの連鎖化 278
Basic関数 279
ROOTノードの定義 280
ROOTノードのシンタックス 280
ROOTノードのプロパティ 280
ROOTノードのサブノード 285
PAGEノードの定義 286
PAGEノードのシンタックス 286
PAGEノードのプロパティ 286
PAGEノードのサブノード 287
TRANSITIONノードの定義 288
TRANSITIONノードのシンタックス 288

TRANSITIONノードのプロパティ 288
TRANSITIONノードの特徴 289
FINISHノードの定義 290
STARTノードの定義 291
TIMERノードの定義 292
倍長整数および文字列ノードの定義 293
コントロールノードの定義 293
コントロールノードの一般的なシンタックス 293
コントロールのタイプと関連プロパティ 294
グラフィカルエディタの使用方法 317
ウィザードの作成例 320
ウィザードに関する一般的な質問 326

16. ニュース 331

ニュースの概要 331
ニュースの概要 332
ニュースの重要度 332
配信するメッセージ 332
ニュースの配信先リスト 332
ニュースを表示する 333

索引 335

図の一覧表

3.1. クエリエディタの活用法 26
7.1. ［実行可能ファイル］タイプのアクション 139
7.2. 参照するオブジェクトを指定するメッセージタイプのアクションの設定例 140
8.1. メッセージシステムの仕組み 148
9.1. ワークフロースキームの概略 152
9.2. Asset Managerのワークフロー - 概要 154
9.3. ワークフロー - 依頼の検証 158
9.4. ワークフロースキーム - 依頼の承認 172
9.5. ［データベース］タイプアクションの［パラメータ］タブページ . 189
9.6. 同期ワークフロースキームの例 196
9.7. 非同期ワークフロースキームの例 197
9.8. 終了イベント付きのワークフロースキーム 197
9.9. ワークフロースキームを定義できるメインテーブル 207
9.10. 実行中のワークフローに関連するメインテーブル 208
11.1. スクリプトビルダ 223
12.1. カレンダの詳細画面の［予定表］タブページ 241
12.2. カレンダの詳細画面の［プレビュー］タブページ 245
15.1. ウィザードの構造 271
15.2. 実行およびデバッグ ボタン 318
15.3. ウィザードの編成 322

表の一覧表

1.1. システムリストデータの値 15
3.1. 表記法 45
3.2. 論理演算子 49
3.3. 比較演算子 50
3.4. 集合型のAQL関数 57
3.5. 文字列型のAQL関数 57
3.6. 日付型のAQL関数 59
3.7. 日付型のAQL関数 60
3.8. 数値型のAQL関数 61
3.9. テスト型のAQL関数 62
9.1. ワークフローを制限する場合 192
9.2. イベントの様々な処理方法 193
9.3. イベントの様々な処理方法 195
11.1. 使用可能なデータ型 224
14.1. 特殊フィールドのタイプ 261
15.1. 表記法 268
15.2. ROOTノードの論理プロパティ 281
15.3. ROOTノードの物理プロパティ 283
15.4. ROOTノードのサブノード 285
15.5. PAGEノードの論理プロパティ 287
15.6. PAGEノードの物理プロパティ 287
15.7. PAGEノードのサブノード 288
15.8. TRANSITIONノードの論理プロパティ 289
15.9. FINISHノードの論理プロパティ 290
15.10. FINISHノードの物理プロパティ 290

15.11. STARTノードの論理プロパティ 291
15.12. TIMERノードの論理プロパティ 292
15.13. 倍長整数および文字列ノードの論理プロパティ 293
15.14. すべてのコントロールに共通の論理プロパティ 294
15.15. すべてのコントロールに共通の物理プロパティ 296
15.16. CHECKBOXコントロールのプロパティ 297
15.17. COMBOBOXコントロールの物理プロパティ 298
15.18. OPTIONBUTTONSコントロールの物理プロパティ 299
15.19. LISTBOXコントロールの物理プロパティ 299
15.20. LISTBOXコントロールのメソッド 301
15.21. LISTBOXコントロールの必須論理プロパティ 302
15.22. LABELコントロールの物理プロパティ 303
15.23. PROGRESSBARコントロールの物理プロパティ 303
15.24. COMMANDBUTTONコントロールの物理プロパティ 303
15.25. DBLISTBOXコントロールの物理プロパティ 304
15.26. DBQUERYBOXコントロールの物理プロパティ 307
15.27. 標準モードのDBEDITコントロールの物理プロパティ 309
15.28. DBPATHコントロールの必須論理プロパティ 311
15.29. LINKEDITコントロールの論理プロパティ 311
15.30. TEXTBOXコントロールの物理プロパティ 313
15.31. CHARTコントロールの論理プロパティ 313
15.32. CHARTコントロールの物理プロパティ 314
15.33. FILEEDITLコントロールのプロパティ 315
15.34. TICKEDITコントロールのプロパティ 315
15.35. NUMBOXコントロールのプロパティ 316
15.36. COMBOEDITコントロールのプロパティ 317
15.37. コントロールのプロパティ 317
16.1. マーキーツールバーのボタン 333

1 リストデータ

リストデータは、役職、位置、国、ブランドなど、Asset Managerのいくつかのフィールド（詳細画面の標準フィールドまたは任意管理項目の値）に入る値のリストです。これらのリストは、データベースの［**リストデータ**］（seItemizedList）テーブルに格納されています。

リストデータによって、特定のフィールドの値が標準化されるため、データ入力が容易になります。

画面にリストデータフィールドが含まれている場合、リストデータの値はドロップダウンリストに表示されます。値をフィールドに割り当てるには、リストからエントリを選択するだけです。

リストデータ値は、データベースの［**リストデータの値**］（amItemVal）テーブルに格納され、［**リストデータ**］（seItemizedList）テーブル内の対応するリストにリンクされます。

Asset Managerに付属のデータベースには、次の2種類のリストデータにリンクされるフィールドがあります。

- カスタムリストデータ
- システムリストデータ

カスタムリストデータは、**クローズ済**（ユーザが変更不可）にしたり、**オープン**（ユーザが新規リストエントリを追加可能）にしたりできます。リストデータのオープン/クローズ済は、［**リストデータ**］（seItemizedList）テーブルに対する読み書きアクセス権を所有しているユーザが変更できます。

注意:

管理者は、Asset Manager Application Designerを使用して、システムまたはカスタムのリストデータに関連付けられる新しいフィールドを作成できます。

カスタムリストデータ

カスタムリストデータには、ナビゲーションバーの［**管理／システム／リストデータ**］リンクを使用してアクセスできます。

カスタムリストデータの値

リストデータの詳細画面に表示される値のリストは、フィールドのドロップダウンリストに表示される値です。

リストデータの詳細画面で、値を削除、変更、または追加できます。

オープンカスタムリストデータ

オープンリストデータは、リストデータの詳細画面の［**タイプ**］フィールド（SQL名：seType）が［**オープン**］に設定されているリストデータです。

リストデータがオープンになっている場合、Asset Managerのユーザは、ドロップダウンリストに表示される以外の値をフィールドに入力できます。

ユーザが新しい値を入力すると、この値がリストデータの値の一覧に追加されます（値の一覧はすべてのユーザで共有されます）。追加する際、作成を確認するメッセージが表示されます。

クローズ済カスタムリストデータ

クローズ済リストデータは、リストデータの詳細画面の［**タイプ**］フィールド（SQL名：seType）が［**クローズ済**］に設定されているリストデータです。

クローズ済リストデータの場合、Asset Managerのユーザがドロップダウンリストに表示される値以外の値を入力することはできません。

システムリストデータ

システムリストデータの値のリストは、Asset Manager Application Designerで定義します。このリストは、管理者がカスタマイズできます。

これらのリストデータは、ナビゲーションバーの［**管理／システム／リストデータ**］リンクを使用して編集することはできません。

システムリストデータの値

表示される値は、データベースに保存されている値とは異なります。

データベース内では、これらの値は数値として保存されます。

以下の表では、ポートフォリオ品目の詳細画面の［**割り当て**］フィールド（seAssignment）を例に取り上げています。

表 1.1. システムリストデータの値

データベースに保存される値	表示される値
0	使用中
1	在庫中
2	除却済（または消費済）
3	納品待ち
4	メンテナンスのため返却
5	サプライヤに返却
6	紛失

システムリストデータを参照するには、以下のようにいくつかの方法があります。

- システムリストデータで入力したフィールドのヘルプ（Windowsクライアントのみ）を表示する。
- Asset Manager Application Designerを使う。
- データベース構造を記述している「database.txt」ファイルを使用する。このファイルは、「<Asset Managerインストールフォルダ>\ doc\ infos」ディレクトリにあります。

2 履歴

データベース内のフィールドまたはリンクに加えられた変更を記録できます。履歴の保持が指定されたフィールドの値を作成、変更、または削除するたびに、Asset Managerは問題になっている画面の［**履歴**］タブ内に、履歴項目を加えます。

変更を記録するには、フィールドまたはリンクの履歴の保持を指定する必要があります。手順は次のとおりです。

1 ショートカットメニューから［**オブジェクトの設定**］を選択します。
2 設定画面の［**全般**］タブを選択します。
3 ［**履歴保持**］フィールドを［**はい**］に設定します。
4 ［**OK**］をクリックして作成を確定します。

注意:

データベースのカスタマイズウィンドウで［OK］をクリックすると、履歴オプションに加えた変更が直ちにデータベースに保存されます。フィールドまたはリンクの履歴を保持するかどうかはAsset Manager Application Designerでも定義できます。

フィールドまたはリンクの履歴を保持すると、Asset Managerのすべてのユーザが履歴を利用できます。

いずれかのフィールドまたはリンクで履歴を保持することを指定すると、そのテーブルのレコードの詳細画面に［**履歴**］タブページが表示されます。フィールドまたはリンクに変更を加えると、［履歴］タブページに変更内容の詳細を示す［履歴項目］が加えられます。

履歴項目には次の情報が含まれます。

- ［**更新日**］（SQL名：dtLastModif）：変更を行った日付
- ［**作成者**］（SQL名：Author）：変更を行ったユーザ（ログイン名と氏名）
- ［**フィールドまたはリンク**］（SQL名：Field）：変更したフィールドまたはリンクのSQL名
- ［**元の値**］（SQL名：PreviousVal）：変更されたフィールドの元の値（「コメント」タイプのフィールドを除く）
- ［**新しい値**］（SQL名：NewVal）：変更したフィールドの新しい値（「コメント」タイプのフィールドを除く）。デフォルトでは、このフィールドはリストに表示されません。表示するには、リスト上でマウスの右ボタンをクリックして［**ユーティリティ／リストの設定**］メニューを選択します。

  注意:

  旧バージョンのAsset Managerからデータベースをインポートする場合、履歴項目の［新しい値］フィールドは空になります。

- ［**元のコメント**］（SQL名：memPreviousCmt）：「コメント」タイプのフィールドの修正前の値。「コメント」タイプのフィールド（最大サイズ：半角の場合で32767文字）は、他のフィールドとデータベースに保存する方法が異なるので、処理方法も異なります。

Asset Managerの履歴機能は、履歴を取得するオブジェクトの種類によって以下のように異なります。

レコードを作成する

テーブルの主キーに対応するIDフィールドの変更履歴を保持することを指定した場合は、レコードの作成が記録されます。

記録される内容

- ［**更新日**］：レコードの作成日
- ［**作成者**］：レコードの作成者
- ［**フィールドまたはリンク**］：「作成」
- ［**元の値**］：「作成」

テーブル内のフィールドまたは1リンク（単純リンク）を変更する（例：資産のユーザ）

記録される内容

- ［更新日］：変更した日付
- ［作成者］：変更者
- ［フィールドまたはリンク］：変更したフィールドの名前
- ［元の値］：変更したフィールドの元の値
- ［新しい値］：変更したフィールドの新しい値

注意:

変更したフィールドがシステムのリストデータを使って入力される場合、［前の値］および［新しい値］フィールドには、データベース内の格納されている値ではなく、システムのリストデータの入力フィールドに表示されている値が格納されます。

例：［作業指示］（amWorkOrder）テーブルで、［ステータス］（seStatus）はシステムのリストデータから入力されます。このリストデータのエントリの1つは、［通知済］と表示され、0として格納されます。

［前の値］および［新しい値］フィールドは、0ではなく、［通知済］を格納します。

この動作は、Asset Managerのバージョン5.00以降に有効になりました。

以前のバージョンでは、データベースに格納されている値が使用されていました。

別のテーブルへのnリンクを追加する（例：契約が伴う資産）

記録される内容

- ［更新日］：追加した日付
- ［作成者］：追加者
- ［フィールドまたはリンク］：リンク名
- ［元の値］：追加されたリンク先レコードのリファレンス
- ［新しい値］：変更したリンクの新しい値

リンク（別のテーブルへのリンク）を削除する

記録される内容

- ［更新日］：削除した日付
- ［作成者］：削除者
- ［フィールドまたはリンク］：削除されたリンク先レコードのリファレンス
- ［元の値］：削除されたリンク先レコードのリファレンス

- ［**新しい値**］：変更したリンク（空）の新しい値

リンク（別のテーブルへのnリンク）を変更する

Asset Managerでは、リンクへの変更を記録しません。変更の履歴を記録するには、古いリンクを削除してから新しいリンクを追加する必要があります。

レコードに関連付けられている任意管理項目の履歴を保持する

Asset Managerでは、任意管理項目の履歴をデータベース内の他のフィールドと同様に保持できます。任意管理項目の場合は、次の操作の履歴が保持されます。

- 任意管理項目の追加
- 任意管理項目の削除
- 任意管理項目の値の変更

AssetCenterの履歴機能は、実行する操作によって以下のように異なります。

任意管理項目の追加

任意管理項目のパラメータの詳細画面の［**履歴の保持**］（SQL名：seKeepHistory）フィールドが［**はい**］に設定されており、［**メインレコードの作成中でも履歴を保持**］（SQL名：bCreationHistory）チェックボックスがオンになっている場合は、新しい任意管理項目の追加が記録されます。

記録される内容

- ［**更新日**］：任意管理項目を追加した日付
- ［**作成者**］：任意管理項目を追加した担当者
- ［**元の値**］：「作成」
- ［**フィールドまたはリンク**］：任意管理項目のSQL名

任意管理項目の削除

［**履歴の保持**］（SQL名：seKeepHistory）フィールドが［**はい**］に設定されている場合は、任意管理項目の削除が記録されます。

記録される内容

- ［**更新日**］：任意管理項目を削除した日付
- ［**作成者**］：任意管理項目の削除者
- ［**フィールドまたはリンク**］：任意管理項目のSQL名
- ［**元の値**］：削除した任意管理項目（任意管理項目の値）

- ［新しい値］：任意管理項目（空）の新しい値

任意管理項目の値の変更

［履歴の保持］（SQL名：seKeepHistory）フィールドが［はい］に設定されている場合は、任意管理項目の変更が記録されます。

記録される内容

- ［更新日］：任意管理項目を変更した日付
- ［作成者］：任意管理項目の変更者
- ［フィールドまたはリンク］：任意管理項目のSQL名
- ［元の値］：任意管理項目の変更前の値
- ［新しい値］：変更した任意管理項目の新しい値

警告:

レコードを削除すると、すべての履歴項目の削除は、同時またはAsset Manager Serverの更新時に実行されます。

履歴項目の作成、削除、変更

履歴項目の作成の履歴を保持することはできません。

履歴項目を作成する

任意管理項目の履歴項目の作成をトリガするには、［履歴の保持］フィールドを［はい］に設定する必要があります。この設定を行うには、任意管理項目の詳細画面で［パラメータ］タブを選択し、リストの右側の ボタンをクリックします。

任意管理項目のパラメータの詳細画面が表示されます。［履歴の保持］フィールドは、この画面の［制限］タブページにあります。

このフィールドを［はい］に設定すると、Asset Managerはその任意管理項目の履歴項目を自動的に作成します。履歴項目は、その任意管理項目に関連付けられているテーブルの［履歴］タブページに表示されます。

警告:

レコードを削除すると、すべての履歴項目も同時またはAsset Manager Serverの更新時に削除されます。履歴作成の履歴は保持できません。

3 AQLクエリ

AQLツール

AQLクエリは、状況依存型の場合、Asset Managerグラフィカルユーザインタフェースで実行できます。状況依存型でない場合、Asset Manager Exportで実行できます。

はじめに

このセクションでは、AQL言語と、クエリが必要になる状況について説明します。

AQL

AQL（Advanced Query Language）は、Asset ManagerがAsset Managerデータベースにアクセスするのに使うクエリ言語であり、SQLと同じように機能します。AQLで記述したクエリは、使用中のデータベースエンジンで使う同等のSQL言語に自動的に変換されます。

注意:

AQLを直接使うには、SQLとデータベースに関する十分な知識が必要です。

AQL言語の利点

Asset Managerデータベースのクエリには、次の理由からSQLよりもAQLの方が適しています。

データベースエンジンによらない

Asset Managerがサポートする様々なデータベースエンジンは、すべて異なるバージョンのSQLを採用しており、互換性がありません。AQLは、これらのデータベースエンジンに関係なく使うことができます。

別のデータベースエンジンに移行した場合でも、AQLで記述したクエリは同様に機能します。

例えば、AQLの**Substring**関数はSQL Oracle for WorkGroupsの**Substr**およびMicrosoft SQL Server SQLの**Substring**に相当します。

最適なSQLコードの生成

AQLは、使用するデータベースエンジンに応じて最適なSQLコードを生成します。

特にインデックス機能を使用する場合は、生成されるSQLコードが大きく異なります。例えば、［モデル**ID**］（SQL名：**Model_lModelId**）と［完全名］（SQL名：**FullName**）にインデックスを強制して、モデルの完全名を検索する場合は、次のAQLを記述します。

```
SELECT FIRST_ROWS lModelId, FullName FROM amModel
```

生成されるSQLコードは、使用するDBMSによって異なり、DBMSごとに最適化されます。例えば、Oracleでは次のSQLコードが生成されます。

```
SELECT /*+ FIRST_ROWS INDEX_ASC(M1 Model_lModelId) */ M1.lModelId, M1.FullName FROM amModel M1
```

Microsoft SQL Serverでは、次のSQLコードが生成されます。

```
SELECT M1.lModelId, M1.FullName FROM amModel M1 ORDER BY M1.lModelId
```

IBM DB2では、次のコードが生成されます。

```
SELECT lModelId, FullName FROM amModel OPTIMIZE FOR 100 ROWS
```

Asset Managerデータベースへのアクセスを簡略化

AQLは、リンクと結合の扱いを簡略化します。そのため、AQLでクエリを記述すると、直接SQLを使う場合に比べ、速くデータベースにアクセスできます。

さらに、AQLを使うと任意管理項目へのアクセスが簡単になり、関連付けられているテーブルのフィールドとして直接任意管理項目を使うことができます。

AQLによって、特殊フィールドも容易に活用できます。

AQLの特徴（SQLと比較して）

AQLはDDL（Data Definition Langage）ステートメントをサポートしません。

AQLには、結合、任意管理項目、特殊フィールドの処理を簡単にする拡張機能があります。

警告:

SQLステートメントを使って直接Asset Managerデータベースに書き込まないでください。

Asset Managerのクエリ

クエリを使うと、テーブル内の情報に関する判定基準とそれにリンクしているテーブル内の情報の判定基準を組み合わせることができます。

次の場合にクエリを使うことができます。

- レコードリストで使うフィルタを作成する。通常は、WHERE句を使った簡単なクエリを使います。
- ビューを定義する。
- エクスポートモジュールの出力条件を定義する。
- Crystal Reportsを使ってレポートを作成する。
- ウィザードを作成する。
- Asset Manager APIを使う場合。
- Asset ManagerをDDEサーバとして使う場合。

AQL（Advanced Query Language）は、Asset Manager内部に組み込まれたクエリ言語であり、Asset Managerデータベースへのアクセスのみに使うことを想定しています。

Asset Managerに組み込まれているエディタを使って、次の方法でAQLクエリを作成できます。

- グラフィカルユーザインタフェースを使って作成
- AQLで直接クエリを記述

警告:

AQLの使い方を分かりやすく説明するために、後述の例ではすべてのAQLシンタックスを使用しています。特にSELECT、WHERE、およびFROM句が明示的に示されています。AQLクエリのWHERE句だけを使ったクエリフィルタや、式ビルダなどの特定の機能を使うと、より簡単にクエリを作成できます（これは一部の句のみが表示されるためです）。ただし、後述の例をこれらの機能用に使うことはできません。

クエリエディタ

Asset Managerはクエリエディタを内蔵しています。このツールを使うと、クエリを設計し、結果をプレビューすることができます。クエリエディタは、特にデータベース管理者やクエリに精通したパワーユーザを対象としています。

注意:

プレビュー機能は、Asset Manager Webクライアント、およびAsset Manager Windowsクライアント内の一部の簡易版のクエリエディタには適用できません。

概要

クエリエディタでは、次のいずれかの方法でクエリを設計できます。

- グラフィカルユーザインタフェースを使って作成
- AQLで直接クエリを記述

Asset Manager Windowsクライアントを使用している場合、グラフィカルユーザインタフェースを使う場合も直接AQLで記述する場合も（多くの場合、2つの方法を組み合わせて使います）、作成したクエリをSQL言語でリアルタイムで表示することができます。ただし、クエリを直接SQLで記述することはできません。

図 3.1. クエリエディタの活用法

パワーユーザや管理者は、クエリエディタを使ってAQLクエリを作成、変更、削除できます。作成されたクエリは作成者や他のユーザが適切な状況で使うことが出来ます。

クエリエディタにアクセスする

Windowsクライアント

Asset Manager Windowsクライアントのクエリエディタには、次の方法でアクセスできます。

- ［**ツール／クエリ**］メニューから。このメニューを使ってクエリを作成すると、他のユーザもそのクエリを自由に使用できるようになります。クエリは、次の方法で実行できます。

  注意:

  ［クエリ］タブには、、次の［フィルタ（WHERE句）］と［プレビュー］タブページがあります。

  - **［フィルタ（WHERE句）］タブページは、クエリの条件を指定するグラフィカルユーザインタフェースです。このタブページでSQL WHERE句の要素を定義します。**
  - **［プレビュー］タブページには、クエリをSQLコードに変換したものが表示されます。このタブページでクエリをテストできます。**

- クエリのメインテーブルが表示されている時に、クエリフィルタを使用する。
- アクセス制限、クエリフィルタ、リストの設定など、Asset Managerのクエリを呼び出す機能を使う。
- Asset Manager Export Toolなどの外部プログラムを使う。

以下で示しているバージョンのクエリエディタは、状況に応じてある程度簡略化されています。

例：次のようなクエリがあるとします。

```
SELECT [FIRST_ROWS] <フィールド>[, <フィールド>...] FROM <テーブル>
[WHERE <句>] [ORDER BY <句>]
```

簡易版のクエリエディタ（シンプルフィルタ、クエリフィルタなど）では、クエリのWHERE句だけを定義するだけで済みます。クエリのその他のコンポーネント（クエリを実行するテーブル、フィールドなど）は明示的に指定する必要がありません。例えば、クエリフィルタの場合、テーブルはフィルタを適用するテーブルであり、フィールドと並べ替え条件は［**ユーティリティ／リストの設定**］ショートカットメニューメニューで定義した列／並べ替えの条件です。［**ツール／クエリ**］メニューからクエリエディタにアクセスした場合も同様です。

例えば、次のような明示的に記述したクエリがあるとします。

```
SELECT self FROM amModel WHERE Brand.Name='Compaq'
```

これと同じ内容のクエリを、モデルのテーブルでクエリフィルタを使って行う場合は、次のように記述されます。

```
Brand.Name='Compaq'
```

一方、［**リストの設定**］コマンドを使うと、次のように複雑なクエリエディタにアクセスできます。

- ［**列／並べ替え**］タブページでは、リストの列に表示するフィールドと並べ替え条件を定義します（これらの並べ替え条件はORDER BY句に相当します）。
- ［**インデックスの強制使用**］チェックボックスは、SQLコードのFIRST_ROWS句に置き換わります。
- ［**フィルタ（WHERE句）**］タブページでは、WHERE句を定義します。
- テーブルは明示的に指定しません。

Webクライアント

クエリテーブルからのみ、クエリエディタにアクセスできます。

- ［**管理／クエリ**］メニューを選択します。
- ［**新規作成**］ボタンをクリックします。
- ［**クエリ**］タブを選択します。
- テキストボックスの上にあるアイコンをクリックすると、ポップアップウィンドウにクエリエディタが表示されます。

クエリエディタを使ってクエリを作成する

手順1：クエリの詳細画面上部のフィールドに入力する

クエリの開始テーブルを指定する必要があります。

作成するクエリに他のユーザがアクセスできるようにする場合は、［**全般**］タブの［**共有しない**］（SQL名：bPrivate）チェックボックスをオフにします。

注意:

管理者は、［共有しない］オプションが選択されているクエリも含め、データベース内のすべてのクエリにアクセスできます。

クエリの基本的な情報を入力してから［**クエリ**］タブをクリックすると、クエリを使うことができます。

手順2：フィルタ条件を定義する

Asset Managerのクエリエディタを使うと、フィールド、計算式、定数、演算子を組み合わせる条件を定義できます。

複数のフィルタ条件を定義できます。

Windowsクライアント

Windowsクライアントでフィルタ条件を定義するには、次の手順に従います。

1 比較したいフィールド、定数、式を［**フィールド1**］、［**フィールド2**］に指定し、演算子を選択します。

2 ボタンを使ってこのクエリを画面下部のウィンドウに転送し、フィルタ条件を確認します。

3 **AND**や**OR**の論理演算子で連結した複数のフィルタ条件を定義するには、その他のフィルタ条件を作成し、［**AND**］ボタンまたは［**OR**］ボタンで条件を連結します。

4 ［**変更**］をクリックし、クエリの作成を確定します。

注意:

フィルタ条件を変更するには、ボタンをクリックしてウィンドウの内容を消去するか、またはAQLコードを直接変更します。

注意:

グラフィックツールを使わずに、［フィルタ（**WHERE**句）］タブページの下部のテキストボックスに直接AQLでクエリを入力することもできます。

Webクライアント

Webクライアントでフィルタ条件を定義するには、フィールド、関数、および演算子をそれぞれのタブから選択します。

Webクライアントには、次の種類の演算子があります。これらの演算子では、同じインタフェースを使用して、式をビルドし、フィルタ条件を定義できます。

- 算術
- 関係
- 論理

注意:

Asset Manager Webクライアントでは、クエリのさまざまな要素（フィールドや演算子など）が別個のフィールドでなく1つのテキストフィールドに定義されています。クエリに含まれる要素を正しいシーケンスで選択および並べてください。

手順3：クエリの実行をプレビューする（Windowsクライアントのみ）

クエリをテストし、SQL言語に変換したクエリを表示するには：

1 クエリの詳細画面の［**プレビュー**］タブページに移動します。

2 アイコンをクリックします。クエリによって検索されたレコードが下のウィンドウに一覧表示されます。クエリ条件に一致するレコード数は、ウィンドウの右下に表示されます。

注意:

［プレビュー］タブページに表示されるSQLコードを直接変更することはできません。

クエリで使うフィールド

クエリのフィルタ条件を定義する場合、次のフィールドを指定できます。

- クエリの対象となるテーブル内のフィールド
- リンクしているフィールド
- テーブルに関連付けられている任意管理項目

式を記述する

式を使うとクエリで計算を実行できます。例えば**Count**関数を使うと、クエリによって検索されたレコード数をカウントできます。

式を記述するには、次の方法があります。

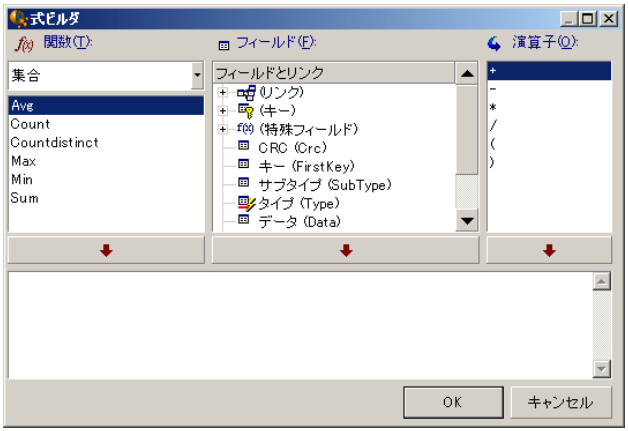
- クエリを記述するフィールドに直接入力する。
- Asset Managerの式ビルダを使う。

Windowsクライアント

Windowsクライアントでは、式ビルダはクエリエディタに統合され、クエリエディタ内からのみアクセスできます。

式ビルダにアクセスするには、クエリエディタの［**フィルタ（WHERE句）**］タブから、フィールドの値を表すボックスの隣にあるボタンをクリックします。

このボタンは、フィールドの型（［フィールド1］または［フィールド2］）に［式］が入力された場合のみ、使用できます。



式ビルダは、次の3つの列で構成されます。

- ［関数］列には、既存のAQL関数のリストが表示されます。をクリックすると、AQL関数のタイプ（集合、文字列、日付、数値、テスト）を選択できます。選択したタイプがフィルタとなり、そのタイプの関数だけを表示できます。
- ［フィールド］列には、クエリに使用できるフィールドのリストが表示されます。
- ［演算子］列には、式で使用できる演算子のリストが表示されます。

［関数］、［フィールド］、［演算子］を式に挿入するには、次の手順に従います。

1 関数、フィールド、演算子のいずれかを選択します。

2 をクリックします。

式の定義が終了し、［**OK**］をクリックすると、クエリの詳細画面の［**フィルタ（WHERE句）**］タブページに画面が切り替り、定義した式が表示されます。

Webクライアント

Webクライアントでは、式ビルダはスタンドアロンツールで、クエリエディタとは独立しています。ただし、両方のツールには、非常によく似たインタフェース、機能、用途があります。唯一の違いは、式ビルダの演算子がクエリビルダより少ないことです。

式ビルダにアクセスするには、式編集用のテキストボックスの上にあるアイコンをクリックします。ポップアップウィンドウに式ビルダが表示されます。

式ビルダは、次の3つのタブで構成されます。

- ［**Function**］タブには、カテゴリでグループ化されたさまざまな関数が含まれています。ドロップダウンリストから異なるカテゴリを選択すると、使用

可能な関数のリストが変わります。関数をダブルクリックすると、式ビルダの上部にある編集ボックスに関数が挿入されます。

- ［**Field**］タブはコンテキストに依存し、現在のAsset Managerデータベーステーブルで使用可能なフィールドとリンクが含まれています。フィールドまたはリンクをクリックすると、編集ボックスに挿入されます。
- ［**Operators**］タブには、式に適用可能な算術演算子が含まれています。演算子をクリックすると、編集ボックスに挿入されます。

式ビルダの下部にある次のボタンを使用すると、式の構築が容易になります。

- ［**クリア**］ボタンは、警告なしで編集ボックスの内容を消去します。
- ［**チェック**］ボタンは、式のシンタックスをチェックします。式が有効かどうかを示すメッセージが、編集ボックスの上部に表示されます。
- ［**保存**］ボタンは、最初に式のシンタックスを検証します。シンタックスが有効な場合、ウィンドウが閉じられ、式がAsset Managerデータベースに保存されます。それ以外の場合、ウィンドウは開いたままで、そのウィンドウにエラーメッセージが表示されます。
- ［**キャンセル**］ボタンは、変更なしでウィンドウを閉じます。

定数：Windowsクライアントのみ

定数 x は、選択基準に割り当てる固定の値です。例えば、ブランドが「**3Com**」であるすべてのモデルを検索する場合は、モデルのテーブルの［**Brand.Name**］リンクフィールドに定数値「**3Com**」を割り当てます。

定数を選択するには：

1 アイコンをクリックします。
2 検索条件として指定されたフィールドのデータベース内の既存の値を選択するウィンドウが表示されます。

注意:

リストデータ型のフィールドの場合でも、アイコンをクリックすると上記の選択ウィンドウが表示され、データベースで使っているリストデータの値のみが表示されます。

AQLクエリの記述に関する推奨事項

AQLによるクエリの記述を始める前に、このセクションをお読みください。

このセクションでは、次の内容について説明します。

- AQLに固有の表記
- クエリの最適な設計に有効なAQLとAsset Managerデータベースの特殊性

「AQLシンタックス [献 44]」と「AQL関数 [献 56]」のセクションに、この節の補足説明があります。

警告:

AQLで記述したクエリには、データベース内のフィールド、リンク、およびテーブルのSQL名（SQLName）を使います。SQL名については、「Database.txt」ファイルと「Database.xml」ファイルを参照してください。このファイルには、データベースの構造が記述されており、すべてのSQL名が記載されています。

このファイルは、「［Asset Managerのインストール先フォルダ］/doc/infos」にあります。このファイルにアクセスするためには、ドキュメントをインストールする必要があります。

AQL結合について

定義

結合とは、複数のデータテーブルを1つのクエリに統合することです。

AQL結合

Asset Managerのデータベースの記述では、テーブルとフィールドだけでなく、テーブル間のリンクも定義します。これによって、AQLレベルでの結合を自動化できます。

AQLリンクは次のように表現されます。

```
Link[.Link[.Field]]
```

AQLでは、上記のように結合を簡単に処理できるので、Asset Managerデータベースで使用する大部分のクエリを簡単に作成できます。

例

次のAQLで記述したクエリは、モデルごとに以下のデータを返します。

- モデルの**ID**（**lModelId**）
- モデルの**完全名**（**FullName**）
- ［ブランド］（**amBrand**）にリンクするテーブルの［**名前**］

```
SELECT lModelId, FullName, Brand.Name FROM amModel
```

同じクエリをOracleまたはMicrosoft SQL ServerのSQLで記述すると以下のようになります。

```
SELECT M1.lModelId, M1.FullName, B2.Name FROM amModel M1, amBrand B2 WHERE M1.lBrandId=B2.lBrandId
```

[モデル]（**amModel**）テーブルと**[ブランド]**（**amBrand**）テーブルの間の2つの結合は、AQLでは自動的に処理されます。Asset Managerのグラフィッククエリエディタを使えば、ツリー構造のリストで選択したテーブルまたはリンクしているテーブルのフィールドをクリックするだけで、対応するAQLコードを作成できます。

OracleとDB2以外のシステムでは、外部結合の数は1つに限られています。

Microsoft SQL Server 7とMSSQL 2000では、「amdb.ini」ファイルを変更して、クエリ実行に関連する問題を防ぎます。このファイルの中の該当する接続のセクションに以下の命令を追加してください。

```
UseSql92Join=1
```

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

主キーが0のレコードの理由と有用性

主キーが「0」（ゼロ）のレコード

Asset Managerデータモジュールには、次の特殊性があります。

- 各テーブルの主キーと外部キーは「32ビット整数」型のフィールドです。
- レコードにリンクしていない外部キーは、NULLではなく「0」に設定されます。
- 各テーブルには、主キーが「0」に設定された空のレコードがあります。

有用性

主キーが「0」のレコードを使うと、AとBの2つのテーブル間での外部結合を使わないクエリの結果に、テーブルB内の実際のレコードにリンクしない（リンクが存在しない）テーブルAのレコードを含めることができます。このレコードは、テーブルBの主キーが「0」のレコードにリンクしているテーブルAのレコードです。

例：

次のAQLで記述したクエリは、ポートフォリオ品目の資産タグごとに、そのユーザおよび責任者の名前を返します。

```
SELECT AssetTag, User.Name, Supervisor.Name FROM amPortfolio
```

このクエリの結果には、ユーザや責任者に割り当てられていないポートフォリオ品目が含まれます。データベースでは、このようなポートフォリオ品目は、[部署と従業員]テーブル内で主キーが「0」のレコードにリンクされています。

主キーが「0」のレコードを使う理由

このセクションでは、外部のSQL結合を使ったクエリではテーブルBのレコードにリンクしていないテーブルAのレコードを選択できるのに、AQLではなぜ主キーが「0」のレコードを使うのかを説明します。

一部のRDBMでは複数の外部結合を処理できないという欠点がありますが、主キーが「0」のレコードを使えば、AQLクエリから生成されたSQLコードで外部結合を使う必要がないため、この欠点を補うことができます。

例：

次のAQLクエリは、ポートフォリオ品目ごとにその資産タグとユーザの場所の名前を検索します。結果には、ユーザのないポートフォリオ品目と、場所のないユーザのポートフォリオ品目が含まれます。

```
SELECT AssetTag, user.location.name FROM amPortfolio
```

生成されたSQLコードでDBMSの外部結合を使った場合は、特定のデータベースエンジン用に生成されたSQLコードは次のようになります。

```
SELECT a.AssetTag, l.name FROM amPortfolio a, amEmplDept e, amLocati
on l WHERE a.lUserId *= e.lEmplDeptId AND e.lLocaId *= l.lLocaId
```

このコードは複数の外部結合を次々に使うので、特定のデータベースエンジンではサポートされません。

しかし、部署と従業員および場所のテーブル内に主キーが「0」のレコードがあるので、外部結合を使う必要はありません。Asset Managerは次のように外部結合のない、SQLコードを生成します。

```
SELECT l.name FROM amPortfolio a, amEmplDept e, amLocation l WHERE
a.lUserId = e.lEmplDeptId AND e.lLocaId = l.lLocaId
```

上記のクエリでは、**「ユーザ」**（User）と**「場所」**（Location）のリンクが、依然として部署と従業員または場所のテーブル内のレコードを指し示す（リンクがない場合は主キーが「0」のレコードを指し示す）ので、期待した結果が得られます。

結果

- 特に集計関数を使う場合などは、記述したクエリで前述のようなレコードの存在を考慮に入れることが重要になります。

例：

```
SELECT count(AssetTag) FROM amPortfolio
```

資産のテーブル内の資産数を数える上記のクエリを実行すると、主キーが「0」のレコードも結果に含まれます。そのため、データベース内の実際の資産数を求めるには、結果から1を引く必要があります。

- DBMSレベルの外部結合を生成する必要はほとんどありません。

注意:

実際にDBMSレベルの外部結合を処理する場合は、AQL演算子（「=*」および「*=」）を使います。

NULLの使い方

Asset Managerでは、次の2つのフィールドにのみDBMSのNULL値を使います。

- 空の「テキスト」型のフィールド
- 未入力の「日付」または「日付+時刻」型のフィールド

AQLでは、以下に示す複数のシンタックスを使うことができます。AQLはこれらのシンタックスを、ユーザのデータベースエンジンに有効なSQLコードに変換します。

空の「テキスト」型のフィールドについては、データベースにNULL値が保存されるので、次のいずれのシンタックスでも使うことができます。

WHERE <テキスト型フィールド> = NULL

WHERE <テキスト型フィールド> IS NULL

WHERE <テキスト型フィールド> = ''

未入力の「日付」または「日付+時刻」型のフィールドについては、データベースにNULL値が保存されるので、次のいずれのシンタックスでも使うことができます。

WHERE <日付または日付+時刻フィールド> = NULL

WHERE <日付または日付+時刻フィールド> IS NULL

WHERE <日付または日付+時刻フィールド> = []

注意:

「数値」型のフィールドが未入力の場合は、フィールド値は「0」になります。同様に、リンクが存在しない場合は「リンク = 0」または「外部キー = 0」で表します。例：「Location=0」 または「lLocaId=0」。

Self

Selfは、適用先のテーブルの表記文字列と同値です。

Selfを使うとクエリが簡単になり、Asset Managerデータベースのカスタマイズを活用できるようになります。

例：

部署と従業員のテーブルの表記文字列が次のように構成されているとします。

```
[Name], [FirstName], ([Phone])
```

AQLクエリとして以下のものを考えます。

```
SELECT self FROM amEmplDept
```

これは、次のクエリに相当します。

```
SELECT (((((Name + ',') + FirstName) + '(') + Phone) + ')') FROM amEmplDep
t
```

CurrentUser

「CurrentUser」を使うと、データベースに接続しているユーザに依存するクエリを記述できます。

「CurrentUser」は、クエリの式として、またはリンクとして使うことができます。この式はクエリエディタでは記述できないので、手動で入力する必要があります。

式として使う

例：データベースに接続している従業員が使うすべてのポートフォリオ品目を検索します。

```
SELECT lPortfolioItemId FROM amPortfolio WHERE User = CurrentUser
```

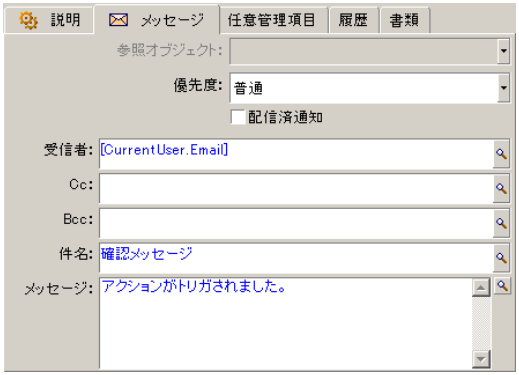
リンクとして使う

「CurrentUser」は、どのテーブルにもあって、[部署と従業員]テーブル内の現在接続しているユーザに対応するレコードを指し示すリンクであると見なすことが出来ます。

- 「CurrentUser」の形では、この関数は現在のユーザに対応するレコードを指し示します。
- 「CurrentUser.Field」の形では、この関数は現在のユーザの当該フィールドの値を返します。

例：データベースに接続しているユーザがあるアクションを引き起こした時に、状況に応じて別のメッセージタイプのアクションを引き起こして、そのユーザに

警告メッセージを自動的に送ることが出来ます。これを実現するにはアクションの詳細画面で、以下のように入力します。



システムリストデータ

AQLクエリでシステムリストデータを使う場合は、画面に表示される値ではなく、データベースに保存されている値を使う必要があります。

例：

次のクエリでは、［**タイプ**］（seType）フィールドが［**マスターリース**］に設定された契約を選択します。

```
SELECT Self FROM amContract WHERE seType = 1
```

［**タイプ**］（seType）フィールドはシステムリストデータです。データベースに保存される値は次のとおりです。

- 0：［**その他**］
- 1：［**マスターリース**］
- 2：［**リース明細**］
- 3：［**保険**］
- 4：［**メンテナンス**］

注意:

Asset Manager Application Designerを使用するか、またはデータベース構造を記述した「**Database.txt**」ファイルを参照すると、システムリストデータの値を調べることができます。

このファイルは、「［Asset Managerのインストール先フォルダ］/doc/infos」にあります。

階層構造をもつテーブル

階層構造のもつすべてのテーブルには、次のフィールドが含まれています。

- ［FullName］フィールド
- ［sLvl］フィールド

［FullName］フィールド

階層構造のテーブルの各レコードには［FullName］フィールドがあります。このフィールド値の前には、親レコード（ルートまで）のフィールド値で構成されるツリー構造のパスが付きます。

値はスペースなしのスラッシュ（/）記号で区切られます。ツリー構造の最初と最後にもスラッシュが入ります。

例：

- ［資産］テーブルの［FullName］フィールドでは、次のように現在の資産の資産タグ（CR012）の前に、親資産の資産タグが、その前に親資産の親資産の資産タグが置かれます。

FullName = '/PC118/DD054/CR012/'

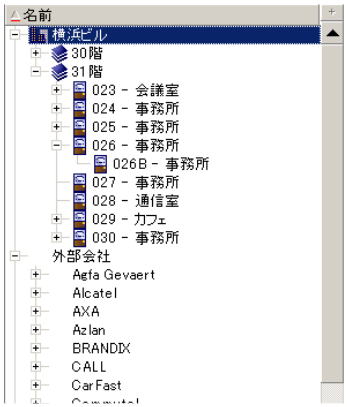
- ［場所］テーブルの［FullName］フィールドの場合も同様に、場所名（5階）の前に親場所名が付いた形式で保存されます。

FullName = '/東京/府中支社/府中ビル/5階/'

［sLvl］フィールド

階層構造のテーブルのすべてのレコードには、ツリー構造のレベルを示す［sLvl］フィールドがあります。

ルートはレベル0になります。



次のクエリは、「アジア地域営業部」のレコードとその付属部署を選択します。

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/アジア地域営業部/マーケティング/%') AND (sLvl >= 1)
```

次のクエリは、「アジア地域営業部」のレコードは選択しますが、その付属部署は除外します。

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/アジア地域営業部/マーケティング/%') AND (sLvl = 1)
```

次のクエリは「アジア地域営業部」のレコードの付属部署は選択しますが、「アジア地域営業部」のレコード自体は除外します。

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/アジア地域営業部/マーケティング/%') AND (sLvl > 1)
```

AQLの簡単な表記法

ここでは、AQLの記述を簡略化するための表記法を紹介します。

外部キー

SELECT句とORDER BY句以外の句では、最後にピリオド（.）がないリンクのSQL名は、関連付けられた外部キーのSQL名と同等に扱われます。

例えば次の句

```
WHERE location = 0
```

これは、次のクエリに相当します。

```
WHERE lLocaId = 0
```

この場合「location」は、［部署と従業員］テーブルを［場所］テーブルへつなぐ「場所」リンクのSQL名で、「lLocaId」は対応する［資産］テーブルの外部キーのSQL名です。

表記文字列

SELECT句とORDER By句では、最後にピリオドのないSQL名は<リンクのSQL名>.self結合と同等、つまり<リンクのSQL名>.<表記文字列>と同等に扱われます。

例：

部署と従業員のテーブルの表記文字列が次のように構成されているとします。

```
[Name], [FirstName] ([Phone])
```

AQLクエリは次のようになります。

```
SELECT user FROM amPortfolio
```

これは、次のクエリに相当します。

```
SELECT user.self FROM amPortfolio
```

これ自体は次と同等です。

```
SELECT (((((User.Name + ',') + User.FirstName) + '(') + User.Phone) + ')') FR
OM amPortfolio
```

任意管理項目

AQLでは、テーブルのフィールドに直接アクセスできるように、テーブルの任意管理項目にも直接アクセスできます。テーブルで任意管理項目の値を検索するには、任意管理項目のSQL名の前に「**fv_**」を付けます。

例：次のクエリは、［**部署と従業員**］（**amEmplDept**）テーブルでSQL名が「**fv_WorkUnit**」の任意管理項目の値を検索します。

```
SELECT fv_WorkUnit FROM amEmplDept
```

特殊フィールド

AQLでは、テーブルに関連付けられている特殊フィールドを活用できます。

特殊フィールドのSQL名は「**cf_**」で始める必要があります。

ソートとインデックス

AQLでソート（ORDER BY句）を使うクエリには、次の2とおりがあります。

- クエリで指定したインデックスをAsset Managerでそのまま使うモード。このモードでは、検索結果がそのまま順次表示されます。
- クエリで指定したインデックスをAsset Managerで使わないモード。この場合は、DBMSがデータのソートの仕方を決定します。

例

以下のクエリの場合について説明します。

```
SELECT lModelId, Brand FROM amModel ORDER BY Brand
```

- **インデックスの強制使用**しないアクセス：データベースエンジンは、クエリで指定した［Brand］（ブランド）のインデックスを使わずに全テーブルを走査します。クエリの条件を満たすすべてのデータ項目を検索してから「ブランド」を基準にして並べ替え、ユーザに送信します。結果は一定の時間が経過しないと表示されません。
- インデックスの強制使用をするアクセス：データベースエンジンは、［Brand］のインデックスを使って検出した結果をそのままの順番で即時表示します。

このため、最初のデータ項目はすぐに表示されますが、全体を処理する時間は長くなります。

インデックスの強制使用をする方法

インデックスを強制的に使う方法は、クエリを作成する方法によって変わります。

［リストの設定］メニューを使う

Asset Managerのリストごとにデータアクセスのタイプを設定できます。メインリストおよびタブページ内のリストで設定できます。手順は次のとおりです。

1 設定したいリストに移動します。
2 右クリックします。
3 ショートカットメニューから［**ユーティリティ／リストの設定**］を選択します。
4 ［**列／並べ替え**］タブページで［**インデックスの強制使用**］チェックボックスをオンにし、クエリで指定されたインデックスを使って実行したクエリ結果を、そのままの順番で即時に表示するよう指定します。別のタイプのアクセスを選択する場合は、このチェックボックスをオフにします。

AQLを使う

AQLで直接クエリを記述する場合は、FIRST_ROWS句を使ってインデックスを強制的に使うことができます。

例：

```
SELECT FIRST_ROWS AssetTag FROM amAsset ORDER BY AssetTag
```

注意:

任意管理項目テーブルの［seDataType］フィールドなどの、システムリストデータで並べ替えを行う際にインデックスを強制すると、うまく機能しないことがあります。

ソートの順序

ソートの順序は次の条件に左右されます。

- データベースエンジン
- インデックス使用の有無

Oracle

インデックスを強制的に使用する場合

- NULLのレコードは表示されません。
- ASCIIコードの値によって並べ替えるため、大文字と小文字を区別します（バイナリソート）。

インデックスを強制的に使用しない場合

- NULLのレコードが表示されます。
- Oracleでは大文字と小文字の区別はありません。

例

ソート

元のリスト	A B C D a b NULL NULL
インデックスを使ったリスト	A B C D a b
インデックスを使わないリスト	NULL NULL A a B b C D

Microsoft SQL Server

ソートの順序は、データベース作成時に設定したパラメータによって決まります。これらのエンジンでは、大文字と小文字を区別する、アクセント付きの文字も区別するなどの設定が可能です。

注意事項

複雑なクエリでは、インデックスを使う場合と使わない場合のどちらがパフォーマンスの点で優れているかを即時に判断するのは困難です。実際に使用の有無を決定する前に、テストを行うことをお奨めします。

特に、シンプルフィルタやクエリなどの直接的なフィルタや、アクセス制限による間接的なフィルタを適用したリストの場合は、インデックスのテストを行うようにしてください。

AQLシンタックス

AQLを使うには、SQL言語に関する知識が必要です。ただし、このガイドにはSQLのシンタックスに関する詳しい説明はありません。詳細については、他の参考資料を参照してください。

表記法

AQLのシンタックスの表記法

表 3.1. 表記法

[ ]	大括弧は、オプションの項目を示します。実際に項目を入力するときは、大括弧は必要ありません。
< >	大括弧は、オプションの項目を示します。実際に項目を入力するときは、大括弧は必要ありません。
|	縦線（パイプ文字）は、複数の項目候補を区切るのに使います。
...	この省略記号は直前の内容が繰り返されることを表します。
FROM	大文字の単語はそのまま記述します。

クエリのシンタックス

シンプルクエリ

SELECT [DISTINCT] [FIRST_ROWS] <選択リスト>
[FROM句 [献 51]]
[WHERE句 [献 52]]
[GROUP BY句 [献 53]]
[HAVING句 [献 53]]
[ORDER BY句 [献 54]]

サブクエリ

AQLでは、フィールドの代わりにサブクエリを使うことができます。

注意:

サブクエリでは、SELECTステートメントで式を1個だけ使うことができます。

(**SELECT** [DISTINCT] <式>
[FROM句 [献 51]]
[WHERE句 [献 52]]
[GROUP BY句 [献 53]]
[HAVING句 [献 53]]
)

警告:

サブクエリは括弧で囲む必要があります。

使用例

```
SELECT Self FROM amAsset WHERE mPrice >= (SELECT Max(mPrice)/2 F
ROM amAsset)
```

UNION型のクエリ

UNIONを使うと、次のように複数のクエリの結果を1つにまとめることができます。

SELECT <選択リスト>

[FROM句 [献 51]]

[WHERE句 [献 52]]

[GROUP BY句 [献 53]]

[HAVING句 [献 53]]

[**UNION** | **UNION ALL** | **INTERSECTS** | **MINUS**]

SELECT <選択リスト>

[FROM句 [献 51]]

[WHERE句 [献 52]]

[WHERE句 [献 52]]

[GROUP BY句 [献 53]]

[HAVING句 [献 53]]...]

[ORDER BY句 [献 54]]

クエリの要素

フィールドとリンク

クエリには、Asset Managerデータベース内のフィールドとリンクを指定します。

フィールド名は次の方法で指定できます。

- クエリの開始テーブルの場合は、そのテーブルの名前を指定する必要はありません。

[Link....[Link.]]<field>

ポートフォリオ品目（AmPortfolio）テーブルの例：

Model User.Name User.Location.Name

- フィールドを絶対参照で指定する場合は、次のいずれかの方法でフィールドが属するテーブルの名前を指定する必要があります。
  - **FROM**句でテーブルを宣言してテーブルの名前を指定します（エイリアスも可）。

    <table.[link...]field>

    <alias.[link...]field>
  - **FROM**句でテーブルを宣言しない代わりにコロン（:）を使います。

    <table.[link...]field>

    <table[_alias]:[link[_alias]...]field>

後半の2つの表記は、**FROM**句が使えない場合に便利です。

例えば、Asset Managerでクエリを記述する場合は、**WHERE**句しか使えません。クエリの開始テーブルは暗黙的に指定されます（フィルタを適用するテーブルをクエリの詳細画面の［テーブル］（**TableName**）フィールドで指定する場合は暗黙的な指定です）。これに対し、クエリで他のテーブルを使う必要がある場合は、コロン（:）を使って明示的に指定します。

定数

クエリで使用できる定数として、以下のシンタックスが有効です。

数値定数

小数点の区切り文字としてピリオド（.）を使います。

例：

12

52.23

テキスト型の定数

シングルクォートで囲みます。

例：

'コンピュータ'

'モニタ'

日付型または時刻型の定数

日付型または時刻型の定数は、シャープ（#）文字で囲みます。日付と時刻の形式は、次の規則に従います。

- 年は4桁で表します。
- 日付は年-月-日で表します。

- 時刻は時間-分-秒で表します。
- 24時間制を使います（A.M.またはP.M.を付けた12時間制は使いません）。
- 日付はスラッシュ（/）またはハイフン（-）で区切ります。
- 時刻はコロン（:）で区切ります。
- 月、日、時間、分、秒は、2桁で表します。
- 日付と時刻を合わせて表示する場合は、必ず日付の後に時刻を記述し、2つをスペースで区切ります。

例：

#yyyy-mm-dd hh:mm:ss#

#yyyy-mm-dd#

#hh:mm:ss#

#2004-01-01 01:00:03#

変数

定数を指定する代わりに（▶定数 [献 47]）、ユーザがこのクエリを使用するフィルタを選択する際にユーザが値を入力する変数を挿入できます。

変数を定義するにはさまざまなシンタックスを使用できます。

- $Variable [献 48]
- amDbVal() [献 48]

$Variable

次のシンタックスを使用する：

```
<フィールドまたはリンクのSQL名> = $<変数名>
```

依頼を実行する前に、クエリフィルタのユーザは［**<ラベル>**］を入力するように求められます。

<変数名>には空白を入れないでください。

amDbVal()

次のシンタックスを使用する：

```
amDbVal(<フィールドまたはリンクのSQL名>,'<ラベル>',<デフォルト値>)
```

依頼を実行する前に、クエリフィルタのユーザは［**<ラベル>**］を入力するように求められます。

入力フィールドには、デフォルトで**<デフォルト値>**が表示されます。**<デフォルト値>**は、▶定数 [献 47]セクションで説明されるシンタックスを使用する必要があります。

<フィールドまたはリンクのSQL名>にリンクを指定する場合は、**<デフォルト値>**にリンクされたレコードの主キーを含める必要があります。

<フィールドまたはリンクのSQL名>にシステムのデータリストを指定する場合は、**<デフォルト値>**にデータベースに格納された番号を含める必要があります。

式

式は、次の要素で構成されます。

- 定数
- フィールド
- 関数
- サブクエリ

これらの要素を演算子や括弧で結びつけると、複雑な式を構成できます。

比較の式のシンタックスは次のとおりです。

<式> <比較演算子> <式>

論理式のシンタックスは次のとおりです。

<比較式> <AND | OR> <比較式>

括弧を使って複数の論理式を組み合わせることができます。

演算子

論理演算子

論理演算子を使うと、2つの式を結合できます。

表 3.2. 論理演算子

演算子	説明
AND	論理的AND
OR	論理的OR

クエリを最適化するために、比較演算子を使える場合は論理演算子を使わない方がよい場合があります。以下の例は、［**割当**］フィールド（SQL名：seAssignment）が［**納品待ち**］または［**メンテナンスのため返却**］であるポートフォリオ品目を選択するクエリフィルタを、最適化する方法です。この2つのシステムリストデータの値は、それぞれ「3」と「4」です。

```
(seAssignment=3) OR (seAssignment =4)
```

システムリストデータの値が「4」までの場合は、上のクエリを次のように記述することもできます。

```
seAssignment >=3
```

比較演算子

比較演算子は、2つの式を比較する時に使います。

表 3.3. 比較演算子

演算子	説明
=	等しい
<>	等しくない
=!	
>	より大きい
<	より小さい
>=	以上
=<	以下
=*	右外部結合。この演算子は、AQLでリンクを処理する場合のみ使います。
*=	左外部結合。この演算子は、AQLでリンクを処理する場合のみ使います。
LIKE NOT LIKE	「=」演算子と同様に機能し、ワイルドカード文字を使うこともできます。 次のワイルドカード文字を使用できます。 パーセント（%）：任意の文字列に置き換わります。 アンダースコア（_）：任意の1文字に置き換わります。 次の指定は、データベースエンジンによって変わります（SQL Serverはサポートしますが、Oracle for WorkGroupsはサポートしません）。 [abc...]は任意の並び文字（間にスペースがない）を定義します。 [a-c]は、とりうる値の範囲を定義します。 DB2では、LIKE X演算子のXにSQL列名が含まれる場合は、この演算子を使うことはできません。この演算子で使用できるのは定数のみです。例えば、次のようなクエリはDB2で機能しません。 **SELECT** COL1, COL2 **FROM** TABLE1 **WHERE** COL1 LIKE COL2
IS NULL IS NOT NULL	フィールド値がNULLかどうかを調べます。 Asset Managerでは、空のテキストフィールドと、値が入力されていない**日付**または**日付+時刻**型フィールドにだけNULL値が許されています。

サブクエリ専用の演算子

次の演算子を使って、サブクエリの結果と値を比較できます。

- **= ANY（サブクエリ）**
- **= ALL（サブクエリ）**
- **= SOME（サブクエリ）**

例：

- 次のAsset Manager Exportクエリでは、「府中支社」で使用しているブランドのポートフォリオ品目のリストを取得できます（このクエリには関連場所は含まれません）。

```
SELECTDISTINCT lModelId, Model.Brand FROM amPortfolio WHERE Model.Brand = ANY (SELECT Model.lBrandID FROM amPortfolio WHERE Location.FullName LIKE '/府中支社')
```

このスクリプトの場合、リンク先レコードに従うということは、レコードの外部キーを使用することを意味します。

選択リスト

選択リストは、抽出または表示する項目を定義します。選択リストは、クエリ内のSELECTステートメントを指定します。

選択リストは、次のようにカンマで区切られた式で構成されます。

<式> [,<式>...]

各式をエイリアスにリンクできます。例：

```
SELECT MrMrs, (Name + FirstName) Identity FROM amEmplDept
```

選択リストは、出力クエリで出力する列名を指定する場合に特に便利です。

注意:

DBMSによっては、1つのSELECTステートメントに一定数の式しか記述できない場合があります。

FROM句

FROM句には、**SELECT**ステートメントの対象となるテーブルを指定します。

AQLでは、テーブル名のエイリアスを利用できます。

シンタックス

FROM <テーブル名> [テーブルのエイリアス] [, <テーブル名> [テーブルのエイリアス] ...]

クエリの開始テーブル

クエリの**FROM**句で最初に指定したテーブルが、クエリの開始テーブルになります。

テーブルが指定されていないフィールドをクエリに使った場合、AQLはそのフィールドがクエリの開始テーブルに属していると見なします。AQLの**FROM**句はSQLのFROM句とは異なります。

例えば次のサブクエリでは、AQLは［**資産**］テーブルの［**資産タグ**］フィールドを検索します。

```
SELECT AssetTag FROM amAsset
```

クエリ内のテーブル数

クエリで指定できるテーブル数は、使用しているDBMSによって異なります。

例：

- Oracle：テーブルをいくつでも使うことができます。
- Microsoft SQL Server：クエリで使えるテーブル数は16個までです。

警告:

特にクエリでリンクを使っている場合にクエリ内のテーブル数を数えるときは、暗黙的に指定されているテーブルも忘れずに数えるようにしてください。DBMSのSQLで追加の結合を生成する「fv_」表記（任意管理項目の値の検索）にも注意してください。同様に、特殊フィールドの「cf_」も追加の結合を生成できる表記です。

例

```
FROM amPortfolio
FROM amPortfolio a, amLocation l
```

次のクエリはすべて同じように機能します。

```
SELECT AssetTag FROM amAsset
SELECT a.AssetTag FROM amAsset a
SELECT amAsset.AssetTag FROM AmAsset
```

WHERE句

AQLの**WHERE**句は、SQLのWHERE句と同様に機能します。

WHERE句は、データベースから抽出する項目を検索条件として指定します。検索条件を記述するは、**HAVING**句を使うこともできます。

シンタックス

WHERE <検索条件>

検索条件の作成

多くの場合は、次の書式で条件を記述する必要があります。

```
<WHERE | HAVING> [NOT] <式> <比較演算子> <式>
<WHERE | HAVING> [NOT] <論理式>
<WHERE | HAVING> [NOT] <フィールド> [NOT] LIKE 'xxxxx'
<WHERE | HAVING> [NOT] <論理式> <AND | OR> <論理式>
<WHERE | HAVING> [NOT] <フィールド> IS [NOT] NULL
```

場合によっては、次のように複雑なクエリを記述する必要があります。

```
<WHERE | HAVING> [NOT] EXISTS (<サブクエリ>)
<WHERE | HAVING> [NOT] <式> [NOT] IN (<値のリスト> | <サブクエリ>)
```

```
<WHERE | HAVING> [NOT] <式> <比較演算子> <ANY | ALL> (<サブクエリ>)
```

GROUP BY句

AQLの**GROUP BY**句は、SQLのGROUP BY句と同等に機能します。

シンタックス

GROUP BY <集合関数を使わない式> [, <集合関数を使わない式>]...

注意事項

GROUP BY句はテーブルのサブセットを指定する時に使います。サブセットを**GROUP BY**句に指定するには、フィールド名などの式を使います。

SELECTステートメントの選択リストに集合関数を使うと、**GROUP BY**が各サブセットの結果の値を検索します。これで得られた結果を、**HAVING**句に使うことができます。

クエリで**GROUP BY**句を使うと、選択リストの各式でサブセットごとに1つの値が得られます。

GROUP BY - 例

次のクエリは、データベース内のブランドの総数を算出します。このクエリの場合Asset Managerは、1つのブランドに関連付けられている資産ごとに、1つのブランドインスタンスを返します。

```
SELECT Count(Model.Brand.Name) FROM amAsset
```

次のように**GROUP BY**句を使うと、ブランドのリストとブランドごとの資産数が算出されます。

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset GROUP BY Model.Brand
```

HAVING句

AQLの**HAVING**句は、SQLのHAVING句と同等に機能します。

シンタックス

HAVING <検索条件>

WHERE句との相違点

HAVING句は、**WHERE**句と同様に検索条件を指定する時に使います。ただし、HAVING句とWHERE句は次のように異なります。

- **HAVING**句では、選択リスト内の集合関数に適用する制限を指定します。この場合は、クエリの結果に表示される項目数は制限されますが、集合関数にリンクしている計算には影響しません。
- クエリで**WHERE**句を使う場合は、検索条件によって集合関数が計算に使う項目数は制限されますが、結果として得られる項目数に影響はありません。

例

WHERE句と**HAVING**句が同等に機能するクエリの例です。

次のクエリは、名前の文字列が文字「**B**」より大きいブランドのリストと、そのブランドごとの資産数を返します。

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset GROUP BY Model.Brand.Name HAVING Model.Brand.Name > 'B'
```

次の**WHERE**句を使ったクエリでも同じ結果が得られます。

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset WHERE Model.Brand.Name > 'B' GROUP BY Model.Brand.Name
```

HAVING句を使ったクエリの例

HAVING句では集合関数（**Count**など）を使うことができます。**WHERE**句では使えません。集合関数を使うと、次のクエリのように、複数の資産が存在するブランドをすべて検索できます。

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset GROUP BY Model.Brand.Name HAVING count(Model.Brand) > 1
```

ORDER BY句

AQLの**ORDER BY**句は、SQLのORDER BY句と同等に機能します。

項目の並べ替え順には、次のタイプがあります。

- 昇順：**ASC**。デフォルトの並べ替え順です。
- 降順：**DESC**。

シンタックス

ORDER BY <式> [**ASC** | **DESC**] [, <式> [**ASC** | **DESC**]...]

INSERT句

INSERT句は、データベースのテーブルに1つまたは複数のレコードを挿入します。

シンタックス

INSERT INTO <テーブル名> [テーブルのエイリアス] (<フィールド名> [, <フィールド名>]...) **VALUES** (<式> [, 式]...)| **AQLサブクエリ**)

この句はAsset Manager API 「AmDbExecAql」に含まれています。

Asset Manager APIの詳細については、『Programmer's reference』ガイドの「**Alphabetic reference**」の章を参照してください。

例

INSERT句を使うと、**受領に関する補足情報**ウィザードのコードを簡略化できます。

INSERT句を使用しないウィザードのコード

```
hrAlarm = AmCreateRecord("amDateAlarm")
lErr = AmSetFieldLongValue(hrAlarm, "bSecondLevel", 0)
lErr = AmSetFieldLongValue(hrAlarm, "dtTrig1", AmGetFieldLongValue(hr
Asset, 2)-lDaysBefore*86400)
lErr = AmSetFieldLongValue(hrAlarm, "lAction1Id", lActionId)
lErr = AmSetFieldLongValue(hrAlarm, "lMonitObjId", lAstId)
lErr = AmSetFieldStrValue(hrAlarm, "MonitoredField", "dWarrEnd")
lErr = AmSetFieldStrValue(hrAlarm, "MonitoredTable", "amAsset")
lErr = AmSetFieldLongValue(hrAlarm, "sDaysBefore1", lDaysBefore)
lErr = AmInsertRecord(hrAlarm)
```

INSERT句を使用したウィザードのコード

```
lErr = AmDbExecAql("insert into amDateAlarm (bSecondLevel, dtTrig1, lAct
ionId, lMonitObjId, MonitoredField, MonitoredTable, sDaysBefore1) values (
0, " & AmGetFieldLongValue(AmGetFieldLongValue(hrAsset, 2)-lDaysBefore
*86400 & ", " & lAstId & ", 'dWarrEnd', 'amAsset', " & lDaysBefore & ")")
```

UPDATE句

UPDATE句は、データベースのテーブルのレコードのフィールドを更新します。

シンタックス

UPDATE <テーブル名> [テーブルのエイリアス] **SET** (<フィールド名> [, <フィールド名>...]) [FROM句 [献 51]] [WHERE句 [献 52]]

例

UPDATE句を使うと、コマンドアクションを起動するアクションのコードを簡略化できます。

UPDATE句を使用しないアクションのコード

```
hr = AmGetRecordFromMainId("amPOrder", [lPOrdId])
lErr = AmSetFieldLongValue(hr, "seStatus", "$(IDS_POSTATUS_ORDERED
)")
lErr = AmUpdateRecord(hr)
```

UPDATE句を使用したアクションのコード

```
lErr = AmDbExecAql("update amPOrder set seStatus = 21 where lPOrdId = "
&
[lPOrdId])
```

DUPLICATE句

DUPLICATEは、データベースのテーブルのレコードを複製します。

この関数はAsset Manager特有のものです。

詳細については、『**はじめに**』ガイドの「Asset Managerを初めて使用する」の章の**「レコードの処理」／「レコードを複製する」**を参照してください。

シンタックス

DUPLICATE <テーブル名> [テーブルのエイリアス] **SET** (<フィールド名> [, <フィールド名>...]) [FROM句 [献 51]] [WHERE句 [献 52]]

DELETE句

DELETE句は、データベースのテーブルのレコードのフィールドを削除します。

シンタックス

DELETE [FROM句 [献 51]] [WHERE句 [献 52]]

AQL関数

次のAQL関数は、クエリと計算式で使うことができます。

- 集合型のAQL関数
- 文字列型のAQL関数

- 日付型のAQL関数
- 数値型のAQL関数
- テスト型のAQL関数

注意:

使用中のDBMS専用のSQL関数を使うこともできます。この場合、他のデータベースエンジンにコードを移植することはできません。

集合型のAQL関数

表 3.4. 集合型のAQL関数

関数	説明
Avg(<列>)	数値型の列の平均を返します。列に値がない場合は「0」を返します。
Count(<列>)	列内の非NULL値をカウントします。
Countdistinct(<列>)	列内の異なる非NULL値をカウントします。
Max(<列>)	「数値」、「テキスト」、「日付」型の列の最大値を返します。 列にレコードがない場合は、「0」（「数値」型の列）、「空の文字列」（「テキスト」型の列）、または「空の日付」（「日付」型の列）を返します。
Min(<列>)	「数値」、「テキスト」、「日付」型の列の最小値を返します。 列にレコードがない場合は、「0」（「数値」型の列）、「空の文字列」（「テキスト」型の列）、または「空の日付」（「日付」型の列）を返します。
Sum(<列>)	数値型の列の値の合計を返します。列に値がない場合は「0」を返します。

上記の関数には、GROUP BY句とHAVING句を組み合わせて使います。

文字列型のAQL関数

表 3.5. 文字列型のAQL関数

関数	説明
Ascii(<文字列>)	<文字列>の先頭の文字のASCIIコードを返します。
Char(<n>)	ASCIIコード「n」の文字を返します。
Length(<文字列>)	<文字列>の長さを返します。

関数	説明
CharIndex(<文字列1>, <文字列2>)	<文字列2>内部の<文字列1>の位置を返します。<文字列1>の最初の文字は、位置1にあります。<文字列2>内に<文字列1>がない場合は、関数が0を返します。
DateToText(<日付>)	<日付>をテキスト型（ISO国際形式の日付を記述する文字列）に変換します。
EmptyString( )	非NULLの空の文字列を作成します。
LikeParam（<フィールド>）	2つのフィールドをAQLオペランドLikeで比較するために使用します。例： フィールド1 like LikeParam(フィールド2)
Left(<文字列>, <n>)	<文字列>の先頭の「n」文字を返します。
Lower(<文字列>)	<文字列>を小文字で返します。
Ltrim(<文字列>)	<文字列>の左側のスペースを削除します。
NullBlob( )	NULLのBLOBを作成します。 **注意:** **プログラム上、BLOBオブジェクトにキャストされたNULLを表します。**
NullMemo( )	NULLのメモを作成します。 **注意:** **プログラム上、メモオブジェクトにキャストされたNULLを表します。**
NullString( )	NULLの文字列を作成します。 **注意:** **プログラム上、文字列オブジェクトにキャストされたNULLを表します。**
NumberToText（ <数値> ）	<数値>をテキストデータ型に変換します。
Right(<文字列>, <n>)	<文字列>の末尾の「n」文字を返します。
Rtrim(<文字列>)	<文字列>の右側のスペースを削除します。
Substring(<文字列>, <n1>, <n2>)	<文字列>の第"n1" 文字から始まる"n1" 文字の部分文字列を抽出します（<文字列>の先頭文字を第1文字とします）。
TimeStampToText（<日付+時刻> ）	日付+時刻型のオブジェクトを次の国際規格に合うテキストデータ型に変換します。 YYYY-MM-DD HH24:MI:SS
TimeToText(<時刻>)	時刻型のオブジェクトを以下のの国際規格に合うテキストデータ型に変換します。 HH:MM:SS
Upper(<文字列>)	<文字列>を大文字で返します。

日付型のAQL関数

表 3.6. 日付型のAQL関数

関数	説明
AddDays(<日付>, <数値>)	「日付」または「日付+時刻」型のフィールドに与えられた日数を加えます。
AddHours(<日付>, <数値>)	「日付」または「日付+時刻」型のフィールドに与えられた時間数を加えます。
AddMinutes(<日付>, <数値>)	「日付」または「日付+時刻」型のフィールドに与えられた分数を加えます。
AddSeconds(<日付>, <数値>)	「日付」または「日付+時刻」型のフィールドに与えられた秒数を加えます。
Day(<日付>)	「日付」または「日付+時刻」型のフィールドで、その月の何日目かを表す数字を返します（1-31）。
DayOfYear(<日付>)	「日付」または「日付+時刻」型のフィールドで、その年の何日目かを表す数字を返します（1-366）。
DaysDiff(<日付1>, <日付2>)	日付1から日付2までの日数（10進の浮動小数点数）
DbToLocalDate(<日付>)	データベースサーバのタイムゾーンの日付を、クライアントマシンで定義したタイムゾーンの日付に変換します。
Getdate()	サーバの現在のシステム日付を返します。
Hour(<時刻>)	「時刻」または「日付+時刻」型のフィールドの「時」を表す数字を返します（0-23）。
HoursDiff(<日付1>, <日付2>)	日付1から日付2までの時間数（10進の浮動小数点数）
LocalToDbDate(<日付>)	クライアントマシンのタイムゾーンの日付を、データベースサーバのタイムゾーンの日付に変換します。
Minute(<時刻>)	「時刻」または「日付+時刻」型のフィールドの「分」を表す数字を返します（0-59）。
MinutesDiff(<日付1>, <日付2>)	日付1から日付2までの分数（10進の浮動小数点数）
Month(<日付>)	「日付」または「日付+時刻」型のフィールドの月を表す数字を返します（1-12）。
NullDate( )	NULL日付を作成します。 **注意:** **プログラム上、日付オブジェクトにキャストされたNULLを表します。**
NullTime( )	NULL時刻を作成します。 **注意:** **プログラム上、時刻オブジェクトにキャストされたNULLを表します。**

関数	説明
NullTimeStamp( )	NULL日付+時刻を作成します。 **注意:** **プログラム上、日付+時刻オブジェクトにキャストされたNULLを表します。**
NumberToTime（ <数値> ）	数値を「日付+時刻」型のデータに変換します。
Second(<時間>)	「時刻」または「日付+時刻」型のフィールドの「秒」を表す数字を返します（0-59）。
SecondsDiff(<日付1>, <日付2>)	日付1から日付2までの秒数（10進の浮動小数点数）
TextToTime(<テキスト>, <フォーマット>, <言語>)	テキストを日付に変換します。追加（およびオプション）の言語とフォーマットのパラメータを使用すると、データベース固有のフォーマットにアクセスできるようになり、変換予定のフォーマットと言語サポート設定を指定できるようになります。詳細については、データベースのドキュメントを参照してください。
WeekDay(<日付>)	「日付」または「日付+時刻」型のフィールドの曜日を表す数字を返します。 この数字は、サーバの設定によって変わります。例えば、Microsoft SQL Serverのデフォルトの設定は（1=日, 2=月, ..., 7=土）、Oracleのデフォルトの設定は（1=月, ..., 7=日）です。
Year(<日付>)	「日付」または「日付+時刻」型のフィールドの年を表す数字を返します（2000など）。

表 3.7. 日付型のAQL関数

説明	Asset Managerクエリ言語
過去1週間に変更したすべてのレコード	AddDays(dtLastModif,7)>=Getdate()
過去1時間に報告があったすべての作業指示	HoursDiff(Getdate(), dtNotif) <= 1 または AddHours(dtNotif, 1) >= Getdate()
過去30分間にクローズ済になったすべての作業指示	MinutesDiff(Getdate(), dtActualFixed) <= 30 または AddMinutes(dtActualFixed, 30) >= Getdate()

次のクエリは、作業を実際に行った日付と終了した日付が同じである作業指示を検索します。クライアントマシンのタイムゾーンを使います。

```
SELECT Self FROM amWorkorder WHERE DayOfYear(DbToLocalDate(dtA
ctualFixStart)) = DayOfYear(DbToLocalDate(dtActualFixed))
```

次のクエリは、今日の日付で実際に開始した作業指示を検索します。

```
SELECT Self FROM amWorkorder WHERE DayOfYear(DbToLocalDate(dtA
ctualFixStart)) = DayOfYear(DbToLocalDate(GetDate()))
```

数値型のAQL関数

表 3.8. 数値型のAQL関数

機能	説明
Abs（ <数値> ）	「数値」の絶対値を返します。
Ceil（ <数値> ）	「数値」以上で一番小さい整数を返します。
DataLength (<データ>)	<データ>の長さをバイトで返します。
Floor（ <数値> ）	「数値」以下で一番大きい整数を返します。
Length(<データ>)	<データ>の長さを文字数で返します。
Mod(<a>, <b>)	「a」を「b」で割った余りを返します(a = qb + r、ただしqは整数で0 <= r < q)。
NullNumeric( )	NULL数値を作成します。 **注意:** **プログラム上、数値オブジェクトにキャストされたNULLを表します。**
NumberToNumber（ <数値> ）	RDBMSレベルで実行されるのではない場合に、異なるタイプの数値間の変換を行います。
Round(<a>, <n>)	「a」を小数第「n」位未満四捨五入します。
Sign（ <数値> ）	<数値>パラメータの符号を判定することができます。 ■ 関数が1を返す場合、<数値>は正の数です。 ■ 関数が-1を返す場合、<数値>は負の数です。 ■ 関数が0を返す場合、<数値>は無効（=0）です。
TextToNumber（ <テキスト> ）	<テキスト>を数値データ型に変換します。
Trunc(<a>, <n>)	「a」を小数第「n」位未満切り捨てにします。

適用例

Abs (2.516) = 2.

Ceil (2.516) = 3.

Floor (2.516) = 2.

Mod (6,4) = 2.

Round (31.16, 1) = 31.20.

Round (31.16, 0) = 31.00.

Round (31.16, -1) = 30.00.

Trunc (31.16, 1) = 31.1.

テスト型のAQL関数

表 3.9. テスト型のAQL関数

関数	説明
IsNull(<a>, <b>)	「a」がNullの場合は「a」を「b」で置き換えます。「a」と「b」のデータ型に互換性がなければなりません。

クエリの例

ここでは、各例につきクエリ作成の1側面を説明します。次の例を参考にして独自のクエリを作成してください。

ここで紹介する例では、完全なシンタックスでクエリを記述しています。これらのクエリをそのままの形でテストしたい場合は、Asset Manager Export Toolを使います。これらのクエリをクエリフィルタなどで使用するには、シンタックスを変更する必要があります。

例えば、次のような明示的に記述したクエリがあるとします。

```
SELECT self FROM amAsset WHERE Model.Brand.Name='Compaq'
```

これを資産のテーブルのクエリフィルタで使うには、次のようにWHERE句だけを明示的に指定したシンタックスに変更します。

```
Model.Brand.Name='Compaq'
```

Asset Managerプログラムに付属しているデモ用データベースに格納されているクエリも、参照してください。

注意:

使用中のDBMSのSQLコードに書き換えられたクエリは、クエリの詳細画面の［プレビュー］タブページに表示されます。

メインテーブルのフィールドを特定の値と比較する

例：「Compaq」ブランドのポートフォリオ品目すべて

```
SELECT Self FROM amPortfolio WHERE Model.Brand.Name = 'Compaq'
```

メインテーブルのリンクを別のリンクと比較する

例：親資産と場所が同じであるすべてのポートフォリオ品目

```
SELECT Self FROM amPortfolio WHERE Location = Parent.Location
```

メインテーブルのリンクを特定の値と比較する

例：「大阪支社」の直下にあるすべての部署と従業員

```
SELECT Self FROM amEmplDept WHERE Parent.Name = '大阪支社'
```

メインテーブルにリンクしたテーブルのフィールドの値に応じて比較する

例：親資産と場所名が同じであるすべてのポートフォリオ品目

```
SELECT Self FROM amPortfolio WHERE Location.Name = Parent.Location.
Name
```

階層構造をもつテーブル

[FullName] フィールドを使う

例：「大阪支社」という名前の場所の、すべての従属場所

```
SELECT Self FROM amLocation WHERE FullName LIKE '/大阪支社/%'
```

[FullName] フィールドと [sLvl] フィールドを使う

階層構造をもつテーブルのクエリでは、[FullName] フィールドと [sLvl] フィールドがよく使用されます。

例：「大阪支社」場所に関連する従属場所で、階層のレベルが3より低いレコード

ツリー構造のルートの階層レベルは「0」です。

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/大阪支社/%') AN
D (sLvl < 3)
```

フルネームの先頭と末尾にスラッシュ（/）記号が付くことに注意してください。

2つの条件を組み合わせたクエリ

例：役職が「経理部長」で場所「大阪事務所」に位置するすべての従業員

```
SELECT Self FROM amEmplDEpt WHERE (Title = '経理部長') AND (Locatio
n.Name = '大阪事務所')
```

数値、日付、テキスト型フィールドの比較

例：2006年1月1日から2006年12月31日までに実行したすべての作業指示

```
SELECT self FROM amWorkOrder WHERE (dtActualFixStart >= #2006-01-01 00:00:00#) AND (dtActualFixStart <= #2006-12-31 00:00:00#)
```

任意管理項目に関するクエリ

例：［**fv_Size**］（SQL名）任意管理項目の値が150 cm以上であるすべてのポートフォリオ品目

```
SELECT Self FROM amPortfolio WHERE fv_Size >= 150.00
```

式でレコードを検索する

例：購入価格が、データベース内で最高の購入価格に等しいすべての資産。メインクエリ内にサブクエリを使って最高価格を指定します。

```
SELECT Self FROM amAsset WHERE mPrice = (SELECT max(mPrice) FROM amAsset)
```

未入力のフィールドを検索する

例：電話番号のないすべての従業員。空の文字列をシングルクォートで囲みます。

```
SELECT Self FROM amEmplDept WHERE Phone=''
```

リンクなしのレコードを検索する

1リンクの場合

例：ユーザに割り当てられていないすべてのポートフォリオ品目。リンクがないことを「0」で表します。

```
SELECT Self FROM amPortfolio WHERE User = 0
```

nリンクの場合

例：資産にリンクしていないすべてのモデル

```
SELECT self FROM amModel WHERE NOT ( EXISTS (SELECT A1.lAstId FROM amAsset A1 WHERE A1.lModelId = amModel.lModelId))
```

上記のクエリは、モデルのテーブルを走査して各モデルに属する資産の数を0と比較します。

1リンクとnリンクのテストを組み合わせた例

例：親モデルも従属モデルもないすべてのモデル

```
SELECT self FROM amModel WHERE (NOT ( EXISTS (SELECT A1.lModelId FROM amModel A1 WHERE A1.lParentId = amModel:lModelId))) AND (Parent = 0)
```

上記のクエリでは次の処理を行います。

- 1リンク（Parent = 0）のテスト。親資産のないモデルを選択します。
- nリンク（0 = (SELECT COUNT(a.lModelId) FROM amModel a WHERE a.lParentId = lModelId)）のテスト。従属モデルのないモデルを選択します。nリンクのテストは、各モデルの識別子［lModelId］を選択し、［lParentId］識別子が［lModelId］に等しいすべてのモデルをカウントします。

注意:

SELECT COUNT句は、すべてのデータベースレコードを数えるのでEXIST句よりもパフォーマンスを消費します。

別の組み合わせの例

「コンピュータ」属性の従属モデルがないすべてのモデル

```
SELECT self FROM amModel p WHERE NOT ( EXISTS (SELECT lModelId FROM amModel WHERE (FullName LIKE (p.FullName + '%/')) AND (Nature.Name = 'コンピュータ')))
```

注意:

Asset Manager Export Toolを使ってこのクエリを試行すると、エラーメッセージが表示されますが、これは無視してください。クエリは正しく機能しています。

エイリアスを使ったクエリ

例：「HP Software」研修プログラムと「データベース」研修プログラムを受講したすべての従業員

開始テーブル：［部署と従業員］テーブル

クエリは次のとおりです。

```
SELECT Self FROM amEmplDept WHERE (Trainings_1.Title = 'HP Software') AND (Trainings_2.Title = 'データベース')
```

Training_1とTraining_2と表記したエイリアスによって、Trainingリンクで関連付けられている2つの異なるレコードに2つの条件を定義できます。

以下のように記述すると、

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'HP Software')
AND (Trainings.Title = 'データベース')
```

クエリは、2つの名前がついた1つの研修コースを受講したすべての従業員を選択します。

以下のように記述すると、

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'HP Software')
OR (Trainings.Title = 'データベース')
```

クエリは、2つの名前の内いずれかの研修コースを1つ受講したすべての従業員を選択します。

クエリを短くするために、コロン文字を使用できます。

```
SELECT amPortfolio.self, amModel_FullName:self FROM amPortfolio portfo
lio
```

このクエリと同等のものを次に示します。

```
SELECT amPortfolio.self, FullName.semf FROM amPortfolio portfolio, amM
odel FullName
```

変数を含めたクエリ

シンタックス1

例えば、ユーザがクエリを使用するフィルタを選択する際にユーザが指定する値と等しい名前を持つすべてのモデル。

開始テーブル：［**Models**］（amModel）。

```
名前 = $Nom
```

依頼を実行する前に、クエリフィルタのユーザは［**名前**］フィールドを入力するように求められます。

ユーザが「**Test**」と入力する場合、生成されるクエリは次のとおりです。

```
SELECT M1.Name FROM amModel M1 WHERE M1.Name = 'Test'
```

シンタックス2

例えば、ユーザがクエリを使用するフィルタを選択する際にユーザが指定する値を含む名前を持つすべてのモデル。

開始テーブル：［**Models**］（amModel）。

```
Name LIKE amDbVal(Name, 'Name contains', 'Specify the value that the fiel
d must contain')
```

依頼を実行する前に、クエリフィルタのユーザは［**Name contains**（次の文字を含む名前）］フィールドを入力するように求められます。

［**Name contains**（名前に次の文字を含む）］フィールドには、デフォルトで「**Specify the value that the field must contain**（フィールドに含める必要がある値を指定する）」が表示されます。

ユーザが「**Test**」と入力する場合、生成されるクエリは次のとおりです。

```
SELECT M1.Name FROM amModel M1 WHERE M1.Name LIKE 'Test' ESC
APE '\'
```

4 Crystal Reports

この章では、Asset ManagerでCrystal Reportsを使用して、レポートを表示して印刷する方法について説明します。

制限

Crystal Reportsレポートの使用方法は、レポートへのアクセスがWebクライアント経由であるかWindowsクライアント経由であるか、また、使用するレポートがAsset Managerデータベースに格納されているかCrystal Reports Serverデータベースに格納されているかによって異なります。

▸制限は、『はじめに』ガイドの**「Windowsクライアントと Webクライアントの違い」**の章の**「レポート機能」**セクションで説明します。

概要

Crystal Reportsを使用する理由

Crystal Reportsは、レポート（**.rpt**ファイル）の作成と変更に使用されます。

Crystal Reportsで作成された**.rpt**ファイルは、Asset Manager Windowsクライアント、またはWebクライアントで表示して印刷できます。

Asset Managerで、Crystal Reportsレポートを作成したり変更できません。

レポートの格納先

.rptファイルは、以下のさまざまな方法で格納できます。

- **Asset Managerデータベース**で。

  この方法で格納されたレポートは、Windowsクライアントでのみアクセスできます。

- **Crystal Reports Serverデータベース**で。

  この方法で格納されたレポートは、Windowsクライアント、およびWebクライアントでアクセスできます。

  ユーザがAsset Manager WebクライアントまたはWindowsクライアントからCrystal Reportsレポート（表示または印刷）を要求すると、クエリがCrystal Reports Serverに送信されます。

  Crystal Reports ServerはAsset Managerデータベースからデータを取得し、WebクライアントやInternet Explorer（Windowsクライアント）にレポートを表示します。

レポートの参照場所

各**.rpt**ファイルは、格納方法に関わらず［**レポート**］（amReport）テーブルのレコードで参照されます。

- レポートが**Asset Managerデータベース**に格納されている場合、「**.rpt**」ファイルはWindowsクライアント経由でデータベースにインポートされます（レポート詳細ページの［**インポート**］ボタン）。
- レポートが**Crystal Reports Serverデータベース**に格納されている場合：**.rpt**ファイルは、レポートの詳細ページの［**ファイル**］（FileName）から参照されます。

レポートのタイプ

レポートにはいくつかのタイプがあります。

- **詳細レポート**

  これらのレポートは、指定したテーブルのレポート情報を表示するために使用します。

  これらのレポートは、レポートのコンテキストテーブルが選択されている場合のみアクセスできます。

  これらのレポートは、Windowsクライアントでのみ利用できます。

- **リストレポートとグラフ**

  これらのレポートは、指定したテーブルの選択したレコードの情報を表示するために使用します。

  これらのレポートはいつでもアクセスできます。

これらのレポートは、WindowsクライアントとWebクライアントで利用できます。

Asset Manager WindowsクライアントがAsset Managerデータベース内に格納されたレポートにアクセスできるようにするための インストールと設定

各Windowsクライアントコンピュータ上に、以下のアプリケーションのいずれか1つをインストールする必要があります。

- Crystal Reportsの完全バージョン。

  サポート対象のCrystal Reportsバージョンを確認するには、▶Asset Managerサポート対応表を参照してください。

- **Asset Managerユーティリティ**CDに収録されているCrystal Reports（Designer）for Asset ManagerのOEM（オリジナルの機器の製造元）バージョン。
- Asset Manager付属の制限バージョン（ランタイム）

  このランタイムをインストールするには、Asset Managerをインストールするときに該当するオプションをチェックします。

  このランタイムバージョンを使用することで、Webクライアントではなく Windowsクライアントを使用して、Asset Managerデータベースの現在のデータを含む既存レポートをプレビューして印刷できます。新規レポートの作成や既存レポートを変更をすることはできません。

ヒント:

既にインストールされている場合、またはCrystal Reportsの完全製品バージョン版またはOEMバージョンをインストールする場合は、Asset ManagerにCrystal Reportsの制限付きバージョンをインストールする必要はありません。

Asset Manager WebクライアントがCrystal Reports Serverデータベースに格納されているレポートにアクセスできるようにするためのインストールと設定

前提条件

Asset Managerデータベースの作成

選択したDBMSを使用してAsset Managerデータベースを作成しておく必要があります。

このデータベースは、有効なHP AutoPassライセンスキーを使用してアクティブにしておく必要があります。Crystal Reports Serverからもアクセスできる必要があります。

▶ Asset Manager『管理』ガイドの「**ライセンスキーをインストールする**」の章

インストールが必要なコンポーネント

以下のコンポーネントをインストールします。

- 以下のAsset Managerモジュール：
  - Asset Manager Windowsクライアント
  - Asset Manager ODBCドライバ

  ▶ Asset Manager『**インストールとアップグレード**』ガイドとAsset Manager『管理』ガイドの「**ODBCドライバを使ったデータベースへのアクセス**」の章。
- Asset Managerデータベースで使用されるDBMSのクライアント層

  ▶ Asset Manager『管理』ガイド、「**Asset Managerデータベースの作成、変更、削除**」の章、「**DBMSを使って空のシェルを作成する**」のセクション。

Asset Managerデータベースへの接続の作成

1. Asset Manager Windowsクライアントを起動します。
2. Asset Managerデータベースへの接続を作成します（［**ファイル／接続の管理...**］メニュー）。

   ▶ Asset Manager『管理』ガイド、「**Asset Managerデータベースの作成、変更、削除**」の章、「**Asset Managerへの接続を作成する**」のセクション。

   接続の名前を覚えておきます。後で必要になります。
3. 接続をテストします（［接続の編集］ウィンドウの［**テスト**］ボタン）

データベースへのポートフォリオ - 専門分野データのインポート

統合が正しく機能するためには、**ポートフォリオ - 専門分野データ**をインポートする必要があります。

▶ 『ポートフォリオ』ガイドの「**ITポートフォリオ**」の章の「**コンピュータ／Asset ManagerからのHP Discovery and Dependency Mapping Inventoryデータベース内のコンピュータの表示／予備タスク／データベースへのポートフォリオ－専門分野データのインポート**」のセクション。

Crystal Reports ServerとCrystal Reports（Designer）のインストール

Crystal Reports ServerとCrystal Reports（Designer）のインストール方法には、次のいくつかの方法があります。

- **Asset Managerユーティリティ** CDに収録されている**OEMバージョン**（オリジナルの機器の製造元）を使用する方法。

  無人インストールまたは手動インストールを実行できます。
- 既に所有している**完全製品バージョン**を使用する方法。

  ▶Crystal Reportsドキュメント

  サポート対象のCrystal Reportsバージョンを確認するには、▶Asset Managerサポート対応表を参照してください。

注意:

Crystal Reports Serverで特別な設定は必要ありません。

各レポートについて設定を行います。

注意:

Asset Managerで使用できるのはWindows上にインストールされているCrystal Reportsのみです。これは、Crystal ReportsがAsset Manager ODBCドライバを使用するためです。

ここでは、提供されている**Asset Managerユーティリティ**から、Crystal Reports for Asset ManagerおよびCrystal Reports Server for Asset ManagerのOEMバージョンをインストールする方法について説明します。

重要項目:

Asset Managerに付属しているBusiness Objects製品（Crystal Reports for Asset ManagerとCrystal Reports Server for Asset Manager）は、BTO（Business Technology Optimization）製品でのみ使用できます。

提供されたBusiness Objects for Crystal Reports Serverライセンスには、**20 CAL**が付属しています。同時にCrystal Reports Serverを使用することが認められているユーザは、**20ユーザ**のみです。

Crystal Reports for Asset ManagerとCrystal Reports Server for Asset Managerのインストール（無人モード）

無人モードのインストールは、事前に定義されたオプションを使用して自動的に実行されます。

1 提供された**Asset Managerユーティリティ**CDのルートに保存されている「AssetManagerReportingInstall.bat」スクリプトを起動します。
2 次のいずれかを選択できます。
   - Crystal Reports Serverのインストール
   - Crystal Reports Designerのインストール

   オプションを選択します。
3 次のいずれかを選択できます。
   - Unattended mode installation（無人モードインストール）
   - Manual installation（手動インストール）

   Unattended mode installaion（無人モードインストール）を選択します。
4 インストールは事前に定義されたオプションを使用して実行されます。場合によって、インストールプロセスが中断することがあり、その場合は手動で続行する必要があります。

これらの手順を実行して、両方の製品（Crystal Reports for Asset ManagerとCrystal Reports Server for Asset Manager）をインストールします。

Crystal Reports Server for Asset ManagerとCrystal Reports for Asset Managerを手動でインストール

Crystal Reports Server for Asset ManagerおよびCrystal Reports for Asset Manager、またはそのいずれかのインストールをカスタマイズする場合は、手動インストールを実行する必要があります。

詳細については、Business Objectsの各ガイド（http://support.businessobjects.com/documentation）を参照してください。

ドキュメントサーバにアクセスできない場合は、Crystal Reports Server for Asset ManagerインストールCD-ROM内の「Doc」フォルダに格納されているバージョンメモと英語版インストールガイドを参照してください。

Crystal Reports Server for Asset ManagerまたはCrystal Reports for Asset Managerの手動インストールを実行するには：

1 提供された**Asset Managerユーティリティ**CDのルートに保存されている「AssetManagerReportingInstall.bat」スクリプトを起動します。
2 次のいずれかを選択できます。
   - Crystal Reports Serverのインストール
   - Crystal Reports Designerのインストール

   オプションを選択します。
3 次のいずれかを選択できます。
   - Unattended mode installation（無人モードインストール）

- Manual installation（手動インストール）

  Manual installation（手動インストール）を選択します。

4 次に、インストールウィザードに表示される手順に従います。

これらの手順を実行して、両方の製品（Crystal Reports for Asset ManagerとCrystal Reports Server for Asset Manager）をインストールします。

Crystal Reports Server for Asset Managerの**手動**インストールを実行するには、追加の構成手順を手動で実行する必要があります。▶ 構成（OEMバージョンまたはCrystal Reports Server完全製品バージョンの手動インストール） [献 75]

構成（OEMバージョンまたはCrystal Reports Server完全製品バージョンの手動インストール）

提供されたOEMバージョンからCrystal Reports Server for Asset Manager の無人モードインストールを実行した場合は、このセクションで説明する構成手順を実行する必要はありません（構成情報は、インストール時に自動的に入力されています）。

ただし、提供されたOEMバージョンからCrystal Reports Server for Asset Managerの手動インストールを実行した場合、または Crystal Reports Serverの完全製品バージョンをインストールした場合は、統合のため、追加の構成手順が必要になります。

「u2lamlib.dll」ファイル

- Asset ManagerとCrystal Reports Serverを同じサーバ上にインストールしている場合は、「u2lamlib.dll」ファイルに対する構成は必要ありません。
- Asset ManagerとCrystal Reports Serverを同じサーバ上にインストールしない場合：
  1 エクスプローラを起動します。
  2 **Asset Manager Utilities（Asset Managerユーティリティ）**CD内の「lib\XX」フォルダの内容を表示します（XXはAsset Managerのインストール言語です）。
  3 「u2lamlib.dll」ファイルをコピーします。
  4 「<Crystal Reports Serverのインストールフォルダの完全パス>\Common\3.5\bin」フォルダにこのファイルを貼り付けます。例えば、「C:\Program Files\Business Objects\Common\3.5\bin」フォルダに貼り付けます。

Asset ManagerレポートのCrystal Reports Serverへのインポート

Asset ManagerレポートをCrystal Reports Serverにインポートする方法はいくつかあります。

Asset Manager「.rpt」ファイルの手動によるインポート

このタイプのインポートを実行するには、Crystal Reports Serverをインストールし、レポートにアクセスするためにに Asset Managerをホスティングしているサーバにアクセス可能である必要があります。

この場合は次のようになります。

1 Crystal Report Server ［**Publishing Wizard**（公開ウィザード）］を開始します（Windowsメニューの［スタート／ プログラム／**Business Objects**／**Crystal Reports Server**／**Publishing Wizard**（公開ウィザード））。
2 インポートするレポートを選択し、Crystal Reports Server上でのレポートの格納場所を指定して、ウィザードを完了します。

提供されたBIARファイルを使ったレポートのインポート

- 提供されたIEMバージョンを使用してCrystal Reports Server for Asset Managerの無人モードインストールを実行した場合は、「deployReports.bat」スクリプト（**Asset Manager**ユーティリティCDの「install」フォルダ内に保存されている）を使用してCrystal Reports Serverデータベースの BIAR（Business Intelligence Archive Resource）ファイルを展開することで、Asset Managerが自動的にインポートされます。

  注意:

  このスクリプトは、 Crystal Reports Serverをデフォルトの Asset Managerパラメータを使用してインストールした場合にのみ動作します。

- それ以外の場合は、Crystal Reports Serverのインポートウィザードを使用してください。
  1 Crystal Report Server ［**Import Wizard**（インポートウィザード）］を開始します（Windowsメニューの［スタート／ プログラム／**Business Objects**／**Crystal Reports Server/Import Wizard**（インポートウィザード））。
  2 ［ソース環境］ページで、ソースとして「**Business Intelligence Archive Resource**（**BIAR**）」ファイルを選択し、 **Asset Manager**ユーティリティCD上のこのファイルの保存場所（**<Media>\lib\reports\am_reports_EN.biar**）を選択します。
  3 ［**Destination environment**（ターゲット環境）］ページで、Crystal Reports Serverデータベースを入力します。
  4 ［**Select objects to import**（インポートするオブジェクトの選択）］ページで、以下を選択します。
    - ［**Import users and users groups**（ユーザおよびユーザグループのインポート）］オプションを選択します。ただし、サブオプションを選択しないでください。

- ［**Import folders and objects（フォルダおよびオブジェクトのインポート）**］オプションを選択します。ただし、サブオプションを選択しないでください。
- ［**Import repository objects（レジストリオブジェクトのインポート）**］オプションを選択します。
- 他のすべてのオプションをクリアします。

5 Crystal Reports Serverがユニバースをサポートしていないことを通知する警告メッセージが表示されます。この警告は無視してください。

6 ［**Incremental import（インクリメンタルインポート）**］ページで、すべてのオプションを選択します。

7 ［**Users and groups（ユーザおよびグループ）**］ページで、Asset Managerユーザを選択します。

8 ［**Folders and objects（フォルダおよびオブジェクト）**］ページで、Asset Managerフォルダを選択します（フォルダに含まれるすべてのサブフォルダが選択されます）。

9 ウィザードを終了します。

これで、すべてのAsset ManagerレポートとAsset ManagerユーザがCrystal Reports Serverデータベースにインポートされました。

Crystal Reports ServerのAsset Managerユーザの作成

- 提供されたBIAR（Business Intelligence Archive Resource）ファイルを使用してレポートをインポートした場合、Asset Manager Crystal Reports Serverユーザもインポートされます。
- それ以外の場合は、以下を実行できます。
  - ユーザを手動で作成できます。▶ 手動による作成 [献 77]
  - シングルサインオンプロセスによってユーザを自動的に作成されるようにできます。▶ シングルサインオンプロセスによる自動作成 [献 79]

手動による作成

Asset Manager Crystal Reports Serverユーザを手動で作成するには、次の2つの手順を実行します。

1 ユーザの作成 [献 77]

2 権限の割り当て [献 78]

ユーザの作成

Asset Manager Crystal Reports Serverユーザを作成するには：

1 Crystal Reports Server Java Administration Launchpadを開始します（Windowsメニューの［**スタート／プログラム／Business Objects／Crystal Reports Server／Java Administration Launchpad**］）。

2 ［**Central Management Console**（集中管理コンソール）］リンクをクリックして、自分自身を確認します。
3 ［**ユーザ**］（［**Organize**（整理）］グループ）をクリックします。
4 ユーザ管理ウィンドウが表示されます。［**New User...**（新規ユーザ）］をクリックします。
5 新規ユーザ作成ウィンドウが表示されます。以下のパラメータを入力します。
   - Authentication Type（認証タイプ）：Enterprise
   - アカウント名：AssetManager
   - 完全名：AssetManager
   - パスワード：任意のパスワードを選択します
   - ［**Password never expires**（パスワードは失効しません）］を選択します。
   - その他のすべてのパスワード構成オプションをクリアにします。
   - ［**Concurrent User**（同時使用ユーザ）］接続タイプを選択します。

   ［**OK**］をクリックします。これで、**AssetManager**マネージャが作成されました。

権限の割り当て

Asset Managerレポートはあらかじめ Crystal Reports Serverにインポートしておく必要があります。

Asset Managerレポートに対するすべての権限をAssetManagerユーザに割り当てるには：

1 Crystal Reports Server **Central Management Console**］モジュールを開始します。（Windowsメニューの［**スタート／プログラム／Business Objects／Crystal Reports Server／Java Administration Launchpad**］、次に［**Central Management Console**］リンク）。
2 ［**フォルダ**］（［**整理**］グループ）をクリックします。
3 レポートをエクスポートしたフォルダ（［**Asset Manager**］など)をクリックします。
4 ［**権限**］タブで、**Asset Manager**の権限が表示されます。
5 ［**Add/Remove...**（追加/削除...）］をクリックします。
6 ［**Select Operation**（操作の選択）］フィールドで、［**Add users**（ユーザの追加）］を選択します。

   次に、［**Available users**（使用可能なユーザ）］フィールドで、［**AssetManager**］ユーザを選択し、［**>**］をクリックします。

   これで、［**AssetManager**］ユーザが「**Asset Manager**」フォルダに関連づけられました。

   ［**OK**］をクリックします。

7 ［**AssetManager**］ユーザの場合、［**Access Level**（アクセスレベル）］列にあるドロップダウンリストで、［フルコントロール］を選択します。

8 ［更新］をクリックして変更を保存します。

これで、［**AssetManager**］ユーザにAsset Managerレポートに対する完全なコントロールを割り当てました。

シングルサインオンプロセスによる自動作成

まだAsset Manager Crystal Reports Serverユーザが作成されていない場合で、シングルサインオンが実装されている場合（▶シングルサインオンの設定 [献 82]）は、ユーザがレポートへの接続を試みると、シングルサインオン機構によってユーザが自動的に作成され、適切なグループに追加されます。
「reporting.properties」ファイルで以下のパラメータが定義され、デフォルトで以下のように設定されます。

- ユーザ名：**AssetManager**
- グループ：**Administrator**

これは、BIARファイルによってインポートされた構成にも対応します。

これらのパラメータは、「reporting.properties」ファイルを編集し、Crystal Reports Serverデータベース内の対応する情報を修正することで変更できます。

Asset Managerレポートのソースデータベースの更新

Asset Managerレポートのソースデータベースは、レポートデータを格納しているデータベースに対応します。

レポートのソースデータの更新方法はいくつかあります。

「.rpt」ファイルを使用してのソースデータベースの変更

1 Crystal Reports（Designer）を起動します。

2 修正する「.rpt」ファイルを開きます。

3 「データベース」メニューで、［**Set Datasource Location**（データベースソースの保存場所の設定）］を選択します。

4 ［**Unfold reports**（レポートの表示）］と［**Asset Manager Databases**（**Asset Managerデータベース**）］を選択します。

5 新しいデータ：**<User ID>@<Base>**（**Admin@AMDemo51en**など）を挿入して、［**User ID**（ユーザID）］フィールドを編集します。

6 パスワードの入力を要求するメッセージが表示されます。

パスワードを入力します（ユーザが**Admin**の場合、パスワードはデフォルトで空欄になります）。

次に、［開く］をクリックします。

7 ここで、**ODBC** （**RDO**）から、データベースのパスワードの入力を要求するメッセージが表示されます。

パスワードを入力します（ユーザが**Admin**の場合、パスワードはデフォルトで空欄になります）。

次に、［**完了**］をクリックします。

8 サブレポート用のソースデータベースがある場合は、各データベースについて同じ操作を実行します。

9 ウィンドウを閉じます。

Crystal Reports Server ［**Central Management Console**（集中管理コンソール）］モジュール介してのソースデータベースの手動による変更

Crystal Reports Server **Central Management Console**］モジュールを開始します。（Windowsメニューの［**スタート／プログラム／Business Objects／Crystal Reports Server／Java Administration Launchpad**］、次に［**Central Management Console**］リンク）。

各Asset Managerレポートを順番に開いて、各レポートについて、以下の操作を実行します。

1 ［**Use custom database logon information specified here**（ここで指定したカスタムデータベース情報を使用する）］オプションを選択し、ODBCドライバを選択します。

2 ［**サーバ**］フィールドに、**Asset Managerデータベース**を入力します。

3 ［**データベース**］フィールドに、データベース名（**AMDemo51en**など）を入力します。

4 ［**Prompt the user for database logon**（データベースへのログオンをユーザに要求するメッセージを表示する）］オプションを選択します（このオプションは、シングルサインオン機構で必要です）。

ソースデータベースの自動変更

Asset Managerレポートはあらかじめ Crystal Reports Serverにインポートしておく必要があります。

ソースデータベースを自動的に変更するには：

1 **Asset Managerユーティリティ**CDをCD-ROMドライブに挿入します。

2 CDの「install」フォルダを検索および指定し、DOSコマンドプロンプトを開きます（右クリックして［**Open Command Window Here**（ここでコマンドウィンドウを開く）］）を開きます。

3 Crystal Reports Serverをデフォルトフォルダ（C:\Program Files\Business Objects）にインストールしていない場合は、次のコマンドを入力して、Crystal Reports Serverのインストールフォルダを定義してください。

```
set BOBJ_InstallDir=<Crystal Reports Serverインストールフォルダ>
```

例：

```
set BOBJ_InstallDir=C:\my directory\Business Objects
```

4 **「Asset Manager」**フォルダ以外のフォルダにレポートをインポートした場合は、次のコマンドを入力してレポートをインポートしたフォルダを定義してください。

```
set AMRootReportsFolder=<reports import folder>
```

例：

```
set  AMRootReportsFolder=Asset Manager Reports
```

5 例えば、「buildReportsList.bat」スクリプトを実行して、Crystal Reports Server上のAsset Managerレポートリストをコンパイルします。デフォルトで、このスクリプトの結果は、「C:\am_reports.dat」ファイルに保存されます。この保存場所、または結果のファイル名を変更する場合は、次のコマンドを入力します。

```
set ReportsListFile=<path and file name>
```

例：

```
set ReportsListFile=C:\temp\asset_manager_reports.dat
```

6 次に、「buildReportsList.bat」スクリプトを実行するために次のコマンドを入力します。

```
buildReportsList.bat
```

「buildReportsList.bat」は、ファイルを生成し、実行が完了するとファイルの名前を表示します。

7 Asset Manager データベースへのインポートからレポートを除外するには、「buildReportsList.bat」によって生成されたファイルを開いて、除外する各レポート行の先頭に**#**文字を挿入します。

8 DOSコマンドプロンプトで、データのソース言語を定義するために次のコマンドを入力します。

```
set BOBJ_ClientLanguage=<Asset Manager database language>
```

次の2文字の言語コードのいずれかを使用して言語を選択します。**EN**、**FR**、**DE**、**ES**、**JP**、**CHS**、**CHT**、**KO**、**NL**、**SV**、**IT**、 **PT**。

9 エクスプローラを起動します。

10 「\lib\reports」フォルダに変更します。

11 「am_reportsDSDef_xxx.dat」ファイルをコピーします。この場合、**xxx**はCrystal Reportsインストール言語です。

12 これを任意のフォルダに貼り付けます。

13 「am_reportsDSDef_xxx.dat」ファイルを編集します。

14 Asset Managerデータベースへの接続名を指定して、［**データベース**］の後のパラメータを入力します（▶ Asset Managerデータベースへの接続の作成 [献 72]）。

その他のパラメータは変更しないでください。

15 DOSコマンドプロンプトウィンドウに戻ります。

次のコマンドを入力して、変更したばかりの「am_reportsDSDef_xxx.dat」ファイルを参照します。

```
set DataSourceDefFile=<Full name of the modified am_reportsDSDef_xxx
.dat file>
```

16 「changeReportsDS.bat」スクリプトを実行するために次のコマンドを入力します。

```
changeReportsDS.bat
```

17 これで、Crystal Reports Server内の各Asset Managerレポートにカスタムソースデータベースが関連付けられました。

シングルサインオンの設定

何度も認証が必要になることがないようにするには（Asset Manager Webクライアントへの接続時、レポートへのアクセス時、またはレポートがAsset Managerデータベースからデータを取得する必要があるとき）、シングルサインオンを設定する必要があります。

シングルオンによって、ユーザはすべてのサービスへのアクセスが可能になり、何度も認証情報を入力する必要がなくなります。

Crystal Reports Server for Asset Managerを無人モードでインストールした場合は、既にシングルサインオンが自動設定されています。

ただし、手動インストールを実行した場合は、次の操作を実行してシングルサインオンを手動で設定する必要があります。

1 「am-reporting.common-51.jar」と「am-reporting.serverside-51.jar」ファイル（**Asset Managerユーティリティ**CDの「lib\sso」フォルダに保存されている）をCrystal Reports Server for Asset Manager Webアプリケーションの「lib」フォルダにコピーします。例： C:\Program Files\Business Objects\Tomcat\webapps\businessobjects\enterprise115\desktoplaunch\WEB-INF\lib

2 Crystal Reports Webアプリケーションの「web.xml」ファイル（C:\Program Files\Business Objects\Tomcat\webapps\businessobjects\enterprise115\desktoplaunch\WEB-INFに保存されている）を開きます。

   「reporting-SsoWeb.xml」ファイル（**Asset Managerユーティリティ**CDの「lib\sso」フォルダに保存されている）の内容をこのファイルにコピーします。

3 「reporting.properties」ファイル（**Asset Managerユーティリティ**CDの「lib\sso」フォルダに保存されている）をCrystal Reports Webアプリケーションの「WEB-INF/classes」フォルダにコピーします。

   例：

C:\Program Files\Business Objects\Tomcat\webapps\businessobjects\enterprise115\desktoplaunch\WEB-INF\classes

このファイルを編集して、パラメータをカスタマイズできます。例えば、以下のパラメータをカスタマイズできます。

パラメータ名	デフォルト値	パラメータ説明
server.bo.shared_secret	cradmin	シングルサインオン用の**共有シークレット**パスワード。
server.bo.admin_account	Administrator	Crystal reports管理者アカウント（ユーザを仮に作成する場合にのみ使用）。
server.bo.cms_port	6400	Crystal Reports Serverデータベースポート番号。
server.bo.default_user	AssetManager	このパラメータを定義すると、Crystal Reports Serverへの接続は、シングルサインオン機構によって提供されたユーザ名でなく、このユーザ名で作成されます。 ユーザ名が存在しない場合、**on_the_fly_creation**パラメータが有効であればCrystal Reports Serverへの接続が仮に作成されます。それ以外の場合は、接続に失敗します。
server.bo.user.on_the_fly_creation	true	Crystal Reports Serverのユーザの仮作成を許可または却下します（ユーザがCrystal Reports Serverデータベース内に存在しないと、最初の接続時にユーザがCrystal Reports Serverデータベース内に作成されないことがあります）。
server.bo.user.can_change_password	true	ユーザが仮に作成されるときのユーザのCrystal Reportsプロパティ。
server.bo.user.must_change_password	false	Idem
server.bo.user.password_expires	false	Idem
server.bo.user.is_named	false	Idem
server.bo.user.password	Welcome	仮に作成されたユーザのデフォルトユーザパスワード。
server.bo.user.description	user auto-created by the Hp Ers Sandbox	仮に作成されたユーザのデフォルトユーザ説明。

4 シングルサインオン用のパスワードの設定：
   a Crystal Reports集中管理コンソールを開きます（Windowsメニューの［**スタート／プログラム／Business Objects／Crystal Reports Server／Java Administration Launchpad**］）。
   b ［**Central Management Console（集中管理コンソール）**］をクリックします。
   c デフォルトで**Administrator**としてログインするため、パスワードは必要ありません。
   d ［**管理**］セクションで、［**認証**］をクリックします。
   e ［**Enterprise（エンタープライズ）**］タブで、［**Trusted Authentication is enabled（信頼済み認証を有効にする）**］オプショ

ンを選択し、「reporting.properties」ファイルで使用されているのと同じパスワードで、**server.bo.shared_secret**変数にあらかじめコピーしておいたパスワードを使って、［**共有シークレット**］フィールドを入力します。デフォルトで、**cradmin**が設定されています。

f ［**更新**］をクリックします。

警告:

Asset Managerアプリケーションで使用しているホストがCrystal Reports Serverのホストと同じDNSドメインを使用しない場合、シングルサインオン機構は機能しません。これは、クッキーのプロパゲーションなどのインターネット規格による特定の制限によるものです。

両方のホストが異なる2つのDNSドメイン内にある場合、Asset Managerに接続するとき、Crystal Reports Serverに接続するとき、Asset Managerデータベースに接続するときのいずれの接続時にもユーザは認証される必要があります。

また、シングルサインオンを正しく機能させるために、以下にもドメイン名を入力する必要があります。

- Crystal Reports Serverサーバ名：Crystal Reports Serverの宣言 [献 85]
- Asset Manager Web Tierサーバ名：**http://<ドメイン名を指定したHP Asset Manager Web Tierサーバ名>:<HP Asset Manager Web Tierで使用されるポート>/AssetManager**

sysCoreWebCrystal特殊フィールドの設定

sysCoreWebCrystal特殊フィールドを変更するには：

1 Asset Manager Windowsクライアントを起動します。

2 ［**特殊フィールド**］画面を開きます（［**管理／Calculated fields...**（**特殊フィールド...**）］メニュー）。

3 ［**レポートを表示**］（sysCoreWebCrystal）特殊フィールドを選択します。

以下のいずれかの場合は、この特殊フィールドに使用するスクリプトを変更できます。

Crystal Reports Serverの旧バージョンを使用している場合

Asset Managerと共に提供されるバージョンより以前のバージョンを使用している場合は、**sysCoreWebCrystal**特殊フィールドに使用するスクリプトを次の方法で変更してください。

```
bHasOldCrystal = TRUE
bHasSSO = FALSE
```

Crystal Reports Serverの旧バージョンではシングルサインオン（SSO）機構を使用することはできません。

シングルサインオンを使用する必要がない場合

シングルサインオン（SSO）機構を使用する必要がない場合は、**sysCoreWebCrystal**特殊フィールドを修正して、この機構を無効にすることができます。

```
bHasOldCrystal = FALSE
bHasSSO = FALSE
```

この場合、Asset ManagerアプリケーションはCrystal Reports Serverと直接やりとりするため、Crystal Reports ServerおよびAsset Managerデータベースに接続する際にはユーザは認証される必要があります。

Crystal Reports Serverの宣言

1. Asset Manager WindowsクライアントまたはWebクライアントを起動します。
2. データベースに接続します。
3. ［**アプリケーションサーバのURLアドレスの編集...**］ウィザード（BstBackEndOpt）を起動します（ナビゲーションバーの［**管理／アプリケーションサーバのURLアドレスの編集...**］リンク）。
4. 次のようにウィザードに入力します。

フィールド	値
［**アプリケーションサーバのプロパティの追加/変更**］ページ	
追加するアプリケーションサーバ名	ページの下部分で［**アプリケーション**］列が［**Crystal**］である行が存在しない場合、このフィールドに**Crystal**と入力し、［**アプリケーションサーバの追加**］ボタンをクリックします。
アプリケーションのリスト、［**アプリケーション**］列が［**Crystal**］である行、［**サーバのURLアドレス**］列	httpまたはhttps://<Crystal Reports Serverサーバ名>:<Crystal Reports Serverで使用するポート番号> 例： http://BusinessObjectServer:8088 警告: **シングルサインオンを正しく機能させるには、Crystal Reports Serverサーバ名にドメイン名を入力する必要があります。**
［**変更**］ページ	ページの内容を確認します。
［**完了**］をクリックします。	

フィールド	値
Windowsクライアント：	［**OK**］をクリックします

5 Windowsクライアント：データベースに再接続します（［**ファイル／データベースに接続**］メニュー）。

レポートの表示

レポートを表示するには

レポートの表示方法は、レポートのタイプ（詳細、一覧、グラフ）によって異なります。

詳細レポート

警告:

詳細レポートはWindowsクライアントで利用できますが、Webクライアントでは利用できません。

詳細レポートはさまざまな方法で表示できます。

ナビゲーションバーで選択する方法によるレポートの表示

1 レポートの対象であるレコードを含むリストを表示します。
2 レポートの対象であるレコードを選択します。

重要項目:

1つのレコードのみを選択できます。

3 レポートの機能ドメインに対応するナビゲーションバーを展開します。

ヒント:

レポートの機能ドメインは、レポートの詳細の［ドメイン］（Domain）フィールドで定義します。

4 ナビゲーションバーでレポート名をクリックします。

レポート一覧からのレポートの表示（Windowsクライアントのみ）

1 レポートを表示します（ナビゲーションバーの［**管理／レポート機能／レポート**］リンク）。

2 レポートの詳細を表示します。
3 ［**プレビュー**］をクリックします。
4 レポートの対象であるレコードを選択します。
5 ［**OK**］をクリックします。

［ファイル／印刷］メニューを使用したレポートの表示（Windowsクライアントのみ）

1 レポートの対象であるレコードのリストを表示します。
2 レポートの対象であるレコードを選択します。
3 ［**ファイル／印刷**］メニューを選択します。
4 ［**タイプ**］フィールドに値［**詳細レポート(Crystal Reports)**］を入力します。
5 ［**レポート**］リストでレポートを選択します。
6 ［**プレビュー**］をクリックします。

リストレポートとグラフ

リストレポートまたはグラフは、さまざまな方法で表示できます。

ナビゲーションバーで選択する方法によるレポートの表示

1 レポートの機能ドメインに対応するナビゲーションバーを展開します。

> ヒント:
>
> **レポートの機能ドメインは、レポートの詳細の［ドメイン］（Domain）フィールドで定義します。**

2 ナビゲーションバーでレポート名をクリックします。

レポート一覧からのレポートの表示（Windowsクライアントのみ）

1 レポートを表示します（ナビゲーションバーの［**管理／レポート機能／レポート**］リンク）。
2 レポートの詳細を表示します。
3 ［**プレビュー**］をクリックします。

［ファイル／印刷］メニューを使用したレポートの表示（Windowsクライアントのみ）

1 ［**ファイル／印刷**］メニューを選択します。
2 ［**タイプ**］フィールドに「**規定レポート（Crystal Reports）**」を入力します。
3 ［**レポート**］リストでレポートを選択します。

4 ［**プレビュー**］をクリックします。

自動更新によるレポートの表示（Windowsクライアントのみ）

1 ［**ツール／レポート機能／Crystal Reports統計**］メニューを選択します。
2 レポートの一覧にフィルタをかける場合は、［**属性**］フィールドに入力します（［**属性**］フィールド右にあるリスト）。
3 ［**属性**］フィールド右側のリストでレポートを選択します。
4 レポートを更新するには：アイコンをクリックします。

   レポートの自動更新の頻度を設定するには：アイコンを右クリックします。

   拡大率（3レベル）を変更するには：アイコンをクリックします。

利用可能なレポート

WindowsおよびWebクライアントでレポートを使用するには、レポートが以下の条件を満たす必要があります。

条件	Windowsクライアント	Webクライアント
レポートを格納しているデータベース	■ Asset Managerデータベース ■ Crystal Reports Serverデータベース	Crystal Reports Serverデータベース
	警告: Crystal Reports Serverが［アプリケーションサーバのURLアドレスの編集...］ウィザード（BstBackEndOpt）で宣言されている場合、およびレポートの［ファイル］フィールド（database）に数値が含まれている場合、WebクライアントおよびWindowsクライアントは、Crystal Reports Serverを介してレポートが利用できると判断します。	
レポートのタイプ	■ 詳細レポート 警告: ナビゲーションバーにレポートを表示するには、有効なコンテキストを選択する必要があります。 ■ 一覧レポート ■ グラフ	■ 一覧レポート ■ グラフ

条件	Windowsクライアント	Webクライアント
［メニューに表示］（bVisible）チェックボックス	チェック	チェック

レポートの表示方法

Windowsクライアント：

- 「.rpt」ファイルがAsset Managerデータベースに格納されている場合：レポートはWindowsクライアントによって表示されます。
- 「.rpt」ファイルがAsset Managerデータベースではなく、Crystal Reportsサーバに格納されている場合：レポートはInternetブラウザによって表示されます。

Webクライアント：レポートはWebクライアントに表示されます。

Asset ManagerによるレポートのURLアドレスの作成方法

Asset Managerは、［**レポートを表示**］（sysCoreWebCrystal）特殊フィールドを使用して、ラベルがレポートのラベルである**HREF**属性を持つHTML **<A>**アンカーを生成します。これが、次の項目を連結することで生成されるURLアドレスをポイントします。

- ［**アプリケーションサーバのURLアドレスの編集...**］ウィザード（BstBackEndOpt）で定義されたCrystal Reports ServerのURLアドレス
- 特殊フィールドのスクリプトで指定されたテキスト文字列
- レポートの［**ファイル**］（FileName）フィールド

レポートに続く名前のあるリンクをクリックすると、Asset ManagerによりURLアドレスが処理されます。

注意:

デフォルトで、Business Object Crystal Report Server Ver.11にアクセスするように［レポートを表示］（sysCoreWebCrystal）特殊フィールドが構成されます。

Ver.11以前のバージョンを使用する場合、生成されたリンクは動作しないときには特殊フィールドを手動で変更する必要があります。

1 ［レポートの表示］（sysCoreWebCrystal）特殊フィールドの詳細を表示します（［管理／**Calculated fields...**（特殊フィールド**...**）］メニュー）。
2 スクリプトで、次のURLを置換します。

```
"/businessobjects/enterprise115/desktoplaunch/opendoc/openDocument.jsp
?iDocID="
```

置換の対象：

```
"/businessobjects/viewers/crystalreportviewers115/htmlviewerbridge.jsp?i
d="
```

3 変更を保存します。

レポートを利用可能にする

WindowsまたはWebクライアントでレポートを使用できるようにするには、レポートの「.rpt」ファイルを作成し、次のいずれかのデータベースに格納する必要があります。

- Asset Managerデータベース
- Crystal Reports Serverデータベース

Crystal Reportsレポートを格納するデータベースを決定するには：▶ レポートの格納先 [献 70]。

この章には、以下の点に関する情報が含まれます。

- 「.rpt」ファイルの取得 [献 90]
- Asset Managerデータベースへのレポートの保存 [献 91]
- Crystal Reports Serverデータベースへのレポートの格納 [献 93]

「.rpt」ファイルの取得

Crystal Reportsは、以下に挙げる複数のソースに基づきます。

- Crystal Reportsを使用してユーザが作成したレポート。
- Asset Managerデータベースに既に格納されているレポート。

これらのレポートをCrystal Reports Serverデータベースに転送する場合は、レポートの詳細ページで［**エクスポート**］ボタンをクリックして、Windowsクライアントで1つずつエクスポートする必要があります。

- Asset Managerの専門分野データレポート。

  これらのレポートは、Asset Managerインストールフォルダの「datakit」サブフォルダに配置されています。

注意:

使用可能なレポートのコメント付きリストは、Asset Managerのインストール先フォルダの「datakit\standard\reports」サブフォルダの「reports.txt」ファイル内にあります。

Crystal**レポートは、「\datakit\standard\reports\rpt」フォルダにあります。**

「reports.txt」ファイルはレポートのインポートスクリプトに使用されるため、このファイルの構造は変更しないでください。ただし本番用データベースへ一部のレポートのみをインポートする場合は、インポートスクリプトを実行する前に、このファイルの全部の行を削除できます。ここに別のレポートを追加することもできます。

Asset Managerデータベースへのレポートの保存

この方法で格納されたレポートは、Windowsクライアントでのみアクセスできます。

既存Asset Managerデータベースに1つずつレポートをインポートする

レポートを1つずつ挿入するには：

1. Asset Manager Windowsクライアントを起動します。
2. データベースに接続します。
3. レポートを表示します（ナビゲーションバーの［**管理／レポート機能／レポート**］リンク）。
4. 新しいレポートを作成します。
5. ［**インポート**］をクリックします。
6. Asset Managerのインストール先フォルダのサブフォルダ「\datakit\standard\reports\rpt」で、必要な「.rpt」拡張子のファイルを選択します。

新規データベース作成時におけるAsset Managerの全レポートのインポート

データベース作成時にレポートをインポートするには、

1. Asset Manager Application Designerを起動します。
2. ［**ファイル／開く**］メニューを選択します。

3 ［**データベース記述ファイルを開く（新規データベースの作成）**］オプションを選択します。
4 Asset Managerのインストール先フォルダのサブフォルダ「config」に格納されている「gbbase.xml」を選択します。
5 データベース作成ウィザードを開始します（［**アクション／データベースの作成**］メニュー）。
6 『**管理**』ガイドの「**Asset Managerデータベースの作成、変更、および削除**」の章の指示に従います。

   ページ［**インポートするデータ**］で、［**Crystal Reports**］を選択します。

既存のデータベースにAsset Managerの全レポートのインポート

レポートを既存のデータベースにインポートするには：

1 Asset Manager Application Designerを起動します。
2 ［**ファイル／開く**］メニューを選択します。
3 ［**データベース記述ファイルを開く（新規データベースの作成）**］オプションを選択します。
4 Asset Managerのインストール先フォルダのサブフォルダ「config」に格納されている「gbbase.xml」を選択します。
5 データベース作成ウィザードを開始します（［**アクション／データベースの作成**］メニュー）。
6 ウィザードのページに次のように入力します（ウィザードページ間の移動は［**次へ**］ボタン、［**戻る**］ボタンを使用します）。

   ［**SQLスクリプトの生成/データベースの作成**］ページ：

フィールド	値
データベース	レポートのインポート先にするデータベースへの接続を選択します。
作成	専門分野データをインポートします。
高度な作成オプションを使用	このオプションは選択しません。

   ［**作成パラメータ**］ページ：

フィールド	値
パスワード	管理者のパスワードを入力します。 **注意:** Asset Managerデータベース管理者は、［部署と従業員］（amEmplDept）テーブルに記録され、その［部署名/姓］（Name）フィールドは**Admin**になります。 データベース接続ログインは、［ユーザ名］（UserLogin）フィールドに保存されます。管理者名は「**Admin**」です。 パスワードは［パスワード］フィールド(LoginPassword)に保存されます。

［インポートするデータ］ページ：

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

7 ウィザードを使用して定義されたオプションを実行します（［**完了**］ボタン）。

Crystal Reports Serverデータベースへのレポートの格納

この方法で格納されたレポートは、Windowsクライアント、およびWebクライアントでアクセスできます。

ユーザが利用できるようにする各レポートについて、次の手順に従います。

1 Crystal Reports ServerのCentral Management Consoleモジュールを起動します。
2 レポートをインポートします。
3 レポートを開きます（［**Objects**］メニュー）。
4 以下のプロパティに注意します。
   - 名前
   - ファイル

     例：

     ```
     frs://Input/a_117/002/000/629/1f048f4996817a4.rpt
     ```

     この例では、値**629**が存在することが分かります。

これは、Asset Managerデータベースに格納する必要があるレポートの識別子です。

ヒント:

レポートの識別子を得るためには、Crystal Reports Serverクエリビルダで、以下のようなクエリを実行することもできます。

```
SELECT SI_NAME, SI_ID FROM CI_INFOOBJECTS WHERE SI_PROGID = 'CrystalEnterprise.Report'
```

「buildReportsList.bat」によって生成されたファイルを使用できます： ▶ ソースデータベースの自動変更 [献 80]。

5 ［**プロセス**］タブ、［**データベース**］サブタブに入力します。

プロパティ名	値
サーバ	Asset Manager Databases
データベース	Asset Manager接続の名前
ユーザ	このプロパティは空欄にします
パスワード	このプロパティは空欄にします

6 Asset Manager WindowsクライアントまたはWebクライアントを起動します。
7 データベースに接続します。
8 レポートを表示します（ナビゲーションバーの［**管理／レポート機能／レポート**］リンク）。
9 次のフィールドに入力してレポートを作成します。

フィールドまたはリンクの名前	フィールドまたはリンクのSQL名	コメント
フィールド名	タイトル	Crystal Reports Serverデータベースのレポートの名前 **注意:** **このフィールドは必須ではありませんが便利です。**

フィールドまたはリンクの名前	フィールドまたはリンクのSQL名	コメント
ファイル	FileName	Crystal Reports Serverでのレポートの識別子

Seagate Crystal Reportsのレポートを変更する

Asset Managerデータベースに格納されているレポート

1 レポートを表示します（ナビゲーションバーの［**管理／レポート機能／レポート**］リンク）。
2 レポートの詳細を表示します。
3 レポートをエクスポートして（［**エクスポート**］ボタン）、「**.rpt**」ファイルを作成します。
4 出力した「**.rpt**」レポートをCrystal Reportsで変更して保存します。
5 もう一度レポートの詳細画面を表示します。
6 変更した「**.rpt**」ファイルをインポートします（［**インポート**］ボタン）。
7 変更を保存します（［**変更**］ボタン）。

Crystal Reports Serverデータベースに格納されているレポート

Crystal Reports Serverデータベース内でレポートの識別子を変更するのではなく直接Crystal Reportsでレポートを変更します（▶ Crystal Reports Serverデータベースへのレポートの格納 [献 93]）。

詳細レポートを作成する

詳細レポートは、リストで選択したレコードの詳細情報を印刷するレポートです。

使用例

1 資産のリストを表示します。
2 資産を選択します。
3 ［**ファイル／印刷**］メニューを選択します。
4 ［**タイプ**］フィールドを［詳細レポート（Crystal Reports）］に設定します。
5 レポートを選択します。

6 印刷します。

これで、選択したレコードごとの詳細レポートが作成されます。

Crystal Reportsでレポートを設定する

詳細レポートを作成するには、次の手順に従います（Crystal Reports Professional 5.0を使用した場合）。

1 ［**Insert/ Formula Field**］メニューを使って、計算式フィールドを作成します。次のシンタックスに従ってフィールド名を入力します。

```
<レポートに使うテーブルのSQL名>Id
```

注意:

テーブルのSQL名の大文字と小文字を区別して入力してください。

例えば、［資産］テーブルのレポートを作成する場合は、次の計算式を使います。

```
amAssetId
```

注意:

計算式のフィールド名のシンタックスと主キーのSQL名を混同しないでください。例えば、［資産］テーブルの主キーは［lAstId］で、［amAssetId］ではありません。

計算式［CurrentUserId］（大文字と小文字を区別）を使うと、レポートを印刷しているユーザを識別することができます。印刷時に、この計算式がAsset Managerデータベースに接続しているユーザのID番号（現在のログインのSQL名：lEmplDeptIdの値）を取得します。

指定したテーブルの特定のレコードのレポート結果を表示する場合は、計算式フィールドを編集し、Asset Managerデータベース内にあるそのレコードのテーブルの主キーを入力します。

例：

```
512
```

注意:

新しい計算式フィールドの名前を確定すると表示されるウィンドウでフィールドを編集します。計算式フィールドが既に存在する場合は、［Edit］ボタンをクリックして編集します。

2 ［**Report/ Edit Selection Formula/ Record**］メニューを使って、選択計算式を編集します。この式には、次のシンタックスを使います。

```
{<対象テーブルのSQL名>.<主キーとして使うフィールドのSQL名>} = @<計算式フィールド名>}
```

ここではテーブルとフィールドのSQL名の大文字と小文字の区別は重要ではありません。例：

```
{amAsset.lAstId} = {@amAssetId}
```

このようにして設定したレポートをAsset Managerのデータベースにインポートすると、自動的に対象レポートとして指定されます。これを確認するには、次の手順に従います。

1 ［**ツール／レポート機能／レポート**］メニューを使ってレポートの一覧を表示します。
2 新しいレポートを作成します。
3 ［**インポート**］ボタンをクリックして、前述の手順で設定したCrystal Reportsファイル（拡張子は「**.rpt**」）をインポートします。
4 このファイルが追加されると、［**テーブル**］フィールド（SQL名：TableName）にそのテーブルのSQL名が表示されます。表示されない場合は、計算式フィールドとCrystal Report内の選択計算式をチェックしてください。

レポートを印刷する

レポートの印刷方法は、レポートタイプ（詳細、一覧、グラフ）によって異なります。

詳細レポート（Windowsクライアントのみ）

警告:

詳細レポートはWindowsクライアントで利用できますが、Webクライアントでは利用できません。

詳細レポートの印刷には、次のような方法があります。

ナビゲーションバーでのレポートの選択

1 レポートの対象であるレコードを含むリストを表示します。
2 レポートの対象であるレコードを選択します。

重要項目:

1つのレコードのみを選択できます。

3 レポートの機能ドメインに対応するナビゲーションバーを展開します。

> ヒント:
>
> **レポートの機能ドメインは、レポートの詳細の［ドメイン］（Domain）フィールドで定義します。**

4 ナビゲーションバーでレポート名をクリックします。
5 ［**印刷**］アイコンをクリックします。

レポートの一覧から

1 レポートを表示します（ナビゲーションバーの［**管理／レポート機能／レポート**］リンク）。
2 レポートの詳細を表示します。
3 ［**プレビュー**］をクリックします。
4 レポートの対象であるレコードを選択します。
5 ［**OK**］をクリックします。
6 ［**印刷**］アイコンをクリックします。

［ファイル／印刷］メニューから

1 レポートの対象であるレコードのリストを表示します。
2 レポートの対象であるレコードを選択します。
3 ［**ファイル／印刷**］メニューを選択します。
4 ［**タイプ**］フィールドに値［**詳細レポート(Crystal Reports)**］を入力します。
5 ［**レポート**］リストでレポートを選択します。
6 ［**印刷**］をクリックします。

リストレポートとグラフ

リストレポートまたはグラフは、さまざまな方法で印刷できます。

ナビゲーションバーでのレポートの選択

1 レポートの機能ドメインに対応するナビゲーションバーを展開します。

> ヒント:
>
> **レポートの機能ドメインは、レポートの詳細の［ドメイン］（Domain）フィールドで定義します。**

2 ナビゲーションバーでレポート名をクリックします。

3 ［**印刷**］アイコンをクリックします。

レポートの一覧から（Windowsクライアントのみ）

1 レポートを表示します（ナビゲーションバーの［**管理／レポート機能／レポート**］リンク）。
2 レポートの詳細を表示します。
3 ［**プレビュー**］をクリックします。
4 ［**OK**］をクリックします。
5 ［**印刷**］アイコンをクリックします。

［ファイル／印刷］メニューから（Windowsクライアントのみ）

1 ［**ファイル／印刷**］メニューを選択します。
2 ［**タイプ**］フィールドに「**規定レポート（Crystal Reports）**」を入力します。
3 ［**レポート**］リストでレポートを選択します。
4 ［**印刷**］をクリックします。

モジュールに関連するCrystalレポートの識別

あるモジュールに関連するCrystalレポートを識別するには、

1 Asset Manager Windowsクライアントを起動します。
2 レポートを表示します（ナビゲーションバーの［**管理／レポート機能／レポート**］リンク）。
3 リスト内で右クリックします。
4 ショートカットメニューから［**ユーティリティ／リストの設定**］を選択します。
5 **ドメイン**リンク（Domain）をリスト内の列に追加します。
6 ［**OK**］をクリックします。
7 ［**ドメイン**］列でリストを並べ替えます。
8 モジュールのレポートはドメインの名前で識別できます。

   例：**/ポートフォリオ管理/IT/レポート/**

画面上のボタン（Windowsクライアント）または画面のアクション（Webクライアント）へのレポートの関連付け

画面上のボタン（Windowsクライアント）や画面のアクション（Webクライアント）にレポートを関連付けるにはいくつかの方法があります。

詳細については以下を参照してください。

- **『カスタマイズ』**ガイドの**「クライアントワークステーションのカスタマイズ」**の章**「ボタンのカスタマイズ」**のセクション。
- **『カスタマイズ』**ガイドの**「データベースのカスタマイズ」**の章**「既存のオブジェクトのカスタマイズ／オブジェクトのカスタマイズ／詳細のカスタマイズ／ボタン」**のセクション。
- **『カスタマイズ』**ガイドの**「データベースのカスタマイズ」**の章**「新規オブジェクトの作成／アクションボタンの作成」**のセクション。

5 コントロールパネル

Asset Managerを使用すると、レポートと統計を含むコントロールパネルを作成できます。コントロールパネルは、意思決定を容易に行うために経営管理上の情報を提供します。ちょうど自動車や飛行機の計器盤のように、関連データや重要データを表示することによって問題を予測し企業経営を円滑化することを支援します。コントロールパネルの便利な点は、表示するインジケータの選択に直接つながっていることです。

コントロールパネルの作成

コントロールパネルは**コントロールパネル**タイプのアクションで定義します。

コントロールパネルを作成するには：

1. ナビゲーションバーの［**管理／アクション**］リンクを選択し、アクションの詳細画面で［**新規作成**］をクリックします。
2. コントロールパネルの名前を選択し、アクションの［**タイプ**］として［**コントロールパネル**］を選択します。
3. ナビゲーションバーのツリー構造にコントロールパネルを表示する場合は、コントロールパネルの［**ドメイン**］を定義します。

警告:

コントロールパネルに［ドメイン］を指定しないと、Webクライアント上でコントロールパネルを表示できません。

また、すべてのアクションに共通するその他のプロパティも指定できます。

注意:

詳細については、このガイドの「アクション [献 127]」の章を参照してください。

4 ［**インジケータ**］タブで、コントロールパネルに追加するインジケータを選択します。

インジケータ関連情報は、［**コントロールパネル項目**］（amDashboardItem）テーブルに格納されます。

レポートと統計という2種類のオブジェクトをAsset Managerのコントロールパネルのインジケータとして使用します。ここでは、統計またはレポートの作成についての説明を省略します。Crystalレポートの作成については、このガイドの「Crystal Reports [献 69]」の章を、統計の作成については、このガイドの「統計 [献 105]」の章を参照してください。

インジケータを関連付けるには：

1 コントロールパネルの［**インジケータ**］タブを選択して、⊞ボタンをクリックします。
2 コントロールパネル内部のインジケータを識別するために［**名前**］を選択します。
3 インジケータの［**タイプ**］を定義します。
4 最後に、[タイプ]に応じて**レポート**または**統計**を1つ選択します。

注意:

⊟をクリックして、コントロールパネルからインジケータを削除します。コントロールパネルとのインジケータ関連が削除される旨のメッセージが表示されます。［コントロールパネル項目］（amDashboardItem）テーブルに格納されているオブジェクトは削除されます。

5 また、［**デフォルト更新**］チェックボックスをオンにすることもできます。このチェックボックスをオンにすると、コントロールパネルのデータが［**統計／統計更新の頻度**］で定義された間隔に従って更新されます。

注意:

アプリケーションのオプションは［編集／オプション］メニューによって使用できます。

このチェックボックスがオフになっていると、コントロールパネルが静止状態になり、データが動的に更新されることがありません。

コントロールパネルの例

次の例では、会社の人事部が使用するコントロールパネルの作成について説明します。次に示すインジケータがコントロールパネルに適しているものとして定義されています。

- 社内の男性／女性の内訳
- 部署および場所ごとの従業員の内訳
- 社内の勤務年数別従業員の内訳

これらのインジケータは、『統計 [献 105]』章の「スクリプトではない統計 [献 111]」のセクションで作成しました。この例で必要となる統計を作成するには、この節を参照してください。

手順1：コントロールパネルを作成する

1 ［**ツール／アクション／編集**］メニューを選択して、［**新規作成**］をクリックし、新しいアクションを作成します。
2 新しいレコードのフィールドに、次に示すデータを入力します。

フィールド	値
名前	人事部コントロールパネル
タイプ	コントロールパネル

3 ［**作成**］をクリックして作成を確定します。

手順2：インジケータを関連付ける

新規に作成されたコントロールパネルの詳細で［**インジケータ**］タブを選択します。次の手順に従うと、各種の統計用インジケータを連続して追加できます。

1 をクリックします。
2 インジケータの［**名前**］フィールドに入力します。追加する統計の名前を使用できます。
3 インジケータの［**タイプ**］として［**統計**］を選択します。
4 ［**統計**］フィールドの横にある選択リストをクリックしてから、追加する統計を選択します。
5 ［**追加**］（複数のインジケータを追加する場合は、2つの矢印がある［**追加**］ボタン）をクリックします。

次に［**変更**］をクリックします。

手順3：コントロールパネルを開いてデータ内のナビゲーションをする

コントロールパネルを起動するには、［**ツール／アクション**］メニューを使用するか、［**ドメイン**］（Domain）フィールド内で定義したWindowsクライアントまたはWebクライアントのナビゲーションバーで、コントロールパネルの名前をクリックします。

コントロールパネルの統計は、状況依存メニューのオプションを使用して表示されたときにカスタマイズすることができます。また、データゾーンをクリックすると統計サンプルの詳細を表示できます。例えば、男性の従業員数を表す3Dセクタのゾーンをクリックすると、部署と従業員の画面が開き、［**性別**］フィールドが［**男性**］のデータだけがフィルタリングされて表示されます。

注意:

レポートでは、ナビゲーションは使用できません。

6 統計

データベース情報をグラフで表示するために統計が使用されます。

統計は、それだけでは表示することができず、コントロールパネルまたはビジネスホームページ経由で表示する必要があります。▶統計を見やすくする [献 110]

この章で説明されている**統計**と、『自動更新によるレポートの表示（Windowsクライアントのみ） [献 88]』章で説明されている**Crystal Reports統計**とは異なります。

- Crystal Reports統計は、自動的に更新可能なCrystal Reportsレポートを表示するのに使用されます。
- ［**統計**］グラフを表示する場合、アイテムによって表されるレコードを表示するためにグラフのアイテムを拡大できます。［**Crystal Reports統計**］では拡大できません。

統計の作成

統計を作成するには：

1 統計を表示します（ナビゲーションバーの［**管理／統計...**］リンク）。

2 新規統計を作成し（［**新規**］ボタン）、さまざまなフィールドに入力します。

3 データベースに統計の結果を格納する場合、［**結果の格納**］（bStored）オプションを有効にしてください。

▶パフォーマンスの最適化 [献 108].

4 次の2つのいずれかの方法で統計を定義できます。

- 標準**クエリ**を使用する。
  a ［**クリプト**］(bScript)チェックボックスをオフのままにします。
  b ［**パラメータ**］タブで、さまざまなパラメータを入力して統計を作成します。
    - 必須コンポーネントを以下に挙げます。
      - ［**テーブル**］（TableName）フィールドを分析するためのレコードがあるテーブルを指定します。
      - 要素のグループ化に使用するフィールドやリンクを指定するには、［**グループ**］（GroupBy）フィールドを使用します。
    - オプションのコンポーネントを以下に挙げます。
      - テーブルの一部のレコードのみを使用して統計を作成する場合、［**フィルタ**］フィールドで統計に含めるレコードを返すフィルタを指定します。

        このフィールドを空欄にすると、すべてのレコードが考慮されます。
      - 2番目の値でグループ化する場合、［**グループ**］フィールドに入力してから、［**サブグループ**］（SubGroupBy）フィールドを使用することで、2番目の値でグループ化できます。
      - 式を使用してグラフの値を生成する場合、［**式を使用**］（bUseAgregExpr）ボックスをチェックしてください。

        次に、［**式**］（AqlAgregExpr）フィールドに値を入力します。

        これにより、操作（平均など）を実行したり、ロットを計上できるようになります。

- 統計を表示するために複雑な**スクリプト**を記述する。
  1 ［**スクリプト**］（bScript）チェックボックスをオンにします。
  2 ［**スクリプト**］タブで、統計データを計算するために使用するスクリプトを入力します。

    ▸スクリプトとクエリの違いについては、「クエリの代用としてのスクリプト [献 107]」の段落を参照してください。

    ▸スクリプト統計の例については、「スクリプト統計 [献 115]」の段落を参照してください。

警告:

統計に［ドメイン］を指定していない場合、統計をWebクライアント経由で使用することができません（統計をコントロールパネルまたはビジネスホームページ▸ 統計を見やすくする [献 110])に関連付けていても、使用できません）。

警告:

Webクライアント：選択した表現のタイプ（seType field）が［曲線］の場合、統計が表示できるのは表示された値が［**numbers**（番号）］の場合のみです。［グループ］（GroupBy）および［**Sub-group by**（サブグループ）］（SubGroupBy）フィールドは数値をポイントする必要があります。

5 フィールドに入力したら、［**作成**］をクリックします。
6 グラフを表示するには、［**テスト**］をクリックします。
7 ［**ESC**］をクリックして、グラフを閉じます。

クエリの代用としてのスクリプト

「統計の作成 [献 105]」の段落で説明するように、統計を説明するには次の2つの方法があります。

- 単純なクエリを使用する。
- 複雑なスクリプトを使用する。

クエリではなくスクリプトを使用する理由

標準クエリが、テーブルデータを表示するのに使用されます：

- 1つまたは2つのグループ化基準、
- データベースフィールドおよび特定の計算フィールド、
- 1つのフィルタ、
- 必要であれば、計算方法（式）が適用されます。

複雑な統計を定義する場合は、**スクリプト**を使って定義する必要があります。

例えば、期間別にレコードをグループ化したり、さまざまなテーブルのレコードを比較したり、［**完全名**］タイプフィールドのデータを使用してグループ化する場合、**スクリプト**を記述する必要があります。

クエリのシンタックス

▶ 以下を参照してください。

- スクリプトを記述する際に使用するシンタックスについては、「スクリプト統計 [献 115]」のセクションの例。
- グラフ統計用のカスタムカラーの定義 [献 122]

パフォーマンスの最適化

パフォーマンスの問題を引き起こしている原因

統計はコントロールパネルを介して表示されます。

コントロールパネルを表示すると、Asset Manager は1つまたは複数の統計を介して適切なテーブルにアクセスします。

これにより以下の問題が発生する場合があります。

- クライアントステーションの処理能力が低下する。
- 他のユーザに対して特定のテーブルへのアクセス速度が遅くなる。
- リアルタイムで計算されている場合に結果の表示が遅くなる。また、指定したコントロールパネルに対する計算に複数の統計が必要になる場合がある。

これらのパフォーマンスの問題に対処するソリューション

パフォーマンス問題を減少させるため、統計を自動的にかつ定期的に生成して、結果を［**Statistics memos（統計メモ）**］（amStatMemo）テーブルに格納できます。

この場合、コントロールパネルを表示すると、Asset Managerはこのテーブルをチェックし、データをリアルタイムで収集する実際のテーブルはチェックしません。これにより、計算時間が大幅に削減されます。

このオプションの利点を以下に挙げます。

- クエリを送信する他のユーザのテーブルアクセスが遅くならない。
- クエリはユーザのステーションではなく、Asset Manager Automated Process Managerが起動します。

このソリューションの実装方法

統計の詳細で、［**結果の格納**］（bStored）オプションを有効にします。

自動的に定期的に結果を格納するように、Asset Manager Automated Process Managerを設定します。

1. Asset Manager Automated Process Managerを開始します。
2. 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. モジュール構成ページを開きます（［**ツール／Configure modules（モジュールの構成...）**］メニュー）。
4. ［**テーブルの統計情報の更新**］モジュール（Stats）を選択します。
5. ［**有効**］チェックボックスをオンにします。
6. 更新を実行する間隔を定義します。

7 変更を保存します（［**変更**］ボタン）。

Asset Manager Automated Process Managerは［**結果の格納**］オプションをオンにしているすべての統計に対してスケジュールした間隔で統計を起動します。結果は、［**Statistics memos（統計メモ）**］（amStatMemo）テーブルに格納されます。

統計詳細の拡大

グラフが表示されているとき、枠をクリックすることで各枠の詳細にアクセスできます。これにより、フィルタ基準でのグループ化を使用してフィルタされた統計の表が開きます。

個別に表示するグループの最大数の定義

Windowsクライアント

グラフを見やすくするには、20を超える値グループを使用しないことをお勧めします。

残りのグループは、［**その他の値**］グループにまとめることができます。

デフォルトでグラフを見やすくするために、個別に表すグループ数を20未満にしてください。20未満に設定すると、先頭の20の値グループが表示され、それ以外は［**その他の値**］と呼ばれるグループにまとめられます。

グラフに表示される値のグループは値でソートされます（最後に配置され、表示するグループの最大数を超過したすべてのグループが含まれる**その他の値**グループを除く）。

表示する最大グループ数を変更するには：

1 オプショングループを開きます（［**編集／オプション**］メニュー）。
2 ［**統計／処理する値の最大数**］ブランチを展開します。
3 個別に表示するグループ数を定義します。

統計を見やすくする

コントロールパネルから

統計を表示する前に、以下の操作を行って統計をコントロールパネルに関連付けることができます。

1 アクションを表示します（ナビゲーションバーの［**管理／アクション**］リンク）。

2 新規アクションを作成します（［**新規**］ボタン）。

3 ダッシュボードを使用できるナビゲーションバーノードを指定するためには、［**コントロールパネル**］値と［**ドメイン**］（Domain）フィールドを選択して、［**タイプ**］（seActionType）フィールドを入力します。

> **警告:**
>
> **［ドメイン］フィールドに値を入力しないと、Webクライアント上にコントロールパネルを表示できません。**

4 作成を確定します（［**作成**］ボタン）。

5 ［**インジケータ**］タブで、以下の手順に従い統計インジケータを作成します。

   a ⊞をクリックします。

   b インジケータの［**名前**］（Name）フィールドを入力します。挿入する統計の名前を使用します。

   c ［**タイプ**］（seType）タイプフィールドで、［**統計**］を選択します。

   d 次に、［**統計**］リンクを入力します。

   追加する統計がなくなるまでこれらの手順を実行します。

6 コントロールパネルを更新します（［**変更**］ボタン）。

   コントロールパネルが作成され、使用できるようになりました。

コントロールパネルでこれらの統計をしようするには、「コントロールパネル [献 101]」の章を参照してください。

ビジネスホームページから

統計を使用するには、次の方法で統計をビジネスホームページに関連付けてください。

1 ビジネスホームページを作成します（ナビゲーションバーの［**管理／ビジネスホームページ**］リンク）。

2 ［**統計**］タブで、▣アイコン（Windowsクライアントの場合）または［**追加**］ボタン（Webクライアント）をクリックしてから統計を選択することで、目的の統計を挿入します。

▶ビジネスホームページについては、『**Tailoring**』ガイドの「**Customizing Web clients**」セクションの「**Business home pages**」の章を参照してください。

統計の表示

統計は、それだけでは表示することができず、コントロールパネル経由で表示する必要があります。

コントロールパネルで統計を使用する方法については、このガイドの「コントロールパネル [献 101]」の章を参照してください。

統計の例

専門分野データには、コントロールパネルからデータベースに作成された統計の例が用意されています。

これらのコントロールパネルは、さまざまな専門分野データで利用できます。

統計はデフォルトでデモデータベースで使用できます。

ただし、統計を本番データベースで使用するには、統計を含む専門分野データをインポートする必要があります。

▶ 専門分野データのインポートの例は、『**調達**』ガイドの「**概要**」の章、「**準備段階**」のセクション、「**専門分野データの既存データベースへのインポート**」の段落にあります。

スクリプトではない統計

例1：コストセンタおよびCPU周波数別コンピュータの平均価格に関する統計

コストセンタおよびCPU周波数別にコンピュータの平均価格の統計を表示します。この統計の情報を以下に挙げます。

- 2つのグループ：コストセンタとPC周波数の表現
- コンピュータに使用するフィルタ
- 式を使用して平均価格を計算します

この統計を作成するには：

1 統計を表示します（ナビゲーションバーの［**管理／統計...**］リンク）。

2 新規統計を作成します（［**新規**］ボタン）。
3 次の表に示すフィールドとリンクを入力します。

フィールドまたはリンク（ラベル）	フィールドまたはリンク（SQL名）	値
名前	名前	コストセンタとCPU周波数別コンピュータの平均価格
［**全般**］タブ		
表現のタイプ	seType	横棒
凡例	bLegend	チェックするボックス
［**クエリ**］タブ		
テーブル	TableName	コンピュータ（amComputer）
グループ	GroupBy	Portfolio.CostCenter.Title
フィールドの値を、リストから選ぶのではなく入力した場合、入力を確定するかカーソルを別のフィールドに移動すると、次のフィールドが表示されます。		
サブグループ	SubGroupBy	lCPUSpeedMHz
式の使用	bUseAgregExpr	チェックするボックス
式	AqlAgregExpr	Avg(Portfolio.mAvgPrice)

4 統計を作成します（［**作成**］ボタン）。
5 グラフを表示するには、［**テスト...**］をクリックします。
6 ［**ESC**］をクリックして、グラフを閉じます。
7 すべてのウィンドウを閉じます。

例2：男女の内訳

会社で働く社員を男女に分けて表示するものとします。グループ化は、1つの基準（従業員の役職）のみを使用して実行する必要があります。同じテーブルで定義されているサービスを除外するために使用するフィルタを作成して、統計にその情報が使用されないようにする必要があります。

最初にこのフィルタを作成します：

1 クエリを表示します（ナビゲーションバーの［**管理／クエリ**］リンク）。
2 以下の表で指定されるフィールドとリンクを入力して新規フィルタを作成します。

フィールドまたはリンク（ラベル）	フィールドまたはリンク（SQL名）	値
名前	名前	従業員のみ
テーブル	TableName	部署と従業員（amEmplDept）
［**クエリ**］タブ		
クエリ	memQueryText	bDepartment = 0

3 ［**作成**］をクリックします。

4 すべてのウィンドウを閉じます。

ここで、統計を作成します。

1 統計を表示します（ナビゲーションバーの［**管理／統計...**］リンク）。

2 新規統計を作成します（［**新規**］ボタン）。

3 次の表に示すフィールドとリンクを入力します。

フィールドまたはリンク（ラベル）	フィールドまたはリンク（SQL名）	値
名前	名前	男女の内訳
［**全般**］タブ		
凡例	bLegend	チェックするボックス
表現のタイプ	seType	セクタ
3D表示	b3D	チェックするボックス
［**クエリ**］タブ		
テーブル	TableName	部署と従業員（amEmplDept）
フィルタ	フィルタ	従業員のみ
グループ	GroupBy	MrMrs

4 ［**作成**］をクリックします。

5 グラフを表示するには、［**テスト...**］をクリックします。

6 ［**ESC**］をクリックして、グラフを閉じます。

7 すべてのウィンドウを閉じます。

例3：部署とサイト別従業員の内訳

この例では、部署とサイト別の従業員の内訳を表示します。情報は2つの基準（部署とサイト）を使用してグループ化され、従業員のみにフィルタが適用されます（以前と同様）。

1 統計を表示します（ナビゲーションバーの［**管理／統計...**］リンク）。

2 新規統計を作成します（［**新規**］ボタン）。

3 次の表に示すフィールドとリンクを入力します。

フィールドまたはリンク（ラベル）	フィールドまたはリンク（SQL名）	値
名前	名前	社内の場所別従業員の内訳
［**全般**］タブ		
表現のタイプ	seType	横棒
凡例	bLegend	チェックするボックス
［**クエリ**］タブ		
テーブル	TableName	部署と従業員（amEmplDept）
フィルタ	フィルタ	従業員のみ

フィールドまたはリンク（ラベル）	フィールドまたはリンク（SQL名）	値
グループ	GroupBy	Parent.Name
フィールドの値を、リストから選ぶのではなく入力した場合、入力を確定するかカーソルを別のフィールドに移動すると、次のフィールドが表示されます。		
サブグループ	SubGroupBy	Location.FullName
［**全般**］タブに戻ります		
重ねて表示	bStacked	チェックするボックス

4 ［**作成**］をクリックします。
5 グラフを表示するには、［**テスト...**］をクリックします。
6 ［**ESC**］をクリックして、グラフを閉じます。
7 すべてのウィンドウを閉じます。

例4：時間別社内従業員の内訳

この例では、年功別でデータを分類します。以前と同様、従業員は必要に応じて年功でフィルタされグループ化されます。年功はデータベースには格納されていません。しかし、［**部署と従業員**］テーブルには従業員が会社に就職した日時（［**雇用日**］）を示すフィールドがあります。この値を使用して、従業員の年功を計算できます。これには、情報をグループ化するのに使用する値を持つ特殊フィールドを使用します。

特殊フィールドを作成するには：

1 特殊フィールドを表示します（ナビゲーションバーの［**管理／システム／特殊フィールド**］リンク）。
2 以下の情報を持つ新規レコードを作成します。

フィールドまたはリンク	値
名前	会社での従業員の在籍期間
テーブル	部署と従業員（amEmplDept）
フィールドタイプ	AQL
結果のデータ型	数値
AQLシンタックス	Round((DaysDiff(GetDate(), dHire) / 365), 0) AQLクエリは、以下の日時間の差を計算する**DaysDiff()**関数を使用します。 ■ 計算時刻：**GetDate()**関数 ■ 雇用日：［**雇用日**］（dHire）フィールド 期間は日数で返されます。結果は1年の日数である365で除算されます。その後、結果は**Round()**関数でもっとも近い整数値に丸められます。

3 作成を確定します（［**作成**］ボタン）。
4 すべてのウィンドウを閉じます。

ここで、統計を作成します。

1 統計を表示します（ナビゲーションバーの［**管理／統計...**］リンク）。
2 新規統計を作成します（［**新規**］ボタン）。
3 次の表に示すフィールドとリンクを入力します。

フィールドまたはリンク（ラベル）	フィールドまたはリンク（SQL名）	値
名前	名前	社内の勤務年数別従業員の内訳
［**全般**］タブ		
凡例	bLegend	チェックするボックス
表現のタイプ	seType	曲線
［**クエリ**］タブ		
テーブル	TableName	部署と従業員（amEmplDept）
フィルタ	フィルタ	従業員のみ
グループ	GroupBy	以前に作成した［**年功**］特殊フィールドを選択します

4 ［**作成**］をクリックします。
5 ［**テスト...**］をクリックして、グラフを表示します。
6 ［**ESC**］をクリックして、グラフウィンドウを閉じます。
7 すべてのウィンドウを閉じます。

スクリプト統計

自作するためにテストや変更できるスクリプト統計の例を以下に挙げます。

デモデータベースには、他のスクリプト統計の例があります（ナビゲーションバーの［**管理／統計...**］リンクを選択します）。

例1（基本例）

この例の目的は、単純な統計（データベースからデータは取得されません）の作成方法と、情報レンダリングの管理方法を示すことです。

この例では、3つのグループ（**France**、**Germany**、および**England**）が作成されます。各グループには特定人数の従業員（**10**、**20**、および**18**）がいて、国別の従業員数を表示するグラフを作成します。

次のクエリを生成する必要があります。

```
<group Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%'" value="10"/>
<group Name="Germany" filtertable="amEmplDept" filter="Location.FullName like '/Germany/%'" value="20"/>
<group Name="England" filtertable="amEmplDept" filter="Location.FullName like '/England/%'" value="18"/>
```

このクエリを生成するのに、［**スクリプト**］オプション（このチェックボックスをチェック）が使用されます。このオプションでは、統計データを生成するスクリプトを記述できます。事前定義されたXML実体を変換するのに**XmlAttribute** APIが使用されます（▶ 詳細については、『**Programmer's Reference**』を参照してください）。

1 統計を表示します（ナビゲーションバーの［**管理／統計...**］リンク）。
2 新規統計を作成します（［**新規**］ボタン）。
3 次の表に示すフィールドとリンクを入力します。

フィールドまたはリンク（ラベル）	フィールドまたはリンク（SQL名）	値
名前	名前	国別従業員数
スクリプト	bScript	チェックするボックス
［**全般**］タブ		
表現のタイプ	seType	縦棒
凡例	bLegend	チェックするボックス
［**スクリプト**］タブ		
データスクリプト	スクリプト	RetVal = "<group " & XmlAttribute("Name", "France") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/France/%'") & " " & XmlAttribute("value", "10") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<group " & XmlAttribute("Name", "Germany") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/Germany/%'") & " " & XmlAttribute("value", "20") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<group " & XmlAttribute("Name", "England") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/England/%'") & " " & XmlAttribute("value", "18") & " />" & Chr(13) & Chr(10)

4 ［**作成**］をクリックします。
5 ［**テスト...**］をクリックして、グラフを表示します。
6 ［**ESC**］をクリックして、グラフウィンドウを閉じます。
7 すべてのウィンドウを閉じます。

例2（サブクグループ）

サブグループは前の例に追加され、各従業員の性別（［**Mr**］、［**Mrs**］、または［**Miss**］）に対応します。

性別と国別に従業員数を表すグラフを表示します。

次のクエリを生成する必要があります。

```
<group Name="Mr.">
<subgroup Name="France" filtertable="amEmplDept" filter="Location.FullN
ame like '/France/%' and MrMrs='Mr'" value="5"/>
<subgroup Name="Germany" filtertable="amEmplDept" filter="Location.Full
Name like '/Germany/%' and MrMrs='Mr'" value="9"/>
<subgroup Name="England" filtertable="amEmplDept" filter="Location.Full
Name like '/England/%' and MrMrs='Mr'" value="10"/>
</group>

<group Name="Mrs.">
<subgroup Name="France" filtertable="amEmplDept" filter="Location.FullN
ame like '/France/%' and MrMrs='Mrs'" value="3"/>
<subgroup Name="Germany" filtertable="amEmplDept" filter="Location.Full
Name like '/Germany/%' and MrMrs='Mrs'" value="7"/>
<subgroup Name="England" filtertable="amEmplDept" filter="Location.Full
Name like '/England/%' and MrMrs='Mrs'" value="7"/>
</group>

<group Name="Miss">
<subgroup Name="France" filtertable="amEmplDept" filter="Location.FullN
ame like '/France/%' and MrMrs='Miss'" value="2"/>
<subgroup Name="Germany" filtertable="amEmplDept" filter="Location.Full
Name like '/Germany/%' and MrMrs='Miss'" value="4"/>
<subgroup Name="England" filtertable="amEmplDept" filter="Location.Full
Name like '/England/%' and MrMrs='Miss'" value="1"/>
</group>
```

以前と同様、この情報をコード化するのに［**スクリプト**］オプションを使用します。

1 統計を表示します（ナビゲーションバーの［**管理／統計...**］リンク）。
2 新規統計を作成します（［**新規**］ボタン）。
3 次の表に示すフィールドとリンクを入力します。

フィールドまたはリンク（ラベル）	フィールドまたはリンク（SQL名）	値
名前	名前	性別と国別での従業員数
スクリプト	bScript	チェックするボックス
［**全般**］タブ		
凡例	bLegend	チェックするボックス
表現のタイプ	seType	縦棒
［**スクリプト**］タブ		

フィールドまたはリンク（ラベル）	フィールドまたはリンク（SQL名）	値
データスクリプト	スクリプト	RetVal = "<group " & XmlAttribute("Name", "Mr.") & " >" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "France") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/France/%' and MrMrs like '/Mr/%' ") & " " & XmlAttribute ("value", "5") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "Germany") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/Germany/%' and MrMrs like '/Mr/%' ") & " " & XmlAttribute("value", "9") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "England") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/England/%' and MrMrs like '/Mr/%' ") & " " & XmlAttribute("value", "10") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "</group>"
		RetVal = RetVal & "<group " & XmlAttribute("Name", "Mrs.") & " >" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "France") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/France/%' and MrMrs like '/Mrs/%' ") & " " & XmlAttribute("value", "3") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "Germany") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/Germany/%' and MrMrs like '/Mrs/%' ") & " " & XmlAttribute("value", "7") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "England") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/England/%' and MrMrs like '/Mrs/%' ") & " " & XmlAttribute("value", "7") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "</group>"

フィールドまたはリンク（ラベル）	フィールドまたはリンク（SQL名）	値
		RetVal = RetVal & "<group " & XmlAttribute("Name", "Miss") & " >" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "France") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/France/%' and MrMrs like '/Miss/%' ") & " " & XmlAttribute("value", "2") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "Germany") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/Germany/%' and MrMrs like '/Miss/%' ") & " " & XmlAttribute("value", "4") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "England") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/England/%' and MrMrs like '/Miss/%' ") & " " & XmlAttribute("value", "1") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "</group>"

4 ［**作成**］をクリックします。
5 ［**テスト...**］をクリックして、グラフを表示します。
6 ［**ESC**］をクリックして、グラフウィンドウを閉じます。
7 すべてのウィンドウを閉じます。

例3（詳しいユーザ）

さらに複雑なこの例では、Asset Managerデータベースからデータがエクスポートされます。

これにより、研修モデル別にソートされた、データベースにあるすべての研修セッションの概要を表示します。

- 研修日数
- 研修プログラムの費用
- 研修セッション数。研修セッション参加者につき1研修セッション

これらのタイプのスクリプト統計は、必ず同じスキームに従って記述されます。この例では、自分独自の統計に合わせたテストおよび変更が可能なスクリプト統計のコメント付きモデルを示します。

以前と同様、新規スクリプト統計を作成します（［**スクリプト**］ボックスをチェック）。

手順1：必要な変数を定義する

まず、統計の作成で使用する変数を定義する必要があります。

統計の構造をデザインする必要があります。

1 データの抽出および表示に使用している主テーブルが選択されます。この例では、選択されたテーブルは［**研修**］（amTraining）テーブルです。
2 いずれか1つのデザインオプションを選択します。
   - 複数の式を使用したグループ化。
   - 1つの式で複数の基準を使用したグループ化。

   この例では、**1つの基準**（研修モデル）と**3つの式**（研修日数、研修セッションの費用、研修セッションの数）を使用してデータをグループ化します。

   列は、主グループで使用される研修モデルに対応しています。

   シリーズは各式（研修セッションの数、研修日数、研修セッションの費用）に対応しています。

［**統計**］タブの［**データスクリプト**］フィールドに次のテキストを入力します。

```
SetMaxInst(1000000000)
const CONST_NumberSecondsPerMonth = 2592000
const CONST_NumberSecondsPerDay = 86400

Dim strRC as String
strRC = Chr(13) & Chr(10)

RetVal = ""

Dim strTable, strColumnField, strSerieField as String

strTable = "amTraining"

Dim strColumns, strSeries as String
strColumns = AmDbGetStringEx("SELECT DISTINCT Model.Name FROM a
mTraining WHERE Model.Name <> "" ,"?","|")
strSeries = "トレーニングセッション数" & "|" & "トレーニング日数" & "|" & "
Cost"
```

注意:

以下を定義できます。
- 指定したテーブルから特定のレコードを選択するための［主フィルタ］。
- ［主式］は、レコードに対して操作を実行するために使用します。

この場合、異なるシリーズの値を取得した後（手順2：値を取得する [献 121]の段落）、値の取得に使用したフィルタに主フィルタを追加し、各サブグループ式に主式を追加する必要があります。

これらの関数は必要ないので例では使用しません。これらの関数の使用例を調べるにはデータベースを参照してください。

注意:

統計を記述する際に、この例で説明されているように式の代わりに日付を操作する場合は、［**StatLib**］スクリプトライブラリを使用してください（このライブラリには、ナビゲーションバーの［管理／スクリプト］リンク経由でアクセスできます）。

列（strColumns）を定義するために使用可能な、事前定義された関数には次の3つがあります。**CURRENTFISCALYEAR**、**LAST12MONTHS**、**NEXT12MONTHS**。

これらの関数の使用方法については、スクリプトライブラリ（ナビゲーションバーの［管理／スクリプト］リンク）のこれらの関数の説明を参照してください。

手順2：値を取得する

この手順では、定義された各グループとシリーズに対して値を取得します。

- **strColumns**変数の研修モデルグループの値を取得するには、以下を入力します。

```
Dim iColumn, iColumnCount as Long
iColumnCount = CountValues(strColumns,"|")

For iColumn = 1 to iColumnCount
Dim strColumn as String
strColumn = GetListItem(strColumns, "|", iColumn)
Dim lColumnValue as Long
```

- **strSeries**で定義された3つのシリーズ（研修セッションの数、研修日数、研修セッションの費用）について、データの計算および値の取得に使用するフィルタ（**strFilter**）と式（**strQuery**）を定義する必要があります。

```
Dim iSerie, iSerieCount as Long
iSerieCount = CountValues(strSeries  ,"|")

dim strFilter as String
dim strQuery as String
dim lSerieValue as Long

for iSerie = 1 to iSerieCount
Dim strSerie as String
strSerie  = GetListItem(strSeries, "|", iSerie)

if "Number of training sessions" = strSerie then
strFilter = " Model.Name = " & AmSQLTextConst(strColumn)
strQuery = "SELECT Count(lTrainingId) FROM amTraining WHERE " &
```

```
strFilter

elseif "Number of training days" = strSerie then
strFilter = " Model.Name = " & AmSQLTextConst(strColumn)
strQuery = "SELECT Sum(tsDuration)/" & CONST_NumberSecondsPerD
ay & "  FROM amTraining WHERE " & strFilter

elseif "Cost" = strSerie then
strFilter = " Model.Name = " & AmSQLTextConst(strColumn)
strQuery = "SELECT Sum(mCost+mTax) FROM amTraining WHERE " &
strFilter

end if

lSerieValue = AmDbGetLong(strQuery)
```

手順3：グループとサブグループを定義する

列データはグループによって定義され、シリーズデータはサブグループによって定義されます。

```
dim subgroup as String
subgroup = subgroup &  "<subgroup " & XmlAttribute("name", strSerie) & " "
& XmlAttribute("filtertable", strTable) & " " & XmlAttribute("filter", strFilter
) & " " & XmlAttribute("value", lSerieValue) & "/> "
lColumnValue = lColumnValue + lSerieValue
next

RetVal = RetVal & "<group " & XmlAttribute("name", strColumn) & " " & Xm
lAttribute("value", lColumnValue) & "> "
RetVal = RetVal & subgroup
RetVal = RetVal & "</group> "
next
```

手順4：統計を表示する

［**テスト...**］をクリックして、統計を表示します。

グラフ統計用のカスタムカラーの定義

一連のサブグループまたはシリーズに対してグラフ統計結果を返す、スクリプトクエリが使用するカスタムカラーのセットを指定できます（複数のサブグループに関わるクエリでのみ色の定義が意味を持ちます）。

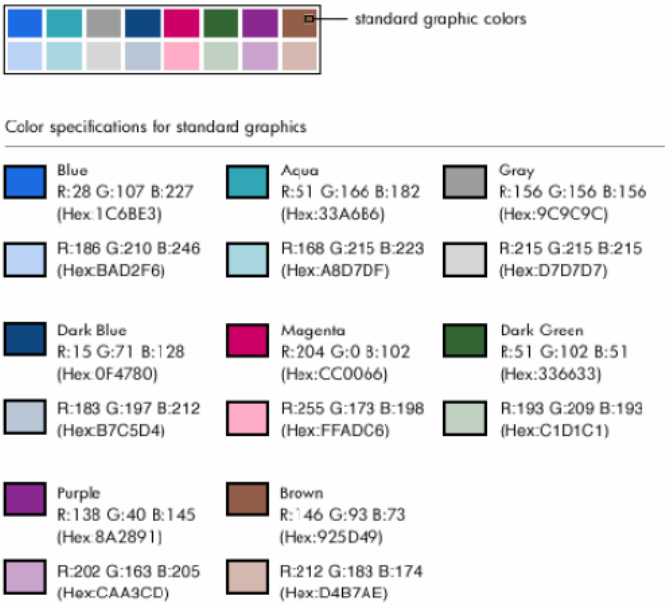
シリーズの色を定義するには、サブグループ定義ステートメントで**color=**パラメータを使用します。例：

```
<subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like '/Germany/%' and MrMrs='Mr'" value="9" color="#cc0066"/>
```

注意:

Asset ManagerのWindowsクライアントとWebクライアントとの互換性を確保するには、色の指定は、RGB値に相当する16進数値（例：紫色の場合、#6f006f）を使用する必要があります。

標準のグラフィック色は、次の値で表されます（ただし、RGB値に相当する16進数値を使用して、他の任意の色を指定できます）。



エラー処理

色の定義での異常を解決するには、次の規則が使用されます。

- クエリ内の1つ以上のサブグループに対して**color=**パラメータが未定義の場合、これらのサブグループにはデフォルトの色が適用されます。
- **color=**パラメータで認識できない値が使用されている場合、デフォルトの色が適用されます。

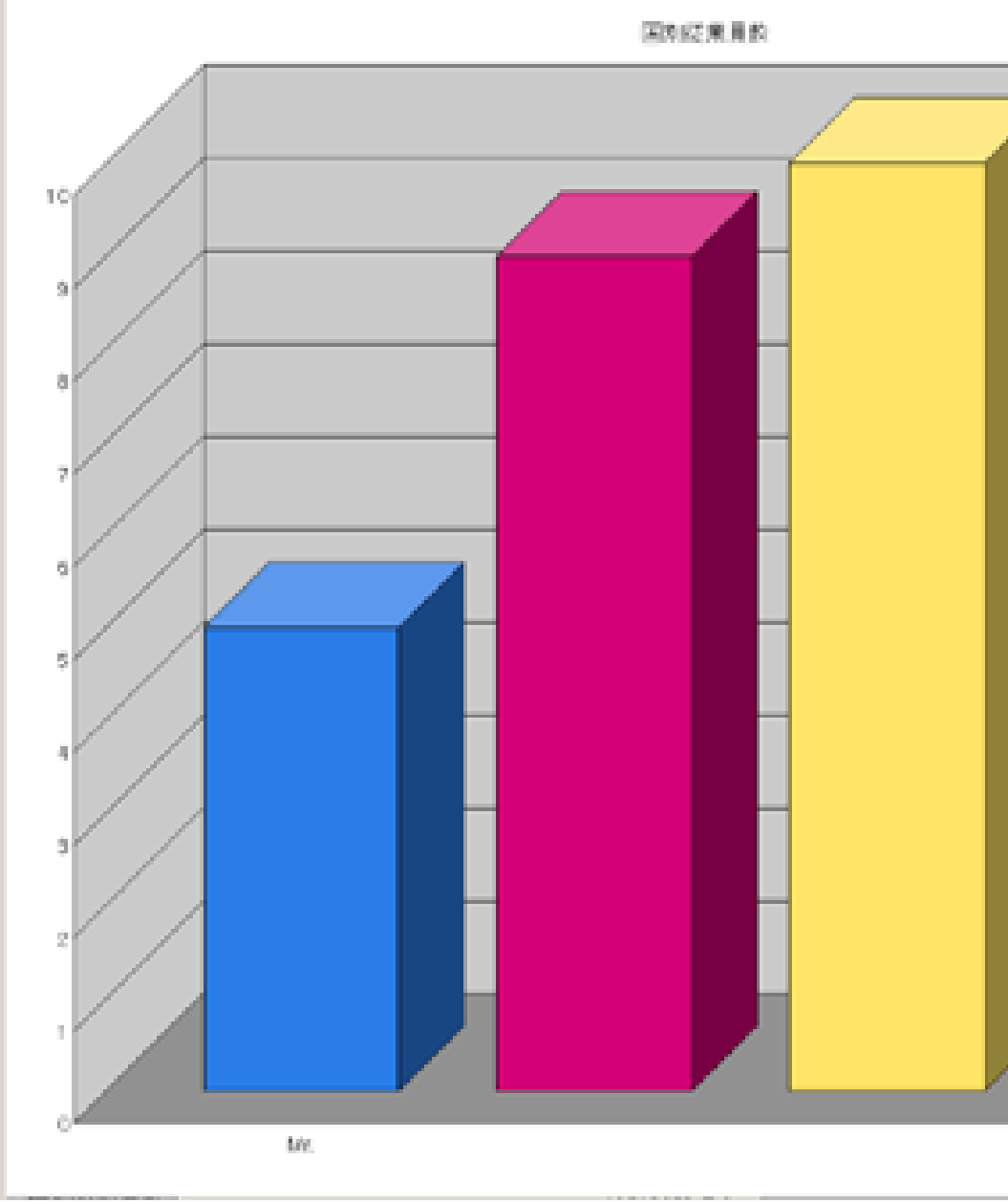
- 色はサブグループごとに1回だけ定義できます（色の再定義の試行は無視されます）。

実用例

以下に定義するクエリでは、**Mr**グループの3つのサブグループに対してカスタムカラーの青色、紫色、黄色を定義します。

```
<group Name="Mr.">
<subgroup Name="France" filtertable="amEmplDept" filter="Location.FullN
ame like '/France/%' and MrMrs='Mr'" value="5" color="#1c6be3" />
<subgroup Name="Germany" filtertable="amEmplDept" filter="Location.Full
Name like '/Germany/%' and MrMrs='Mr'" value="9" color="#cc0066"/>
<subgroup Name="England" filtertable="amEmplDept" filter="Location.Full
Name like '/England/%' and MrMrs='Mr'" value="10" color="#ffde53"/>
</group>
```

このクエリの結果は次のグラフになります。



7 アクション

この章では、Asset Managerでアクションを定義する方法について説明します。

アクションを作成するには、［**ツール／アクション／編集**］メニューを選択します。

アクションは、［ツール／アクション］メニューまたはツールバーのアクションリストから実行できます。

開発のベストプラクティス

テスト機能ドメインを使用してアクションをカスタマイズする

推奨事項

可視にする前に、新規アクションやアクションの新規バージョン（特にウィザードなど）をそのテスト機能ドメインに関連付けることを推奨します。

テスト機能ドメインをマスクして、開発中のアクションをユーザから隠しておくこともできます。

理由

アクションが正しくなければ、そのアクションのリンク先である機能ドメインがAsset Manager WebのWebサービスとして生成されません。

その結果、機能ドメインにリンクされるすべてのオブジェクトは（有効なものも含め）、Asset Manager Webにより公開されるWebサービスから消えます。

実装

アクションをテスト機能ドメインに関連付けるには：

1. Windowsクライアントまたは Webクライアントを起動します。
2. 機能ドメインを表示します（ナビゲーションバーの［**管理／機能ドメイン**］メニュー）。
3. テスト機能ドメインを作成します。

   この機能ドメインにWebサービスが確実に生成されないようにするには、［**Webサービス**］（seWebService）フィールドに値「**なし**」を入力します。

   テスト機能ドメインのSQL名を書き留めます。
4. アクションを表示します（ナビゲーションバーの［**管理／アクション**］リンク）。
5. アクションをテスト機能ドメインにリンクします（［**ドメイン**］（Domain）リンク）。
6. オブジェクトがテストされ有効になると、そのオブジェクトを公開される適切な機能ドメインにリンクします。

Webサービスのタグ付け

スクリプトまたはウィザードタイプのアクションを変更する場合、Asset Manager Webを導入していた場合はWebサービスを再度タグ付けするようにしてください。

▶ **『カスタマイズ』**ガイドの**「データベースのカスタマイズ」**の章、**「開発のベストプラクティス／Webサービスのタグ付け」**のセクション。

アクションの定義

アクションを使うとAsset Managerデータベース上で実行されるタスクを完全に、あるいは部分的に自動化することが出来ます。

次の種類のアクションがあります。

- 実行可能ファイル
- DDE
- メッセージ
- スクリプト：Asset Managerデータベースのオブジェクトを変更します。
- ウィザード
- 印刷

- ソフトウェア配布
- アクション
- コントロールパネル

アクションをリストから選択して実行できるようにするには、まず最初にアクションを定義しなければなりません。

注意:

機能ドメインにより、アクションのドメインを定義し、機能ごとに分類することができます。

機能ドメイン

Asset Managerでは、ソフトウェアの機能を分類するドメインを定義できます。数種の機能ドメインがデフォルトで定義されています。これらのデフォルトの機能ドメインは、［ファイル／モジュールの起動］メニューで起動または終了できるモジュールに対応しています。

機能ドメインは、WindowsクライアントおよびWebクライアントのナビゲーションバーに表示される情報を作成し分類するために使用されます。1つのアクションに対して機能ドメインを選択すると、そのアクションは、WindowsクライアントおよびWebクライアントのナビゲーションバーの機能ドメインの名前の下に表示されるようになります。

注意:

ナビゲーションバーの内容は状況に応じて変わります。アクションが状況依存タイプである場合（例えば、特定の画面が開いていないとアクションが実行されない、など）、現在のコンテキストがその定義にあっている場合のみ（例えばその特定の画面が開かれている場合）のみ、アクションがナビゲーションバーに表示されます。

機能ドメインを定義するには：

1. ［**管理／機能ドメイン**］メニューを選択します。
2. ［**新規作成**］をクリックします。
3. ［**名前**］フィールドに機能ドメインの名前を入力します。この名前はWindowsおよびWebクライアントナビゲーションバーに表示されます。Asset Managerは、機能ドメインに**SQL名**をデフォルトで割り当てます。このデフォルト値は変更可能です。
4. 場合によっては機能ドメインの［**親ドメイン**］を選択します。
5. ［**作成**］をクリックして作成を確定します。

警告:

あるユーザに対して機能ドメインへのアクセスが完全に（読取りも書込みも）禁止されている場合、そのユーザはそのドメインのアクション、ビュー、レポートにアクセスできません。実際には、機能ドメインを定義する際には以下のテーブルへの読取りアクセスを許可することをお勧めします。

- amFuncDomain
- amViewDef
- amReport
- amForm
- amAction
- amScriptLibrary

アクションの作成

ここでは、アクションを作成する方法について説明します。

- アクションのタイプ
- 一般的な作成方法
- ［DDE］タブページに入力する
- ［メッセージ］タブページに入力する

アクションのタイプ

Asset Managerでは、いくつかのタイプのアクションを定義できます。

- 実行可能ファイル
- DDE
- メッセージ
- スクリプト
- ウィザード
- ソフトウェア配布
- 印刷
- アクション
- コントロールパネル

［実行可能ファイル］アクション

［実行可能ファイル］アクションは、プログラムを実行するアクションです。

このアクションによって、**.exe**、**.com**、**.bat**、**.pif**のアプリケーションを起動できます。その他のドキュメントの場合も、拡張子がファイルマネージャでアプリケーションに関連付けられていれば起動することができます。

［DDE］アクション

［DDE］アクションは、DDEリクエストを処理できるDDEサーバアプリケーション（またはDDE準拠のアプリケーション）にDDEリクエストを送ります。

DDEは、Dynamic Data Exchange（動的データ交換）の略で、プログラム間で動的に情報を交換する手法です。Asset ManagerではDDEを使って別のアプリケーションでコマンドを実行できます。

例：DDEを使って、特定のファイルをMicrosoft Wordで開くように指定できます。

［メッセージ］アクション

［メッセージ］アクションを使うと、以下のメッセージシステムを経由してメッセージを送信できます。

- Asset Managerの内部メッセージシステム
- 外部のVIM標準メッセージシステム（Lotus Notes、Lotus cc:Mailなど）
- 外部のMAPI標準メッセージシステム（Microsoft Exchange、Microsoft Outlookなど）
- 外部のSMTP標準メッセージシステム

警告:

ユーザが接続できるメッセージシステムでのみメッセージを送信できます。

VIM、MAPIまたはSMTP標準のメッセージを送信するには、Asset Managerで次の情報を指定する必要があります。

- Asset Managerデータベースを開いた従業員の詳細画面（［部署と従業員］テーブル）の、**［メッセージ］**タブ／**［アカウント］**フィールド（SQL名：MailLogin）および**［パスワード］**フィールド（SQL名：MailPassword）で、メッセージの送信者を指定します。
- 従業員の詳細画面の**［全般］**タブページ／**［電子メール］**フィールド（SQL名：EMail）で、メッセージを受け取るためのアドレスを指定します。

Asset Managerの内部メッセージシステム経由でメッセージを送信するには、送信者と受信者の両方の詳細画面の**［プロファイル］**タブページで、**［ユーザ名］**フィールドと**［パスワード］**フィールドを指定します。

注意:

Asset Managerユーザの内部メッセージ用アドレスは、［ログイン］と同じです。

警告:

管理者は、名前が「Admin」のユーザを作成してから［アカウント］、［電子メール］、［パスワード］の各フィールドに入力し、外部メッセージシステムを使ってAsset Manager Automated Process Managerが正しく機能することを確認してください。

［スクリプト］アクション

［スクリプト］アクションを使うと、Asset Managerデータベースのあらゆる操作を実行できます。上級ユーザの操作範囲が広がり、他のタイプのアクションでは実行できない操作を行うことができます。特に次の操作を実行できます。

- レコードの作成
- レコードの削除
- レコードの複製
- Asset Managerデータベース内の1つまたは複数のオブジェクト（テーブル内のすべてのレコード、フィールド、リンクなど）の変更

このタイプのアクションで実行する操作は、Basicスクリプトで記述します。そのため、Asset Manager APIで使うような複雑な関数を使用できます。

注意:

［スクリプト］アクションで使用できる複雑な関数は、データベースの構造自体も変更できるため、この種のアクションを使用するとデータベースの整合性を損なう危険性があります。このため、スクリプトアクションの使用は適切な知識を持つ上級ユーザのみに限られます。

アクションのコンテキストに応じてデータベースオブジェクトの値を変更するには、様々な関数を使います。

- アクションにコンテキストがない場合は、**AmSetFieldStrValue()**または**AmSetFieldLongValue()**など、Asset Manager APIから由来の関数を使う必要があります。
- アクションがテーブルをコンテキストとしてもつ場合は、**Set()**関数を使うことができます。シンタックスは次のとおりです。

```
Set [<Link.Link.Field>]=<値>
```

［ウィザード］アクション

ウィザードは複雑なアクションです。ウィザードについて詳しくは、このガイドの**ウィザード**の章を参照してください。

ウィザードは、繰り返し行われる複雑な作業を、Asset Managerで表示されるガイドに従って段階的に実行できるアクションです。ウィザードは、専用のプログラミング言語を使って設計します。

［印刷］アクション

［印刷］アクションを使うと、レポートを印刷できます。

このタイプのアクションでは次のフィールドに値を入力します。

- ［タイプ］（seFormType）フィールド：印刷するレポートのタイプを指定します。
- ［レポート］フィールド：どのレポートを使用するのかを指定します。

注意:

アクションのコンテキストは、レポートに定義されているコンテキストにより決定されます。

[ソフトウェア配布]アクション

［ソフトウェア配布］アクションでは、一連のコンピュータにスケジュールされたタスクを作成できます。

このタイプのアクションは、［コンピュータ］テーブル（SQL名：amComputer）に依存し、コンピュータが1台選択されている場合にのみ実行できます。

コンテキストを変更することはできません。

このタイプのアクションの場合に入力するフィールドは、［配布］タブにあります。

ソフトウェア配布の詳細については、▶『**Software distribution**』ガイドを参照してください。

［ソフトウェア配布］タイプのアクションを実行するには：

1. コンピュータを表示します（ナビゲーションバーの［**ポートフォリオ管理／IT／IT機器／コンピュータ**］リンク）。
2. ターゲットコンピュータグループ（この場合は、インタフェースに表示されないものも含めて、グループ内のすべてのコンピュータが選択される）とスケジュールされたタスクを作成するターゲットコンピュータを選択します（一方または両方も可）。
3. アクションを実行します（［**ツール／アクション／<アクション名>**］メニュー）。

スケジュールされたタスクが1つ作成されます。このタスクのターゲットは、［**ソフトウェア配布識別子**］（SWDID）が入力され、アクションを実行する前に選択した各コンピュータです。

［アクション］タイプのアクション

［アクション］タイプのアクションでは、クエリによって選択された一連のレコードに対してアクションを実行できます。

このタイプのアクションの場合に入力するフィールドは、［アクション］タブにあります。

応用例：

- ［**ソフトウェア配布**］タイプのアクションを実行するには：

  このアクションは、一連のコンピュータにスケジュールされたタスクを作成します。

- ［**アクション**］タイプのアクションを作成し、［**アクション**］タイプのアクションの選択クエリで選択したコンピュータに対して、［**ソフトウェア配布**］タイプのアクションを実行します。

  選択クエリを使用して、ターゲットコンピュータが自動的に選択されるようにします。

［コントロールパネル］タイプのアクション

［**コントロールパネル**］タイプのアクションを使用すると、統計とレポートを含むコントロールパネルを作成できます。このアクションにはコンテキストはありません。

［**インジケータ**］タブには、コントロールパネルによって表示されるオブジェクトのリストが含まれます。このタブにオブジェクトを追加したり、このタブからオブジェクトを削除したりすることができます。

アクションの作成

アクションを作成するには：

1. ［**ツール／アクション／編集**］メニューを選択します。
2. ［**新規作成**］をクリックします。
3. アクションの名前を入力します。
4. ［**タイプ**］（SQL名：seActionType)フィールドに、作成するアクションのタイプを指定します。選択したアクションのタイプによって、次のタブページが表示されます。
   - ［**実行可能ファイルまたはDDE**］
   - ［**メッセージ**］
   - ［**スクリプト**］
   - ［**ウィザード**］
   - ［**印刷**］
   - ［**ソフトウェア配布**］
   - ［**アクション**］
   - ［**インジケータ**］
5. アクションの詳細画面で［**SQL名**］フィールド（SQL名：SQLName）に入力することが出来ますが、必須ではありません。SQL名からアクションを特定できます。特にDDEコマンドを使ってアクションを実行する場合（Asset ManagerをDDEコマンドサーバとして使う場合）に使います。

注意:

［SQL名］フィールドに入力しないと、アプリケーションによって自動的にSQL名が入力されます。

6 ［**コンテキスト**］フィールド（SQL名：ContextTable）に次のように入力します。
   - ドロップダウンリストからテーブルを選択した場合は、アクションはそのテーブルに依存します。そのテーブルのレコードのリストまたはレコードの詳細画面を表示した場合にのみアクションが実行されます。
   - アクションにテーブルを指定しない場合は、ドロップダウンリストの一番上の［**（テーブルなし）**］オプションを選択します。
7 ［**ドメイン**］フィールドでアクションが属する機能ドメインを指定できます。アクションは、WindowsクライアントおよびWebクライアントのナビゲーションバーのこのドメインの下に表示されます。
8 アクションにアイコンを付けることができます。必須ではありません。

   アクションにアイコンを付けるには、アイコンをアクションの詳細画面の左上に表示される四角い領域に挿入します。ここで指定したアイコンの画像は、ツールバーのアクションリストに表示されます。リストの中でアクティブなアイコン（デフォルトで画面に表示されるもの）は、一番最後ににツールバーから起動したアクションのアイコンです。
9 ［**説明**］タブページのフィールドと、作成するアクションのタイプに固有のタブページのフィールドに値を入力します。
10 ［**作成**］をクリックします。

注意:

Asset Managerの管理者はすべてのアクションに目を通し、そのアクションが共有されているかどうか、また管理者が作成したアクションかどうかを確認する必要があります。

［DDE］タブページに入力する

[DDD]アクションに固有の情報はアクションの詳細画面のDDEタブページにあります。

このタブページは、アクションの［**タイプ**］フィールド（SQL名：seActionType）を［DDE］に設定した場合にのみ表示されます。

DDEのメカニズムはソフトウェアによって提供される「サービス」に基づいています。DDEのメカニズムを実行するためには、「コマンド」が実行されるべきコンテキストを表す「トピック」を定義しなければなりません。

次のように指定します。

- ［**サービス**］（SQL名：DDEService）フィールドに、呼び出す実行可能ファイルのDDEサービスの名前を指定します。通常は、実行可能プログラム固有のサービス名があります。サービスの一覧は、実行可能ファイルのドキュメントを参照してください。
- ［**トピック**］フィールド（SQL名：DDETopic）に、アクションの実行対象となるコンテキストを指定します
- ［**コマンド**］フィールド（SQL名：DDECommand）に、外部のアプリケーションに実行させたいコマンドを指定します。

  Wordの場合は、WordBasicまたはVisual Basicコマンドになります。

  呼び出すアプリケーションのDDEサービスで複数のコマンドを使用できる場合は、複数のコマンドを並べて記述できます。

  この場合は、外部アプリケーションのシンタックスに従う必要があります。
- サービスが存在しない場合は、［**ファイル**］フィールド（SQL名：ActionFile）に、サービスをアクティブにするアプリケーションを起動するファイルを指定します。これが、DDEコマンドに応答するメインアプリケーションになります。
- ［**フォルダ**］フィールドで、サービスを起動するファイルのパスを指定します。
- ［**パラメータ**］フィールドで、DDEサービスを実行しているプログラムに渡すパラメータを指定します。

重要

外部アプリケーションに送信するコマンドは、大括弧（[", "]）で囲みます。Microsoft Wordを使う例は次のとおりです。

```
[FileOpen("c:\tmp\test.txt")]
```

- アクションがコンテキストをもつ場合は、変数を使ってデータベースのフィールド値を参照することができます。この場合は、変数も大括弧で囲むので、Asset Managerがコマンドと変数を区別できなくなります。区別できるようにするには、コマンドの大括弧の前にバックスラッシュ（\）を付けます。前述の例は、コンテキストをもつアクションの場合には、次のように記述します。

  ```
  \[FileOpen("c:\tmp\test.txt")\]
  ```

  以下のようにコマンドと変数を組み合わせることができます。この例ではコンテキストは[資産]テーブルです。

  ```
  \[FileOpen("c:\tmp\"+"[AssetTag]"+".txt")\]\[FileClose()\]\[FileExit()\]
  ```

- アクションのコンテキストを指定しない場合は、バックスラッシュを使う必要はありません。大括弧で囲んだテキストは、外部アプリケーションに送信するコマンドと見なされます。

［メッセージ］タブページに入力する

［メッセージ］アクションに関する情報は、アクションの詳細画面の［メッセージ］タブページにあります。

このタブページは、アクションの［**タイプ**］（SQL名：seActionType）フィールドを［**メッセージ**］に設定した場合にのみ表示されます。

警告:

システムを正しく機能させるためには、システムのPATH変数にVIM DLL（VIM32.DLL）とMAPI DLL（MAPI32.DLL）を含んだフォルダを指定する必要があります。

［参照オブジェクト］（SQL名：RefObject）フィールドの使用目的

このフィールドは、［**コンテキスト**］フィールドで選択したテーブルのリンクを選択するのに使います。

このフィールドは、Asset Managerの内部メッセージシステムで送信するメッセージ以外には使えません。ここでリンクを指定しておくと、メッセージの詳細画面で［**参照オブジェクト**］ボタンをクリックしただけで、メッセージの発信をトリガしたオブジェクトに直接アクセスできます。参照先のオブジェクトがアクションをトリガするレコードそのものの場合は、［**参照オブジェクト**］フィールド（SQL名：RefObject）に入力する必要はありません。

配信済通知を受信する方法

メッセージが相手先に配信されたことを通知する配信済通知を、送信者が通常のメッセージサービス経由で受信するようにするには、［**配信済通知**］（SQL名：bAcknowledgment）チェックボックスをオンにします。

この通知は、Asset Managerデータベースを開いた従業員（［部署と従業員］テーブル）の［**全般**］タブページ／［**電子メール**］（SQL名：EMail）フィールドに指定したアドレスに送信されます。

注意:

Asset Managerの内部メッセージシステムや、MAPIまたはSMTPメッセージシステムで送信したメッセージの場合は、配信済通知を受信できません。

アドレスを指定する

アドレスを指定する方法は以下のとおりです。

<メッセージエンジン>:<メッセージアドレス>形式のアドレス

<メッセージエンジン>には次の標準を指定できます。

- AM：Asset Managerの内部メッセージシステムを使用します。

- MAPI：MAPI標準のメッセージシステム（Internet Mail、Microsoft Outlookなど）を使用します。
- VIM：VIM標準のメッセージシステム（Lotus Notesなど）を使用します。
- SMTP：SMTP標準のメッセージシステム（インターネット標準）を使用します。

<メッセージアドレス>は、各メッセージシステムの通常の書式で指定します。内部メッセージのアドレスは、ログイン名と同じです。

アドレスの例を次に示します。

- AM:Admin
- MAPI:CathyBernard@taltek.com
- VIM:Cathy Bernard / TALTEK
- SMTP:cbernard@taltek.com

<Asset Managerログイン>形式のアドレス

従業員の［**ログイン**］（従業員の詳細画面の［**プロファイル**］タブページ）がアドレス内で指定されている場合、この従業員の詳細画面／［**全般**］タブページ／［**電子メール**］（SQL名：EMail）フィールドに指定されているメッセージシステムが使用されます。

［**電子メール**］フィールドに値が入力されていないと、内部メッセージシステム経由でメッセージが送信されます。

例：

1. 次の3名のAsset Managerログインにメッセージを送信するとします：Cathy、Gerald、Philip。
2. ［**電子メール**］フィールドは、それぞれ次のように指定されています。Cathyの場合は「MAPI:CathyBernard@taltek.com」、Geraldは「VIM:Gerald Colombo / Taltek」、Philipの［**電子メール**］フィールドは空です。
3. 送信者がMAPIアカウントを持っている場合、CathyにはMAPIで、他の2人にはAsset Managerの内部メッセージシステムでメッセージが送信されます。
4. 送信者がVIMアカウントを有する場合、GeraldにはVIMで、他の2人にはAsset Managerの内部メッセージシステムでメッセージが送信されます。

変数を使ったアドレス

アクションの対象がテーブルに依存する場合は、大括弧（[]）の中に変数を指定します。これらの変数は、Asset Managerデータベース内のフィールド値を参照します。

例：［資産］テーブルで選択した資産のユーザにメッセージを送信する場合は、**[User.Email]**を使うことができます。

アクションの例

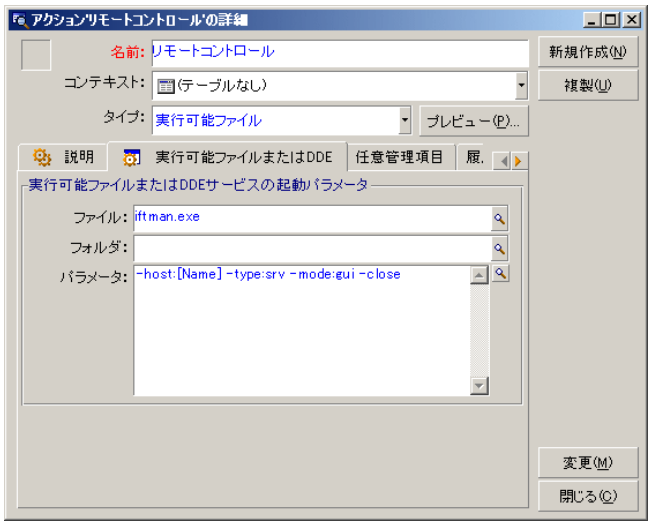
ここでは、Asset Managerによるアクションの実行例を紹介します。

- ［実行可能ファイル］タイプのアクションの例
- ［DDE］タイプのアクションの例
- ［メッセージ］タイプのアクションの例
- ［スクリプト］タイプのアクションの例

［実行可能ファイル］タイプのアクションの例

次の画面はコンテキストを使わないアクションを指定しています。このアクションは、Asset Manager Automated Process Managerを起動して「**amdemo**」データベースに接続します。

図 7.1. ［実行可能ファイル］タイプのアクション



［DDE］タイプのアクションの例

［**DDE**］タイプのアクションを使うと、次のように様々な作業を実行できます。

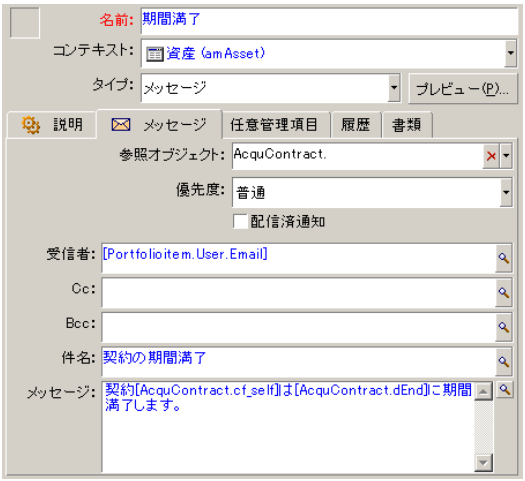
- Asset ManagerのデータをMicrosoft Excelのワークシートに挿入する。
- 発注に関連する情報を会計ソフトウェアに挿入する。
- チケットのクローズまたは作成を確認するメッセージを、ファックスで自動的に送信する。

- 作業指示の依頼をファックスで自動的に送信する。
- ...

［メッセージ］タイプのアクションの例

資産のリストから、ある資産のリース期間の満了日を知らせるメッセージをその資産のユーザに送信します。この資産は、取得方法が［リース］であり、リース明細にリンクされている必要があります（［**取得**］タブページ）。参照するオブジェクトを依頼明細ではなく購入依頼にするには、次のようにアクションの詳細を設定します。

図 7.2. 参照するオブジェクトを指定するメッセージタイプのアクションの設定例



［スクリプト］タイプのアクションの例

［**スクリプト**］タイプのアクションの作成では、Asset Managerデータベースを変更するBasicスクリプトを記述する必要があります。

注意:

スクリプトタイプのアクションで使用できる関数については、『Programmer's Reference』ガイドの「Index of functions by field of application」の章、「Built-in functions」のセクションを参照してください。

準備

アクションの作成を準備する手順は次のとおりです。

1 ［**ツール／アクション／編集**］メニューを選択し、アクションの詳細画面で［**新規作成**］ボタンをクリックします。

2 作成するアクションに「テスト」などの名前を割り当て、［**タイプ**］（SQL名：seActionType）フィールドを［**スクリプト**］に設定します。アクションのコンテキストは選択しないでください。［**作成**］をクリックします。

3 ［**スクリプト**］タブページでボタンをクリックし、スクリプトビルダのウィンドウを表示します。スクリプトタイプのアクションで使われる**Success()**というプログラマブル関数には、明示的な戻りコードは不要です。以下の例では、下の表に示す情報に基づいて属性のテーブル内に新しいレコードを作成します。

フィールドラベル	フィールドのSQL名	フィールドの値
名前	名前	PC
作成	seBasis	ポートフォリオ品目
接続可能	bIsCnxClient	このチェックボックスはオンにします。

スクリプトを記述する

次のように入力します。

```
Dim lrec As Long
Dim lres As Long
lrec=AmCreateRecord("amNature")
lres=AmSetFieldStrValue(lrec, "Name", "PC")
lres=AmSetFieldStrValue(lrec, "seBasis", 1)
lres=AmSetFieldStrValue(lrec, "bIsCnxClient", 1)
AmInsertRecord(lrec)
```

注意:

このアクションにより、ユーザの介入なしに希望の属性を作成できるようになります。

Set()関数の使用例

ここでは、コンテキストに［属性］テーブルを指定して、［**スクリプト**］タイプのアクションで前の例と同じ属性を作成します。次のようにスクリプトを記述します。

```
Set [Name]="PC"
Set [seBasis]=1
Set [bIsCnxClient]=1
```

注意:

このアクションを実行するには、ユーザが［属性］テーブルを開き、［新規作成］をクリックする必要があります。また、スクリプトを実行した後に、［作成］をクリックして作成を確定する必要があります。

ヒント

スクリプト内の1つのアクションを無効にする場合は、戻りコードの値を0以外（例えば12001）に設定します。この値はエラーコードと見なされます。次のコマンドはアクションを中断し、すべての変更を取り消します。

```
RetVal=12001
```

変数を使う

状況依存アクションの詳細画面の［**実行可能ファイル**］、［**DDE**］、および［**メッセージ**］タブページでは、データベース内のフィールド、任意管理項目または特殊フィールドの値を参照する変数を使うことができます。

変数の書式は［**Link.Link.Field**］です。

入力するフィールドの右側にあるをクリックすると、特殊文字列画面が表示され、より簡単に変数を入力できます。

大括弧（[]）の外にあるものはすべてテキストと見なされます。

例：**[Link.Link.Field].doc**は、**Link.Link**というリンクを介してメインテーブルにリンクしたテーブル内の**Field**の値を呼び出します。

警告:

アクションで変数を使う場合は、アクションの詳細画面の［コンテキスト］フィールドでAsset Managerのテーブルを指定し、アクションを実行する前にそのテーブルのレコードリストでレコードを選択しておく必要があります。

アクションをテストする

アクションを作成時にテストするには、アクションの詳細画面の右上隅に表示される［**テスト**］ボタンを使います。

プレビューウィンドウが表示され、アクションのプレビューに対するコンテキストを選択することができます。このウィンドウには、［**計算**］と［**実行**］という2つのボタンがあります。

［計算］ボタン

コンテキストを選択し、［**計算**］ボタンをクリックします。これで［**実行可能ファイル**］または［**DDE**］または［**メッセージ**］のいずれかのタブページのフィールドに値が入ります。［**コンテキスト**］フィールド（SQL名：ContextTable）で選択したレコードから、変数が正しく抽出されていることを確認します。

［実行］ボタン

このボタンをクリックすると、プレビュー画面から直ちにアクションを実行できます。

アクションを実行する

次のいずれかの方法でアクションを実行できます。

- ツールバーのドロップダウンリスト を使います。
  - ボタンは、使用しているコンピュータ上でアクションを実行するたびに、そのアクションに関連付けられたアイコンがもしあれば、そのアイコンで置き換えられます。アクションを既に実行している場合は、アイコンまたは置き換えられたアイコンをクリックすると、アクションのアイコンがリセットされます。
  - ボタンをクリックすると、使用可能なアクションのリストが表示されます。
  - このドロップダウンリストをツールバーに挿入するには、［ツール／ツールバーのカスタマイズ］メニューを使います。このメニューは「ツール」カテゴリにあります。
- ［ツール／アクション］メニューを使います。サブメニューから希望のアクションを選択します。
- アクションの詳細画面の右上隅にある［**テスト**］ボタンを使います。
  - 実行するアクションにコンテキストを指定する必要がある場合は、［コンテキスト］フィールド（SQL名：ContextTable）にアクションの参照テーブルのレコードを指定します。
  - ［**実行**］をクリックして、アクションを実行します。

- ショートカットメニュー（マウスの右ボタンをクリックして表示）を使います。開いているテーブルで使用できるアクションがある場合は、ショートカットメニューにアクション名が表示されます。

リストで複数のレコードを選択

リストで複数のレコードを選択し、これらに1つのアクションを適用できます。

例えば複数の資産を選択し、そのユーザに同じメッセージを送信することができます。

「ウィザード」タイプのアクション

ウィザードは連続するページで構成されています。ユーザは、各ページで情報を選択したり、データを入力したりします。

ウィザードでは、次のように簡単にページ間を移動できます。

- ページに情報を正しく入力したら、［**次へ**］ボタンをクリックして次のページに移動できます（移動先のページは遷移の設定によります）。最後のページまで進むと、このボタンは使用できなくなります。
- ［**戻る**］ボタンをクリックすると、いつでも前のページに戻り、情報を訂正することができます。
- ［**終了**］ボタンをクリックすると、ウィザードの最終アクションをいつでも実行できます。情報不足のためにウィザードが指定されたタスクを実行できない場合は、必要な情報を入力するためのページが表示されます。

［**キャンセル**］ボタンをクリックすると、ウィザード（および関連するアクション）の実行を完全に取り消すことができます。

実行可能タイプアクション

実行可能タイプアクションの動作は、アクション詳細の［**ファイル**］（ActionFile）フィールドの値によって異なります。

以下の表は、状況に応じてWindowsクライアントとWebクライアントが取る可能性のある動作をまとめたものです。

アクションの詳細の［ファイル］（ActionFile）フィールドが、**http**や**https**で始まる場合

Windowsクライアントでアクションがトリガされる場合	Webクライアントでアクションがトリガされる場合
クライアントのWindowsコンピュータ上でデフォルトのInternetブラウザを起動し、［**ファイル**］フィールド（ActionFile）に入力されているURLアドレスにアクセスします。	Webクライアントのワークスペースの［**ファイル**］（ActionFile）フィールドで指定されているURLに対応するページを表示します。

アクションの詳細の［ファイル］（ActionFile）フィールドが、**ftp**で始まる場合

Windowsクライアントでアクションがトリガされる場合	Webクライアントでアクションがトリガされる場合
クライアントのWindowsコンピュータ上でエクスプローラを起動し、［**ファイル**］フィールド（ActionFile）に入力されているftpサイトにアクセスします。	Webクライアントのワークスペースの［**ファイル**］（ActionFile）フィールドで指定されているURLに対応するftpサイトを表示します。

アクションの詳細の［ファイル］（ActionFile）フィールドが、**mailto**で始まる場合

WindowsクライアントやWebクライアントコンピュータ上のデフォルトメッセージングアプリケーションを起動し、［**ファイル**］（ActionFile）フィールドの **mailto**：以降で指定されているパラメータを使用して新規メッセージを作成します。

アクションの詳細の［ファイル］（ActionFile）フィールドが他の値で始まる場合

Windowsクライアントでアクションがトリガされる場合	Webクライアントでアクションがトリガされる場合
［**ファイル**］（ActionFile）フィールドで指定されているファイルの実行	デフォルトで、ファイルはAsset Manager Web Serviceステーションや**Web**クライアント上で実行されず、エラーが返されます。 Asset Manager Web Serviceステーション上でファイルが確実に実行されるようにするには： 1 データベースオプション（Windowsクライアントの［**管理／データベースオプション...**］メニュー）を表示します。 2 ［**アクション／Executing executable-type actions（実行可能タイプのアクションの実行）**］（ExecuteAction）オプションを選択します。 3 このオプションを［**サーバ**］に設定します。 **警告:** **この操作には、若干のリスクが伴います。**

画面上のボタンへのアクションの関連付け

画面上のボタンにアクションを関連付けるにはいくつかの方法があります。

詳細については以下を参照してください。

- 『**カスタマイズ**』ガイドの「**クライアントワークステーションのカスタマイズ**」の章「**ボタンのカスタマイズ**」のセクション。
- 『**カスタマイズ**』ガイドの「**データベースのカスタマイズ**」の章「**既存のオブジェクトのカスタマイズ／オブジェクトをカスタマイズする／詳細のカスタマイズ／アクションボタンの作成**」のセクション。
- 『**カスタマイズ**』ガイドの「**データベースのカスタマイズ**」の章「**新規オブジェクトの作成／アクションボタンの作成**」のセクション。

8 メッセージ

Asset Managerでは、次の2種類のメッセージを管理できます。

- Asset Managerから発行され、内部メッセージシステムを介してAsset Managerデータベースに送信されたメッセージ
- Asset Managerで作成され、外部メッセージシステムを介して送信されたメッセージ

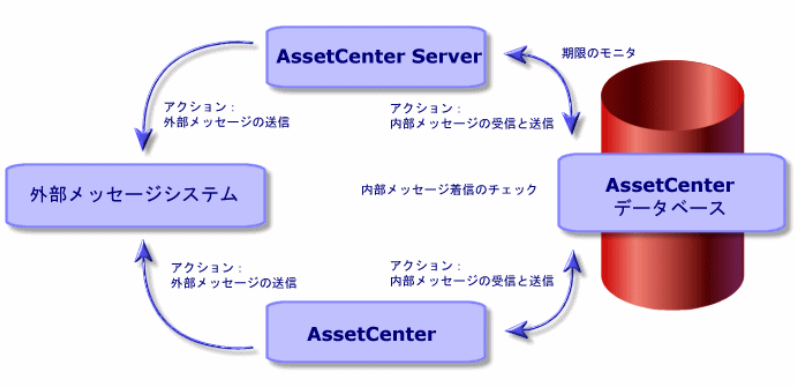
メッセージシステムの仕組み

Asset Managerは、以下のプロトコルを使用してメッセージの送信を管理します。

- AM (Asset Manager)
- SMTP
- MAPI
- VIM

受信時には、Asset ManagerはAM（Asset Manager）タイプのメッセージのみを管理します。

図 8.1. メッセージシステムの仕組み



メッセージシステムを設定する方法については、『**管理**』ガイドの「**メッセージシステム**」の章で詳細を説明しています。

メッセージを送信する方法

メッセージは、［**メッセージ**］タイプのアクションを介して作成されます。

メッセージを送信するには、まずメッセージを送信するアクションを作成する必要があります。

メッセージを送信するアクションは、以下のようにいくつかの異なる方法でトリガされます。

- ［**ツール／アクション**］メニューに表示されるリストからアクションを選択して、手動でトリガする。
- Asset Manager Automated Process Managerによって自動的にトリガされる。
- Asset Managerによって自動的にトリガされる。

［メッセージ］タイプのアクションの作成については、［メッセージ］タブページに入力する [献 137]のセクションを参照してください。

メッセージを参照する方法

内部メッセージシステムで受信したメッセージを参照する

エージェントは新しいメッセージが到着しているかどうかを確認し、ユーザに新着メッセージの受信を通知します。

受け取ったメッセージは次の方法で参照できます。

- ［**ツール／メッセージ**］メニューを使う。
- 新着メッセージを通知するメッセージボックスで参照する。

外部メッセージシステムで受信したメッセージを参照する

メッセージの受信者は、外部メッセージシステムのクライアントプログラム上で通常どおりメッセージを参照できます。

配信済通知

注意:

Asset Managerの内部メッセージシステムや、MAPIまたはSMTPメッセージシステムで送信したメッセージの場合は、配信済通知を受信できません。

配信済通知を受信するには、［**メッセージ**］タイプのアクションの詳細画面で［**配信済通知**］（SQL名：bAcknowledgment）チェックボックスをオンにします。

この通知は、Asset Managerデータベースを開いた従業員（［部署と従業員］テーブル）の［**全般**］タブページ／［**電子メール**］（SQL名：EMail）フィールドに指定したアドレスに送信されます。

9 ワークフロー

ワークフローの目的は、ビジネスプロセスとその実装方法を取りまとめることです。**WfMC**（Workflow Management Coalition：ワークフローによるソリューションの標準化を定義、開発、および促進する非営利団体）の定義を使用すると、プロセスとは「ビジネス目標または計画目標を共同で実現する一連のリンクされた手続きまたはアクティビティ」です。これらのプロセスを管理することは、作業手続きをモデル化して企業の経営にリンクされたすべてのアイテムを考慮することを意味します。

Asset Managerのワークフローシステムには次の特長があります。

- **ワークフロースキーム**を使用して、誰が何をいつどのように行うか、ビジネスプロセスを形式化することができます。
- 形式的定義に従ってこれらのプロセスの**インスタンス**を実行することを支援します。
- アクティビティ内で定義されプロセスに関連付けられている**タスク**を実行するために必要な情報とツールを使用できるようにします。タスクは自動的に実行される場合と、ユーザの処理が必要な場合があります。
- 実行済みまたは実行中のインスタンスを測定します。

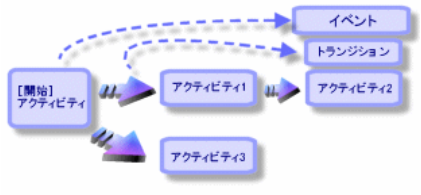
例えばワークフローを使うと、以下のようなプロセスのモデルを作成し、プロセスを自動化することができます。

- 購入依頼の承認手続き
- 資産の移動
- その他

Asset Managerでワークフロースキームを作成するには、次の要素を定義します。

- 開始アクティビティ (プロセスの開始点を構成する)
- イベント（アクティビティに含まれる）
- イベントによって生成される遷移
- 遷移によってトリガされるアクティビティ
- コンテキスト
- 制限時間とアラーム

図 9.1. ワークフロースキームの概略



定義

ここでは、ワークフローで使われるいくつかの重要な用語を定義します。

- ワークフローインスタンス
- ワークフローのアクティビティ
- ワークフローのイベント
- ワークフローの遷移
- ワークフローのタスク
- ワークフローのアクティビティの担当者
- ワークフローの実行グループ

ワークフローインスタンスの定義

「ワークフローインスタンス」とは、ワークフロースキームに定義された公式のプロセスに従って実行されているワークフローを指します。

ワークフローのアクティビティ

ワークフローのアクティビティは次の要素で構成されます。

- 実行するタスク。このタスクにはユーザによる操作が必要なものと、Asset Manager Serverで自動的に実行できるものがあります。
- 他のアクティビティへの遷移を起動するイベント

ワークフローのイベント

ワークフローのイベントは、アクティビティを実行した結果発生します。同様にワークフローのイベントにより、他のアクティビティを引き起こす遷移が起動されます。

ワークフローの遷移

ワークフローの遷移により、現在のアクティビティから別のアクティビティへ移行できます。遷移は、イベントにより発生します。

1つのイベントを複数の遷移に関連付けることができます。

ワークフローのタスク

ワークフローのタスクは、実行するように割り当てられた業務で、アクティビティをトリガすると発生します。

ワークフローのアクティビティの担当者

アクティビティの担当者は、ワークフローのアクティビティから発生する実行する責任者です。

ワークフローの実行グループの定義

ワークフローの実行グループを使って、定義しているワークフロースキームを分類できます。ワークフロースキームが属する実行グループは、ワークフロースキームの詳細で指定されます。

概要

Asset Managerでワークフローを作成するには、まず［**ツール／ワークフロー／ワークフロースキーム**］メニューを選択して、グラフィカルエディタを表示し、ワークフロースキームを定義します。ワークフロースキームでは、アクティビティ、イベント、および遷移を定義します。これらは、Asset Managerのアクションと従業員（ワークフローの担当者）を参照できます。

警告:

ワークフロースキームは、Asset Manager Windowsクライアントでのみ編集できます。

ワークフロースキームは、ワークフローエンジンによって解釈されます。ワークフローエンジンは、Asset Manager ServerまたはAsset Managerのエージェントで実行されます。

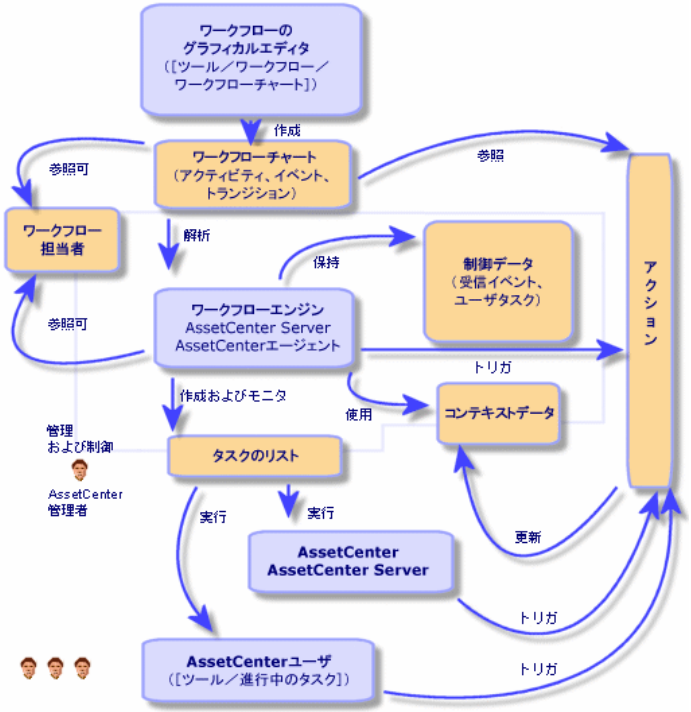
ワークフローエンジンは、イベントに反応して、ワークフローをトリガし、ワークフローインスタンスをモニタします。

- 実行するタスクを生成する。
- これらのタスクと、アクティビティを起動するイベントをモニタする。
- 受信イベントと実行するユーザタスクのログを取ることで、イベントの経過を記録する。

ワークフローのタスクは、ワークフローエンジンまたはAsset Managerユーザによって実行されます。実行されると、さらにイベントが起動され、ワークフローエンジンにその情報が伝えられます。

次の図はワークフローがAsset Managerでどのように実装されているかの概要を示しています。

図 9.2. Asset Managerのワークフロー - 概要



ワークフローインスタンスの処理は、アクティビティやイベントの属性、およびこれらを制御するワークフローエンジンの設定によって異なります。

ワークフローの導入方法

Asset Managerのワークフローを使って、会社の業務を実行することができます。手順は次のとおりです。

1. 形式化したい会社の業務を分析します。
2. 次の要素を作成します。
   1. ワークフローの役割
   2. アクション
3. 次の要素を定義してワークフロースキームを作成します。
   1. アクティビティ、イベント、遷移
   2. アラーム
4. ワークフローの適切な実行グループを定義します。各ワークフロースキームをワークフローの実行グループに関連付けます。
5. 1台または複数のマシンでAsset Manager Automated Process Managerを起動します。Asset Manager Automated Process Managerの各インスタンスで、サーバがモニタするワークフローの実行グループと、モニタ用パラメータを定義します。

ワークフローのグラフィカルエディタの使用方法

ワークフロースキームにアクセスするには、［ツール／ワークフロー／ワークフロースキーム］メニューを使います。このメニューを使用できるのはAsset Managerの管理者だけです。

警告:

ワークフローエディタは、Asset Manager Windowsクライアントでのみ使用できます。

ワークフロースキームの詳細画面の［**アクティビティ**］タブページでは、ワークフロースキームを簡単に作成できるグラフィカルエディタを使用できます。

ここでは、このグラフィカルエディタを使って、ワークフロースキームを作成、変更、削除する方法について説明します。

- アクティビティ
- イベント
- 遷移

- その他の機能

アクティビティ

アクティビティを作成するには：

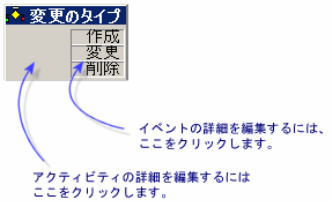
- ［**アクティビティ**］タブページの空白の領域をマウスの右ボタンでクリックして、ショートカットメニューから［**アクティビティの追加**］コマンドを選択します。アクティビティの詳細画面が表示されます。
- ボタンをクリックし、次にグラフィカルエディタ枠をクリックします。アクティビティの詳細画面が表示されます。

アクティビティを削除するには：

- アクティビティをクリックして選択し、［Delete］キーを押します。
- アクティビティを選択し、ショートカットメニューから「**削除**」コマンドを選択します。
- アクティビティを選択し、ボタンをクリックします。

アクティビティの詳細設定を変更するには：

- アクティビティをマウスの右ボタンでクリックし、ショートカットメニューから［**アクティビティの詳細**］を選択します。
- アクティビティをダブルクリックします。



イベント

アクティビティに含まれるイベントには2種類あります。

［データベース］または［定期的］タイプのイベント

アクティビティに出力イベントを追加するには：

- アクティビティを右クリックし、ショートカットメニューから［**イベントの追加**］コマンドを選択します。
- アクティビティを選択し、ボタンをクリックします。

アクティビティから［**データベース**］または［**定期的**］タイプの出力イベントを削除するには：

- イベントを選択し、［Delete］キーを押します。
- イベントをダブルクリックし、ショートカットメニューから［**削除**］コマンドを選択します。
- イベントを選択し、 ボタンをクリックします。

［**データベース**］または［**定期的**］タイプのイベントの詳細を変更するには：

- イベントをダブルクリックします。
- イベントを選択し、ショートカットメニューから［**イベントの詳細**］を選択します。

システムイベント

アプリケーションには、システムイベントとユーザイベントという2種類のイベントがあります。システムイベントは、アクティビティの結果としてアプリケーションによりトリガされるイベントです。ユーザイベントは、例えば質問タイプのイベントの結果として、ユーザによりトリガされます。

イベントタイプは、**seType**フィールド(フィールドは表示されません)で定義されます。

遷移

遷移を作成するには：

1. 開始イベントをクリックして選択します。
2. マウスボタンを押したまま、リンク先のターゲットアクティビティにドラッグします。

遷移を削除するには：

- 遷移をクリックして選択し、［Delete］キーを押します
- 遷移を選択し、ショートカットメニューから「**削除**」コマンドを選択します。
- 遷移を選択し、 ボタンをクリックします。

遷移の詳細設定を変更するには：

1. 遷移をクリックして選択します。
2. ショートカットメニューから［**遷移の詳細**］を選択します。

遷移のソースまたはターゲット、またはその両方を変更するには、次の手順に従います。

1. 遷移を選択します。
2. 遷移の変更したい方の端をドラッグします。

その他の機能

グラフィカルエディタでは次のような操作も行えます。

- リンクしているアクティビティと遷移のドラッグ&ドロップ

- ［ズーム］スライダまたは ボタンを使ったスキームの拡大と縮小

依頼の承認手続きで用いられるワークフローの例

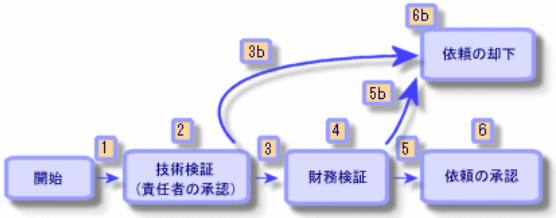
ここでは、簡単なワークフローの例について説明します。

1 目的 [献 158]
2 前提条件 [献 159]
3 アクティビティを作成する [献 164]
4 アクティビティと同時に作成されたイベントの設定 [献 170]
5 開始イベントを作成する [献 170]
6 遷移を作成する [献 171]
7 ワークフローインスタンスの起動例 [献 172]

目的

このワークフロースキームの目的は以下の手順に従って購入依頼の承認手続きを自動化することです。

図 9.3. ワークフロー - 依頼の検証



ワークフロースキームの手順は次のとおりです。

1 ワークフローインスタンスは、購入依頼が検証されるべき状況になると（つまり、購入依頼の［**ステータス**］（SQL名：seStatus）フィールドが［**承認待ち**］に設定された時）開始します。（1）
2 まず責任者の許可をもらいます。この手順では、依頼を部署の責任者に提出して承認を得ます。依頼はメッセージで通知されます。承認依頼メッセージを発信した日の翌営業日の終わりまでに承認者が依頼を処理しない場合は、催促のアラームをトリガするようにプログラムします。（2）
3 責任者が購入依頼を検証したら、次は財務検証の段階に移ります。(3)
それ以外の場合は、依頼は却下されます。（3b）

4 財務検証では、依頼を会社の財務責任者である木村氏に提出します。ここでも依頼はメールで通知され、承認を催促するアラームも設定されます。(4)

5 財務責任者が購入依頼を許可すると、購入依頼が承認されます。

それ以外の場合は、依頼は却下されます。（5b）

6 購入依頼が承認されると、購入依頼の詳細画面の［**ステータス**］フィールドが［**承認済**］に設定されます。

購入依頼が却下された場合、購入依頼の詳細の［**依頼ステータス**］フィールドが［**却下済**］に設定されます。

前提条件

「**Admin**」ログインを使ってデータベースに接続し、メッセージシステムを設定する必要があります（▶ メッセージ [献 147]）。

ワークフローの担当者を作成する

このワークフロースキームに関わる担当者は次のとおりです。

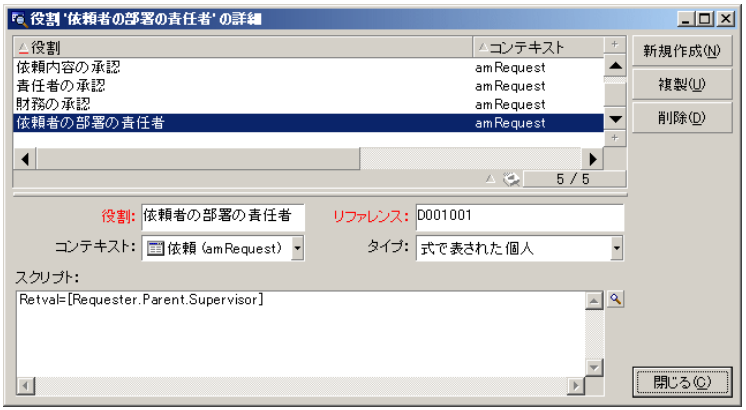
- 依頼者の部署の責任者
- 会社の財務責任者、山田啓介氏

ワークフローの担当者は、［部署と従業員］テーブルに定義された従業員であり、ワークフローの1つまたは複数の段階で操作することが認められています。

これらのワークフローの担当者は［役割］テーブル（SQL名：amWfOrgRole）に作成および保存されます。これらの担当者を作成するには、［**ツール／ワークフロー／役割**］メニューを選択して［**新規作成**］をクリックします。

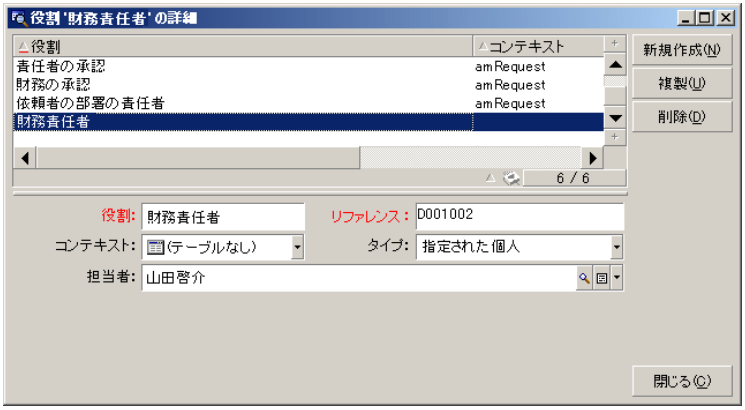
依頼者の部署の責任者

この担当者はスクリプトによって計算されます。このスクリプトを定義するには、役割の詳細画面で次のように入力します。



財務責任者

この担当者は山田啓介氏として指定します。役割の詳細画面で次のように入力します。

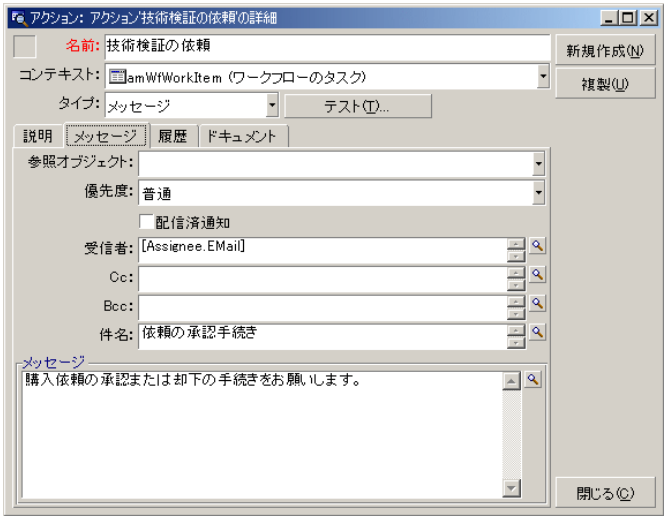


アクションを作成する

ワークフロースキームでは、数多くのアクションを使います。これらを作成するには、［**ツール／アクション／編集**］メニューを選択します。

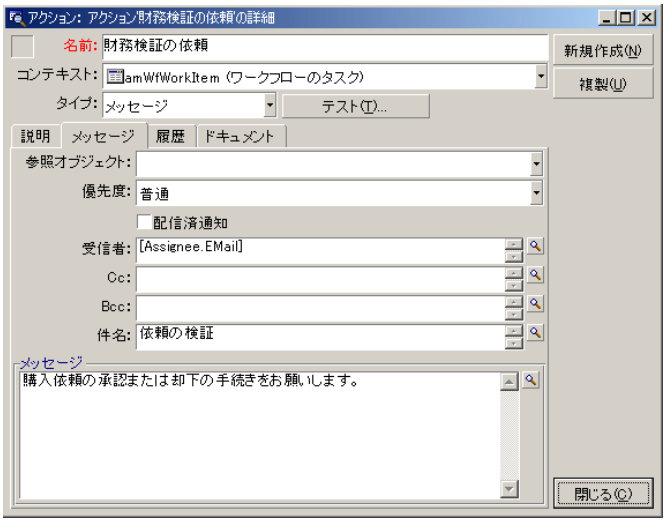
依頼者の部署の責任者に送られる技術検証の依頼

このアクションは、技術検証の段階で使われます。このアクションで、この依頼を検討する必要があることを技術検証の担当者に通知できます。



財務責任者に送られる財務検証の依頼

このアクションは、財務の承認プロセスで使われます。このアクションにより、この依頼の検討が必要であることを伝えるメッセージが、財務検証の担当者に送信されます。



購入依頼の検証

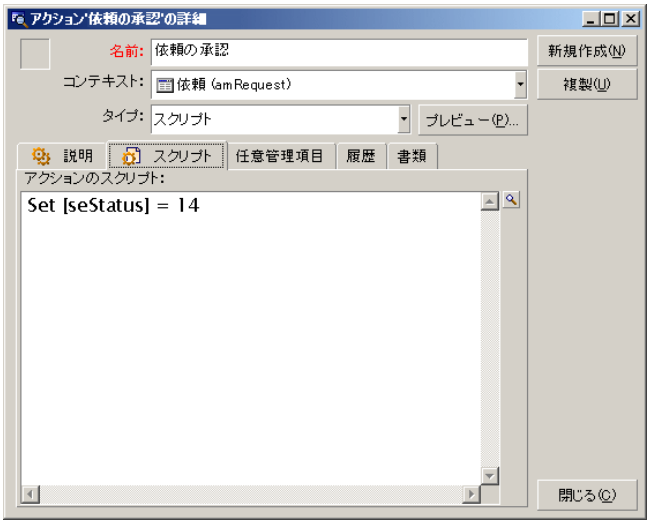
このアクションは、後に作成する「**依頼の承認**」アクティビティの段階で使用されます。

このアクションにより、購入依頼の詳細画面の［**ステータス**］（SQL名：seStatus）フィールドが［**承認済**］に設定されます。このアクションは、［**スクリプト**］タイプのアクションです。

［**ステータス**］フィールドは、システムリストデータです。値のリストを表示するには、次の手順に従ってこのフィールドのヘルプを表示します。

1 依頼の詳細画面の［**ステータス**］フィールドをマウスの右ボタンでクリックします。
2 ショートカットメニューから［**フィールドのヘルプ**］メニューを選択します。［**承認済**］として表示される値は、データベースには「**14**」として保存されます。

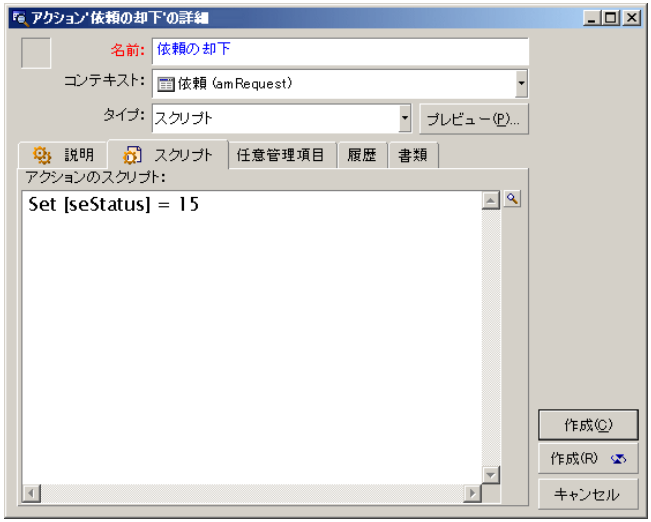
アクションの詳細は次のようになります。



購入依頼の却下

このアクションは、後に作成する「**依頼の却下**」アクティビティの段階で使用されます。

「**依頼の承認**」アクションに似ていますが、購入依頼の詳細画面の［**全般**］タブページにある［**ステータス**］（SQL名：seStatus）フィールドは、［**却下済**］になります。

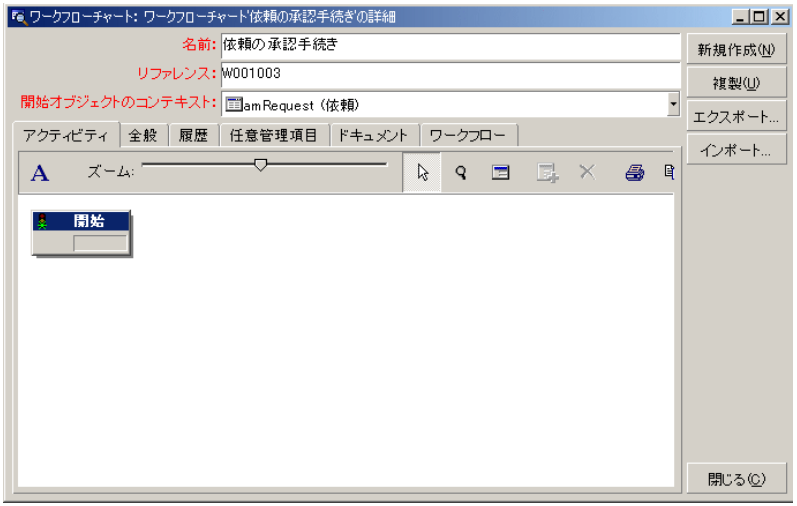


カレンダを作成する

カレンダのリストには、［**ツール／カレンダ**］メニューからアクセスできます。このカレンダはワークフロースキームのアクティビティに関連付けられます。これにより、ワークフロースキームの処理の締切りを設定できます。

ワークフロースキームを準備する

1 ［**ツール／ワークフロー／ワークフロースキーム**］メニューを選択します。
2 ［**新規作成**］をクリックします。
3 ワークフローに「**依頼の承認**」という名前を付けます。
4 ワークフロースキームのすべてのアクティビティにデフォルトで適用される開始オブジェクトのコンテキストを指定します。この例では、［依頼］のテーブル（SQL名：amRequest）です。
5 ［**作成**］をクリックします。開始アクティビティ（「**開始**」）は、［**アクティビティ**］タブページのグラフィカルエディタで自動的に作成されます。



アクティビティを作成する

アクティビティは、ワークフロースキームの詳細画面の［**アクティビティ**］タブページ内で、図を使って作成できます。

1 ワークフローのオブジェクト以外の場所にカーソルを置きます。
2 右クリックします。
3 ショートカットメニューから［**アクティビティの追加**］メニューを選択します。アクティビティの詳細画面が表示されます。

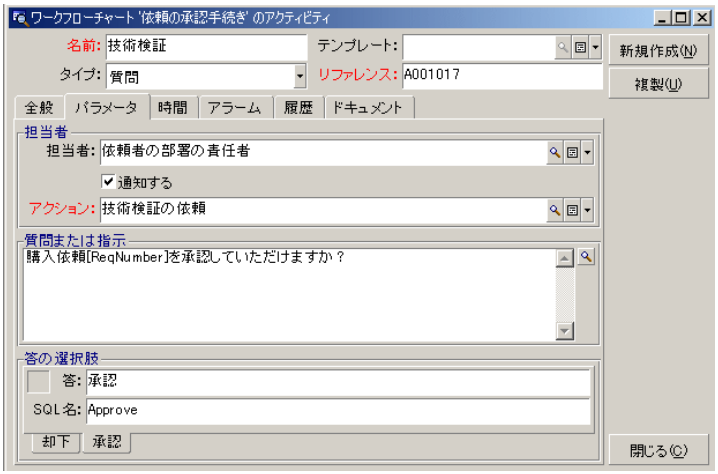
「技術検証」アクティビティを作成する

1 アクティビティに「**技術検証**」という名前を付けます。
2 部署の責任者に承認の依頼を提出するアクティビティなので、［**タイプ**］フィールド（SQL名：seType）のドロップダウンリストから［**質問**］という値を選択します。

3 ［**全般**］タブページの［**コンテキスト**］（SQL名：ContextTable）フィールドは変更しません。

「技術検証」アクティビティの設定

1 ［**パラメータ**］タブページに以下のように入力します。



2 尋ねる質問事項を指定します。
   1 質問のテキストが購入依頼の番号を参照するように指定します。
   2 却下と承認の2つの回答を用意します。質問に対する答を説明するサブタブページを追加するには、サブタブのラベル枠をマウスの右ボタンでクリックし、［**リンクレコードの追加**］または［**リンクレコードの複製**］を選択します。

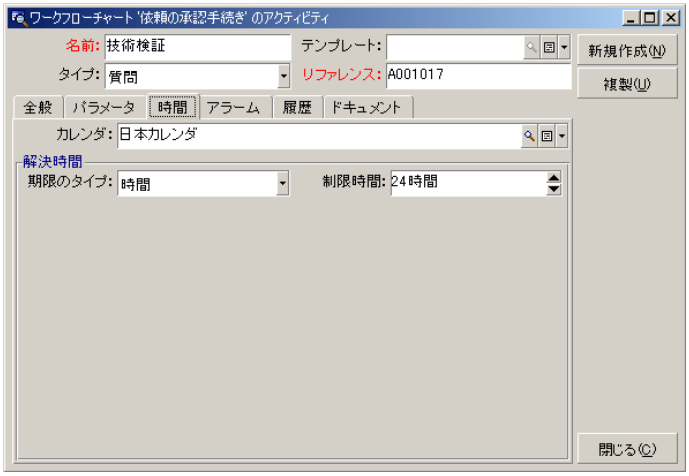
3 ［**担当者**］フィールド（SQL名：Assignee）に質問の宛先を指定します。今の場合、ワークフローの担当者は依頼者の部署の責任者です。この担当者は、準備段階でワークフローの役割のテーブルに作成されています。

4 ワークフローの担当者にこの依頼を検討するよう通知するには、次のように指定します。
   1 ［**通知する**］フィールド（SQL名：bNotifAssignee）を［**はい**］に設定します。
   2 実行するアクションを指定します。これは、準備段階で作成した「**技術検証の依頼**」アクションです。このアクションは、購入依頼の技術検証が開始されると、自動的にトリガされます。

「技術の承認」アクティビティの制限時間を指定する

アクティビティの詳細画面の［**時間**］タブページで次の内容を指定します。

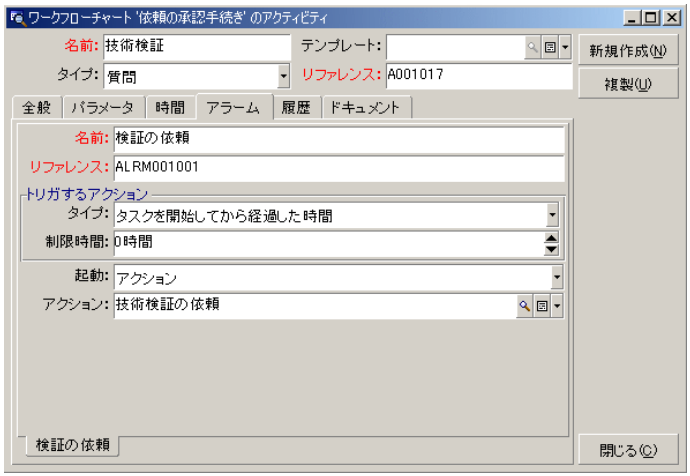
1 アクティビティにリンクする業務用カレンダを指定します。このカレンダを使って制限時間が計算されます。前の準備段階で作成した**「日本カレンダ」**を選択します。
2 アクティビティがトリガされる時間を基準にして、決定を下さなければならない時間を定義します。この例では、ワークフローの担当者は24時間以内に質問に答えることにします。



「技術検証」アクティビティのアラームを定義する

［**時間**］タブページで指定した制限時間までに決定が下されなかった場合に、催促のメッセージを送信するためのアラームを、アクティビティの詳細画面の［**アラーム**］タブページで定義します。

処理を簡単にするために、このアラームでは**「技術検証の依頼」**アクションをトリガすることにします。



ショートカットメニューの［**リンクレコードの追加**］コマンドを使って、さらにアラームを定義することもできます。

イベント

アクティビティを作成すると、質問に対する2種類の回答に対応する2つのシステムイベント（**「承認」**と**「却下」**）が作成されます。

これらのイベントが発生すると、Asset Managerエージェントは、これらを［ワークフローの基本イベント］テーブル（SQL名：amWfOccurEvent）に記録します。

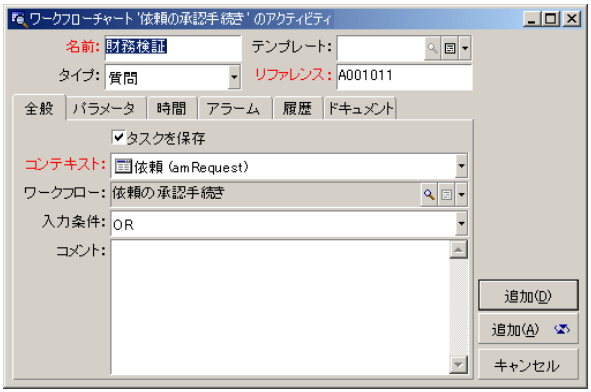
Asset ManagerまたはAsset Manager Automated Process Managerにより、ワークフローのイベントの設定に応じて次のアクティビティがトリガされます。

- ［**処理**］（SQL名：seProcessingMode）フィールドが［**イベントを保存してすぐに処理**］または［**イベントを保存しないですぐに処理**］の場合、Asset Managerが次のアクティビティをトリガします。
- ［**処理**］（SQL名：seProcessingMode）フィールドが［**イベントを保存してからサーバで処理**］の場合、Asset Manager Automated Process Managerが次のアクティビティをトリガします。

イベントの詳細画面の［**全般**］タブの［**処理**］（SQL名：seProcessingMode）フィールドのデフォルト値は、［**イベントを保存してからサーバで処理**］です。

「財務の承認」アクティビティを作成する

このアクティビティは、前のアクティビティと類似しています。



- アクティビティの担当者は異なります。今の場合、会社の財務責任者である木村五郎氏（指定された個人）です。この担当者は、前の手順で財務責任者の役割と共に［役割］テーブル内に作成されました。この担当者に通知するために、**「財務検証の依頼」**アクションを選択します(［**アクション**］フィールド)。
- 期限とアラームは、技術検証アクティビティに対する方法とほぼ同じ方法で作成します。

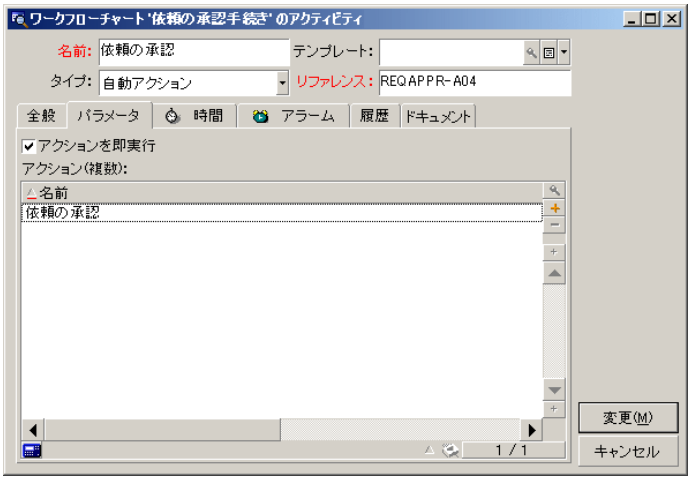
「依頼の承認」アクティビティを作成する

依頼が2つの承認プロセスを問題なく通過すると、この依頼は承認されたことになります。

「依頼の承認」アクティビティはこのワークフロースキームで起こりうる結末の1つです。

このアクティビティでは、依頼の詳細を変更して、依頼が承認されたことを指定する必要があります。

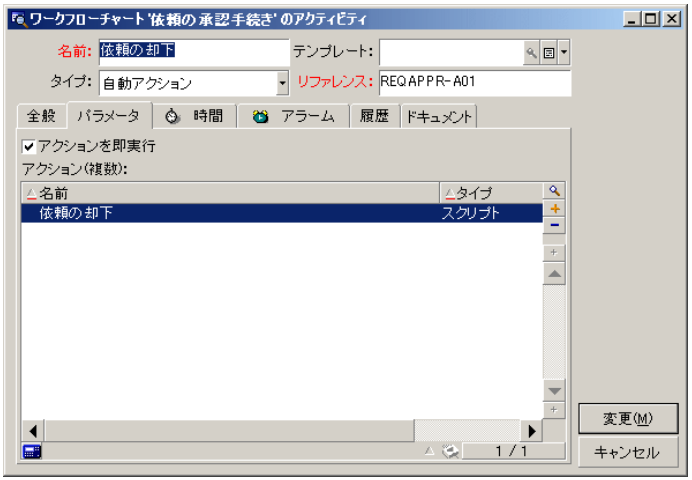
このアクティビティは［**自動アクション**］タイプのアクティビティです。実行されるアクションは、前の手順で作成した「**依頼の承認**」アクションです。



「依頼の却下」アクティビティを作成する

「**依頼の却下**」アクティビティは「**依頼の承認**」アクティビティと類似しています。

今の場合は、依頼の詳細を変更して、依頼が却下されたことを指定する必要があります。実行されるアクションは、前の手順で作成した「**依頼の却下**」アクションです。



アクティビティと同時に作成されたイベントの設定

前の段階でアクティビティが作成された時に、以下のイベントも同時に作成されました。

- **「技術検証」**アクティビティ
  - **「承認」**イベント
  - **「却下」**イベント
- **「財務の承認」**アクティビティ
  - **「承認」**イベント
  - **「却下」**イベント
- **「依頼の承認」**アクティビティ
  - **「実行済」**イベント
- **「依頼の却下」**アクティビティ
  - **「実行済」**イベント

各イベントを次々に選択して以下のオプションを実行すると、Asset Manager Automated Process Managerに依存する必要がありません。

1. イベントの名前を右クリックします。
2. **［イベントの詳細］**メニューを選択します。
3. **［全般］**タブを選択します。
4. **［処理］**（SQL名：seProcessingMode）フィールドで**［イベントを保存してすぐに処理］**を選択します。
5. **［変更］**をクリックします。

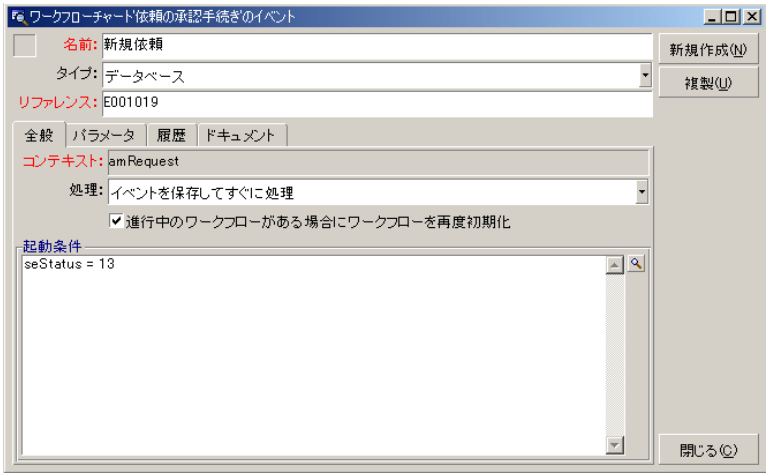
開始イベントを作成する

ワークフローをトリガするイベントは、**「開始」**アクティビティに関連付けられます。

「開始」イベントを定義するには、**「開始」**アクティビティのイベント枠をマウスの右ボタンでクリックし、**［イベントの詳細］**を選択します。

1. この例では、ワークフローは、依頼の詳細画面の**［依頼ステータス］**（SQL名：seStatus）フィールドが**［承認待ち］**になるとトリガされます。

従って、開始イベントは**「データベース」**タイプのイベントで、次のように［**全般**］タブページにパラメータを指定します。



2 イベントの［**全般**］タブページにある［**処理**］（SQL名：seProcessingMode）フィールドを［**イベントを保存してすぐに処理**］に設定します。
3 イベントの［**パラメータ**］タブページで、次の操作を行います。
   1 ［**更新**］（SQL名：bUpdate）チェックボックスをオンにします。
   2 ［**モニタするフィールド**］（SQL名：MonitFields）フィールドに［**seStatus**］フィールドを指定します。

遷移を作成する

アクティビティを作成したら、遷移でリンクする必要があります。

遷移を作成するには：

1 遷移の開始イベントをクリックします。
2 マウスボタンを押したまま、ターゲットアクティビティまでイベントをドラッグします。

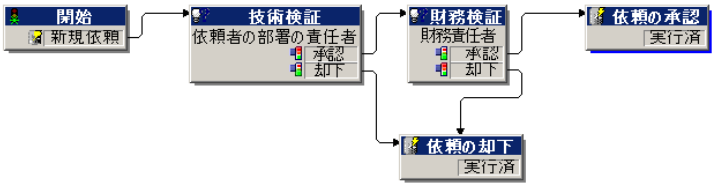
今の例では、以下の遷移を作成します。

- 開始イベントから**「技術検証」**アクティビティまで
- **「技術の承認」**アクティビティの**「承認」**イベントから**「財務の承認」**アクティビティまで
- **「財務検証」**アクティビティの**「承認」**イベントから**「依頼の承認」**アクティビティまで

- **「技術承認」**アクティビティおよび**「財務検証」**アクティビティの**「却下」**イベントから**「依頼の却下」**アクティビティまで

この結果、ワークフロースキームは次のようになります。

図 9.4. ワークフロースキーム - 依頼の承認



ワークフローインスタンスの起動例

ワークフロースキー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、次の操作を実行します。

1. ［部署と従業員］テーブルに入力する [献 172]
2. 承認を受ける購入依頼を作成する [献 173]
3. ワークフローを制御する [献 173]

［部署と従業員］テーブルに入力する

承認を受ける購入依頼を作成する前に、［部署と従業員］テーブルに依頼者と依頼者の部署の責任者を定義する必要があります。責任者には、以下の操作を実行するための適切な権限が必要です。

1. **「情報システム部」**に所属する依頼者**「田中伸一」**を作成します。
2. 依頼者が購入依頼を作成できるように、田中伸一にログイン**「田中」**、パスワード、およびユーザプロファイルを指定します（対応するレコードの詳細画面の**［プロファイル］**タブページで指定）。デモ用データベース内の**「依頼者」**プロファイルを使用できます。
3. **［電子メール］**（SQL名：EMail）フィールドに**「AM:Tanaka」**を入力します。
4. **「情報システム部」**の責任者は**「東郷健二」**です。
5. 後の操作を簡単にするために、「東郷健二」にデータベースの管理者権限を与えます。対応するレコードの詳細画面の**［プロファイル］**タブページを表示し、**［管理者権限］**チェックボックス（SQL名：bAdminRight）をオンにします。東郷健二の**［ログイン］**（SQL名：UserLogin）とパスワードを指定します。

6 ［**電子メール**］（SQL名：EMail）フィールドに「**AM:Tanaka**」を入力します。
7 従業員「**山田啓介**」を選択します。
8 ［**電子メール**］（SQL名：EMail）フィールドに「**AM:Yamada**」を、［**ログイン**］（SQL名：UserLogin）フィールドに「**山田**」を入力します。

承認を受ける購入依頼を作成する

次の手順に従って、承認を受ける購入依頼を作成します。

1 田中伸一のログイン名を使ってデモ用データベースに接続します。
2 Asset Managerの［**調達／購入依頼**］メニューを選択します。
3 ［**新規作成**］をクリックします。
4 ［**カスタム依頼を新規作成する**］オプションを選択します。
5 ［**OK**］をクリックします。
6 依頼の詳細画面の［**全般**］タブページにある［**依頼者**］フィールド（SQL名：Requester）で、「**田中**」のレコードを選択します。
7 依頼の詳細画面の［**依頼ステータス**］フィールド（SQL名：seStatus）を［**承認待ち**］に設定します。
8 購入依頼の作成を確定します。開始イベントが発生し、［ワークフローの基本イベント］（SQL名：amWfOccurEvent）テーブルにイベントが記録されます。

ワークフローを制御する

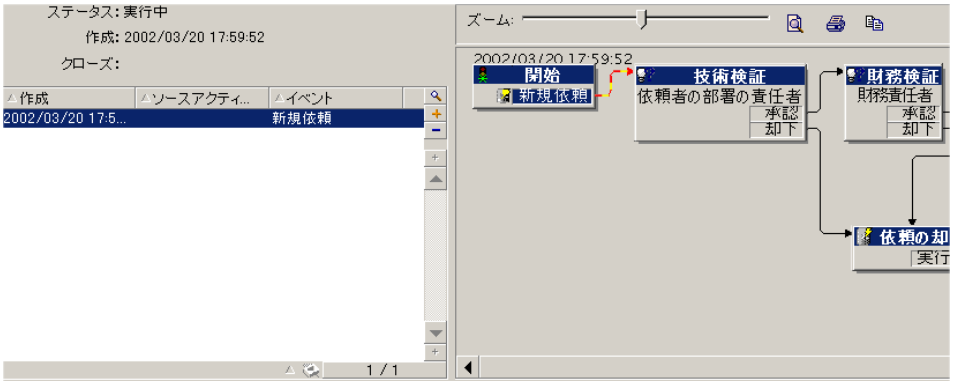
ワークフローが正常に機能することを確認するために、Asset Managerを起動し、「田中伸一」の部署の責任者である「東郷健二」のログインを使ってデモ用データベースに接続します。

ワークフローを表示する

作成した購入依頼の詳細画面を表示します。［**ワークフロー**］タブページには、依頼の現在のワークフローが一覧表示されます。各ワークフローの詳細は、個々のサブタブページに表示されます。

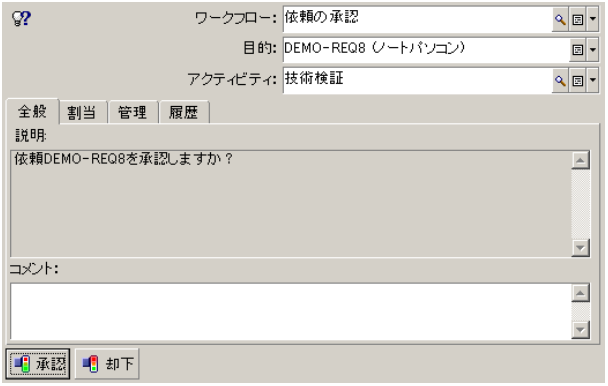
- サブタブページの左側には発生したイベントが一覧表示されます。

- 右側には、ワークフローのステータスが表示されます。ワークフロースキームの「**技術検証**」が点滅しているはずです。



実行するタスクを表示する

1 ［**ツール／進行中のタスク**］メニューを選択します。部署の責任者である「東郷健二」は、この方法で割り当てられたタスクの詳細を表示することができます。



［**全般**］タブページには、「**技術検証**」アクティビティの［**質問**］タブページで定義した質問が表示されます。

［**割当**］タブページには、タスクの担当者と締切りが表示されます。この日付は、「**技術検証**」アクティビティの詳細画面の［**時間**］タブページで指定した情報と、タスクの作成日（つまりAsset Manager Serverによる遷移の起動日）を使って計算されます。

［詳細］をクリックすると、タスクの発生元である依頼の詳細画面にアクセスできます。

2 ［承認］ボタンまたは［却下］ボタンをクリックして、タスクを実行します。［全般］タブページに、決定に関する情報を入力することもできます。

ワークフローのコンテキスト

各ワークフローインスタンスには固有のコンテキストがあります。

ワークフローインスタンスのコンテキストを定義する

ワークフロースキームを定義する場合は、次の要素を定義します。

- ワークフローのデフォルトのコンテキスト
- 全てのイベント、遷移、ワークフローのアクティビティのコンテキスト（遷移の詳細画面、またはイベントおよびアクティビティの詳細画面の［全般］タブページで指定）。このコンテキストは、ワークフローのデフォルトのコンテキストにリンクされます。

いずれの場合も、コンテキストはテーブルです。

ワークフローインスタンスが参照するオブジェクト

レコードが、ワークフロースキームで定義された起動条件を満たしたときに、ワークフローインスタンスがトリガされます。レコードは、開始イベントが参照するオブジェクトです。

ワークフローインスタンスの実行中に、参照されるオブジェクトは、アクティビティ、イベント、ワークフローの遷移のレベルで定義したコンテキストによって変わります。

例：購入依頼が承認されると、ワークフローインスタンスがトリガされます。この依頼に従って発注が作成されます。依頼R1が承認されると、このR1が開始イベントの参照オブジェクトになります。その後参照オブジェクトは発注PO1、つまり購入依頼から作成された発注になります。

特定のオブジェクトに対して同時に実行できるワークフローインスタンスを制限する

［1つのオブジェクトにつき1つのワークフローを起動］オプション（SQL名：bUniqueActive）

Asset Managerでは、ワークフローの詳細画面の［**全般**］タブページにある［**1つのオブジェクトにつき1つのワークフローを起動**］オプションを使って、選択しているオブジェクトで同時に実行できるワークフロー数を制限できます。

あるオブジェクトに対して2つ目のワークフローインスタンスを起動する「開始」アクティビティの出力イベントが発生した場合、［**1つのオブジェクトにつき1つのワークフローを起動**］オプションと［**進行中のワークフローがある場合にワークフローを再度初期化**］オプション（SQL名：bReinitialize）（イベントの詳細画面の［**全般**］タブ）によって出力結果が変わります。

オプションの選択に応じた結果は、次の表のとおりです。

		ワークフロースキームの［全般］タブページの［1つのオブジェクトにつき1つのワークフローを起動］オプション	
		オン	オフ
「開始」アクティビティの出力イベントの［全般］タブページにある［進行中のワークフローがある場合にワークフローを再度初期化］オプション	オン	そのオブジェクトに対して既に実行中のワークフローインスタンスがある場合、そのインスタンスは停止し、新しいワークフローインスタンスが開始します。	
	オフ	そのオブジェクトに対して既に実行中のワークフローインスタンスがある場合、イベントは無視されます（新しいワークフローインスタンスは開始しません）。	新しいワークフローインスタンスが作成されます。

応用例：

購入依頼の承認処理を目的とするワークフロースキームの場合は、次のように設定すると便利です。

- ［**1つのオブジェクトにつき1つのワークフローを起動**］オプションをオンにして、対象となる購入依頼を単一の承認プロセスで処理する。
- 開始イベントで［**進行中のワークフローがある場合にワークフローを再度初期化**］オプションをチェックして、購入依頼の構成が変更された場合にワークフローインスタンスを再起動する。

ワークフローの役割

特定のワークフローのアクティビティから発生するタスクは、担当者が実行する必要があります。

注意:

アクティビティの担当者は、［質問］または［ユーザのアクション］タイプのアクティビティだけに必要です。［自動アクション］または［テスト/スクリプト］タイプのアクティビティには不要です。

アクティビティの担当者は、［ワークフローの役割］テーブル（SQL名：amWfOrgRole）で選択します。［ワークフローの役割］テーブルにアクセスするには、ナビゲーションバーの［**管理/ワークフロー/ワークフロー役割**］リンクを使用します。

ワークフローの役割のタイプ

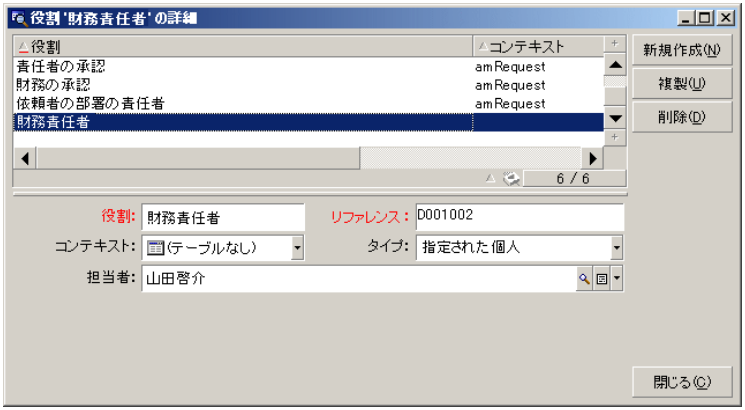
ワークフローの役割には次のタイプがあります（ワークフローの役割の詳細画面／［**タイプ**］（SQL名：seType）フィールドで指定）。

- ［**指定された個人**］
- ［**式で表された個人**］
- ［**グループ**］
- ［**式で表されたグループ**］

［指定された個人］

この場合担当者は、［部署と従業員］テーブルから直接選択されます。

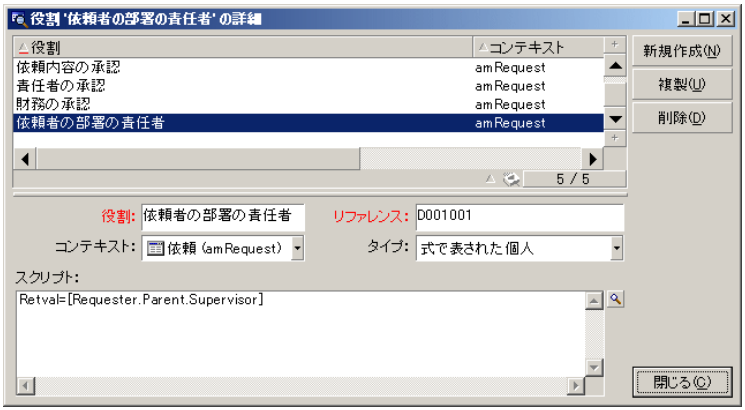
例：



［式で表された個人］

この場合担当者は、［部署と従業員］テーブルに属しますが、スクリプトによって決定されます。

例：



［グループ］

この場合、担当者は［従業員グループ］テーブル（SQL名：amEmplGroup）から選択されます。

［式で表されたグループ］

この場合、担当者は［従業員グループ］テーブル（SQL名：amEmplGroup）内のスクリプトによって計算されます。

アクティビティの担当者を定義する

［質問］タブ（**［質問］**タイプのアクティビティ）または**［アクション］**タブ（**［ユーザのアクション］**タイプのアクティビティ）の**［担当者］**（SQL名：Assignee）フィールドを使って、**［質問］**タイプまたは**［ユーザのアクション］**タイプのアクティビティの担当者を定義できます。

ワークフローのアクティビティ

アクティビティは、次の2つのグループに大別できます。

- ユーザの操作を必要とするアクティビティ：**［質問］**および**［ユーザのアクション］**タイプのアクティビティ（アクティビティの詳細画面の上部にある**［タイプ］**（SQL名：seType）フィールド）

- 自動的に実行されるアクティビティ：［**自動アクション**］および［**テスト/スクリプト**］タイプのアクティビティ

アクティビティの［**タイプ**］フィールドの値によって、アクティビティの詳細画面に表示されるタブが決まります。

ここでは、次の内容について説明します。

- 「質問」タイプのアクティビティ
- 「ユーザのアクション」タイプのアクティビティ
- 「自動アクション」タイプのアクティビティ
- 「テスト/スクリプト」タイプのアクティビティ
- ［開始］アクティビティ
- アクティビティテンプレート
- アクティビティをトリガする

「質問」タイプのアクティビティ

「質問」タイプのアクティビティには、［**担当者**］（SQL名：Assignee）フィールドで指定したユーザの操作が必要になります。

［**質問**］タイプのアクティビティでは、次の内容を定義します。

- 質問または指示
- 答の選択肢

例：

- 購入承認プロセスで、従業員からの依頼が部署の責任者に発信されます。
- ［**質問**］タイプのアクティビティを確認プロセスとして使い、タスクが実行されたことを確認することもできます。この場合、答の選択肢は1つだけです。

［**タイプ**］フィールド（SQL名：seType）フィールドを［**質問**］に設定します。［**質問**］タブページが表示されます。

次の内容を指定します。

1. ワークフローの役割テーブルに含まれている担当者に対応するレコード。この担当者には、Asset Managerで定義するアクションを介して通知できます。通知することを指定するには、［**質問**］タブページの［**通知する**］（SQL名：bNotifAssignee）フィールドで「はい」を選択します。

   注意:

   担当者に通知するアクションは、実行するタスクが作成された時点、つまりアクティビティを発生させる遷移が起動された時点ですぐにトリガされます。

   担当者は、［**ツール／進行中のタスク**］メニューを使って、実行するタスクの詳細情報にアクセスします。
2. 質問または指示の本文

3 答の選択肢。答の説明は個々のサブタブページに入力します。答は、説明とそのSQL名で識別されます。答を追加、複製、または削除するには、サブタブのラベル部分をマウスの右ボタンでクリックし、ショートカットメニューから［**リンクレコードの追加**］、［**リンクレコードの複製**］、または［**リンクの削除**］を選択します。

注意:

答が返ってくると、自動的にアクティビティの出力イベントが作成されます。

「ユーザのアクション」タイプのアクティビティ

「質問」タイプのアクティビティには、［**担当者**］（SQL名：Assignee）フィールドで指定したユーザの操作が必要になります。

次の内容を定義します。

- 指示の内容
- 実行するウィザード

［**タイプ**］フィールドを［**ユーザのアクション**］に設定して、［**アクション**］タブを表示します。

次の内容を指定します。

- 指示の内容
- 実行するウィザード
- ［ワークフローの役割］テーブルに含まれている担当者に対応するレコード。この担当者には、Asset Managerのアクションを介して通知できます。通知することを指定するには、［**アクション**］タブページの［**通知する**］（SQL名：bNotifAssignee）フィールドで「はい」を選択します。

  注意:

  担当者に通知するアクションは、実行するタスクが作成された時点、つまりアクティビティを発生させる遷移が起動された時点ですぐにトリガされます。

  担当者は、［**ツール／進行中のタスク**］メニューを使って、実行するタスクの詳細情報にアクセスします。

注意:

「実行済」イベントが、アクティビティの出力イベントとして自動的に作成されます。

例：受領を管理する時に、納品待ちの発注明細をすべて受領するのか、部分的に受領するのかをウィザードを使って管理できます。

「自動アクション」タイプのアクティビティ

「自動アクション」タイプのアクティビティは、Asset ManagerまたはAsset Manager Automated Process Managerによって自動的に実行されます。

説明

［**自動アクション**］タイプのアクティビティには、実行するアクションが一覧表示されます。

例：「資産の移動」処理では、［**自動アクション**］タイプのアクティビティにより、移動した親資産に属するすべての資産の設置場所が自動的に変更されます。

［**タイプ**］フィールドを［**自動アクション**］に設定して、［**アクション**］タブページを表示します。

実行するアクションのリストをここで指定します。

注意:

「実行済」イベントが、アクティビティの出力イベントとして自動的に作成されます。

実行

アクティビティをトリガする遷移を起動するワークフローエンジンは、アクティビティのアクションを即時に実行します。選択されたオプションに従って、これらのアクションはAsset Manager Automated Process ManagerかAsset Managerのエージェントによって処理されます。

- ［**アクションを即実行**］オプション（SQL名：bExecImmediately）をオンにした場合は、アクティビティをトリガする遷移を起動するワークフローエンジンにより、アクティビティのアクションが自動的に実行されます。
- それ以外の場合は、タスクは次の検証サイクル中にAsset Manager Automated Process Managerによって実行されます。

「テスト/スクリプト」タイプのアクティビティ

これらのアクティビティは、Asset ManagerまたはAsset Manager Automated Process Managerによって自動的に実行されます。

説明

このタイプのアクティビティは、スクリプトとその実行結果の選択肢を使って定義します。

例：在庫や購入依頼管理を行うときに、[テスト／スクリプト]タイプのアクティビティを使って、発注明細が参照している品目が在庫にあり、予約されていないかどうかを確認できます。予約されていない場合は、依頼者に在庫の品目を予約

するかどうかを尋ねる［**質問**］タイプのアクティビティをトリガさせることができます。

［**タイプ**］を［**テスト/スクリプト**］に設定して、［**テスト**］タブを表示します。

次の内容を指定します。

- 実行するテストスクリプト
- テスト結果の選択肢。各結果の説明は個々のサブタブページに入力します。結果は説明とSQL名で識別されます。結果の選択肢を追加、複製、削除するには、サブタブのラベル部分をマウスの右ボタンでクリックし、ショートカットメニューから［**リンクレコードの追加**］、［**リンクレコードの複製**］、［**リンクの削除**］を選択します。

注意:

注意：各結果のSQL名は、テストスクリプトの戻り値と対応させる必要があります。

注意:

結果が出るたびに、自動的にアクティビティの出力イベントが作成されます。

実行

アクティビティをトリガする遷移を起動するワークフローエンジンは、アクティビティのアクションを即時に実行します。選択されたオプションに従って、これらのアクションはAsset Manager Automated Process ManagerかAsset Managerのエージェントによって処理されます。

- ［アクションを即実行］オプション（SQL名：bExecImmediately）をオンにした場合は、アクティビティをトリガさせる遷移を起動するワークフローエンジンによって、アクティビティのアクションが自動的に実行されます。遷移を起動するイベント用に選択した処理モードに従って、Asset Manager Automated Process ManagerまたはAsset Managerのエージェントがアクションを実行します。
- それ以外の場合は、タスクは次の検証サイクル中にAsset Manager Automated Process Managerによって実行されます。

［開始］アクティビティ

［**開始**］アクティビティは、ワークフロースキームの開始点です。

開始アクティビティは必須で、ワークフロースキームの作成時に自動的に作成されます。詳細を編集することはできません。

開始アクティビティは実行する作業を定義しません。

［**開始**］アクティビティの出力イベントによってワークフローインスタンスがトリガされます。

アクティビティテンプレート

アクティビティテンプレートを使うと、ワークフロースキームのアクティビティを迅速に作成できます。

アクティビティテンプレートは、［アクティビティ］（SQL名：amWfActivity）テーブルに保存されます。

アクティビティテンプレートのリストを開くには、［ツール／ワークフロー／アクティビティテンプレート］メニューを選択します。

注意:

注意：アクティビティテンプレートの詳細画面に含まれている情報（アクティビティのタイプなど）を、このテンプレート（アクティビティの詳細画面の［テンプレート］（SQL名：Template）フィールド）を参照しているアクティビティのレベルに自動的にコピーするには、Asset Manager管理者がアクティビティの詳細画面のフィールドとリンクに適切なデフォルト値を定義する必要があります。

アクティビティをトリガする

アクティビティを起動するには、アクティビティの詳細画面の［**全般**］タブページの［**入力条件**］（SQL名：seInCond）フィールドに入力する必要があります。これは、アクティビティをトリガする遷移に関する条件です。

- アクティビティを起動する遷移が1つだけの場合は、（Asset ManagerまたはAsset Manager Automated Process Managerによって）遷移を起動するだけで、アクティビティがトリガされます。
- アクティビティをトリガできる遷移が複数ある場合は、次のように処理されます。
  - アクティビティの入力条件が［**AND**］の場合、アクティビティをトリガするには、アクティビティをトリガできるすべての遷移が起動されなければなりません。
  - アクティビティの入力条件が［**OR**］の場合、遷移を1つ起動するだけで、アクティビティがトリガされます。

注意:

アクティビティの入力条件が複雑な場合（［AND］と［OR］の組合せ）は、中間に［テスト/スクリプト］タイプのアクティビティをいくつも作成して実現することができます。

タスク

ここでは、ワークフローのタスクを作成し、実行する方法について説明します。

タスクを作成する

アクティビティをトリガする遷移が起動すると、遷移を起動したワークフローエンジンによって、実行するタスクが自動的に作成されます。

アクティビティの詳細画面の［**全般**］タブページにある［**タスクを保存**］（SQL名：bLogWorkItem）フィールドで選択したオプションに従って、タスクは［ワークフローのタスク］（SQL名：WkElem）テーブルに記録されます。

［**タスクを保存**］オプションは、次の場合に自動的にオンになります。

- ［**質問**］タイプまたは［**ユーザのアクション**］タイプのアクティビティの場合
- ［**アクションを即実行**］（SQL名：bExecImmediately）オプションが選択されていない［**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の場合

警告:

タスクが記録されないと、このタスクに関連付けられているワークフローのアラームを作成することはできません。［タスクを保存］オプションを選択しないと、アクティビティの詳細画面に［時間］タブおよび［アラーム］タブが表示されません。

タスクは、ユーザの操作が必要かどうかによって、異なる方法で実行されます。

[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

［**アクションを即実行**］（SQL名：bExecImmediately）オプションが選択されている［**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したタスクは、タスクを作成した遷移を起動したワークフローエンジンによって即座に実行されます。このエンジンは、Asset Manager Automated Process ManagerかAsset Managerのエージェントのどちらかです。

それ以外の場合は、Asset Manager Automated Process Managerによって、ワークフローのタスクを実行する必要があるかどうかが定期的に検証され、必要に応じて実行されます。

Asset Manager Automated Process Managerがワークフロー関数をモニタする頻度は、Asset Manager Automated Process Managerのオプションで定義します。

すべてのワークフロータスクのリストを表示する

すべてのワークフロータスクのリストを表示するには、［**ワークフローのタスク**］（amWfWorkItem）テーブル（ナビゲーションバーの［**管理／ワークフロー／ワークフローのタスク**］リンク）を開きます。

表示されるリストには、すべてのワークフローインスタンスの**すべての**ワークフロータスクがステータス（承認待ち、完了、クローズ済）に関係なく表示されます。

管理者は、ワークフロースキームの詳細からも特定のアクティビティのタスクのリストにアクセスできます。手順は次のとおりです。

1 フローチャート内のアクティビティをマウスの右ボタンでクリックします。
2 ショートカットメニューから［**タスク...**］を選択します。

ユーザタスクを実行する

ワークフローの担当者は、ナビゲーションバーの［**ポートフォリオ管理／ワークフローのタスク**］リンクを使用してタスクにアクセスできます。

例えば、ユーザは以下を参照します。

- 自分に割り当てられた、自分が実行しなければならないタスク
- 自分には割り当てられていないけれども、自分が属するグループに割り当てられているタスク

タスクが参照するオブジェクトの詳細にアクセスするには、［**詳細**］をクリックします。

指定されたタスクを実行するには、タスクの［**全般**］タブページを表示します。

- タスクの結果発生するアクティビティが［**質問**］タイプのアクティビティの場合、［**全般**］タブには質問または手順のテキストが表示されます。答に対応するボタンがあるので、適切なボタンをクリックします。必要な場合は、決定に関するコメントも入力できます。
- ［**ユーザのアクション**］タイプのアクティビティの場合は、［**ウィザード**］ボタンをクリックして、実行するウィザードを起動します。

また、担当者は、ナビゲーションバーの［**ポートフォリオ管理／ 委任によるワークフローのタスク**］リンク経由で、委任されたタスクにアクセスできます。▶タスクの委任 [献 186]

ユーザタスクの割り当て

ユーザタスクの割り当てに関する情報は、タスクの詳細画面の［**割当**］タブに表示されます。

必要な権限がある場合は、ユーザタスクの割り当てを変更できます。

- ［**割当**］（SQL名：seAssignment）フィールドの値。
- タスクの担当者

タスクの委任

委任によって、ユーザは特定のユーザ（委任する従業員）から別のユーザ（委任される従業員）にタスクの割り当てを転送します。ユーザは定められた期間または未定の期間の間にこの処理を行います。

委任の作成

委任を作成するには、［**委任を作成／変更...**］（sysCoreDelegation）ウィザードを使用してください。このウィザードにアクセスするには、従業員詳細を使用するか、［**委任**］をクリックするか（Windowsクライアントの場合）、「**Actions...（アクション...）**］ドロップダウンメニューから［**委任**］を選択します（Webクライアントの場合）。

▶ 『**主要テーブル**』ガイドの「**部署と従業員**」の章の「**委任を作成する**」セクションを参照してください。

注意:

委任に対して終了日（dtEndDeleg）が指定されていないと、委任は永続的になります。

重要項目:

委任を有効にするには、委任の詳細の［有効］（bActive）オプションを選択します。

これにより、委任を削除するか、委任の終了日を変更しないかぎり、委任は一時的に無効にすることができます。

警告:

委任を作成する際にはプロファイルまたは承認権限は転送されません。したがって、委任を受理する従業員Bは、委任によって受理するタスクを受け付けるか、却下するために、従業員Aと同じ承認権限を持つ必要があります。

Windowsクライアント：従業員の詳細から既存の委任を表示できます。［**委任**］タブをクリックして、［**許可された委任のプレビュー**］と［**受理した委任のプレビュー**］サブタブをクリックします。詳細については、『**はじめに**』ガイドの「**スケジュールのグラフィック表示（Windowsクライアント）**」の章を参照してください。

委任されたタスク：概要と表示

従業員Aから従業員Bへの委任を作成し、有効にすると、従業員A、または従業員Aがメンバーであるグループに割り当てられたすべてのワークフロータスクは、従業員Aと従業員Bに送信されます。

その結果：

- 従業員Aは、委任後も、ナビゲーションバーの［**ポートフォリオ管理／ワークフローのタスク**］リンク経由ですべての割り当て済みのワークフロータスクを参照できます。
- 従業員Bは、ナビゲーションバーの［**ポートフォリオ管理／委任によるワークフローのタスク**］リンク経由で委任によって自分に転送されたすべてのワークフロータスクを、さらにはナビゲーションバーの［**ポートフォリオ管理／ワークフローのタスク**］リンク経由ですべての割り当て済みのワークフロータスクを参照します。

  委任期間中、従業員Bは従業員Aに割り当てられたすべてのワークフロータスク、および従業員Aがメンバーである承認グループに割り当てられたすべてのワークフロータスクを表示、承認、却下できます。

  従業員Bは、委任後も、ナビゲーションバーの［**ポートフォリオ管理／委任によるワークフロータスク**］リンク経由で、従業員A（または従業員Aがメンバーである承認グループ）に割り当てられているワークフロータスクとして参照できます。

ワークフローのタスクを管理する

ワークフロータスクの管理に関する情報は、タスクの詳細画面の［**管理**］タブページに表示されます。

情報を表示できるのは、管理者権限を持つユーザのみです。

イベント

イベントはアクティビティに関連付けられます。イベントは、他のアクティビティへの遷移をトリガします。

アクティビティの段階のイベントには、3種類のシステムタイプがあります。イベントのシステムタイプは、イベントの詳細画面の［**システムタイプ**］（SQL名：seType）フィールドで定義します。

- ［**システム**］イベント
- ［**ユーザ**］イベント
- ［**アラーム**］イベント

システムイベント

［**システム**］イベントは、アクティビティの作成および変更時にAsset Managerによって自動的に定義されます。

これらは、アクティビティで実行するタスクの各結果に対応します。

- ［**質問**］タイプのアクティビティに対する答

- ［テスト/スクリプト］タイプのアクティビティの結果
- ［ユーザのアクション］または［自動アクション］タイプのアクティビティの場合は、「**Excuted**」（実行済）イベント

例：アクティビティで、答が「はい」または「いいえ」となる質問を尋ねる場合、アクティビティのレベルでは、「はい」と「いいえ」の2つのシステムイベントが作成されます。

アラームイベント

アクティビティの[アラーム]イベントは、イベントをトリガするアクティビティのアラームを定義すると作成されます。

このようなアラームは、アクティビティの詳細画面の［アラーム］タブで定義します。イベントの名前はアラームと同じになります。

ユーザイベント

ユーザイベントは、アクティビティ内で実行されるタスクとは関係ありません。これらは、グラフィカルワークフローエディタから手動で作成します（［イベントの追加］ショートカットメニューを使います）。

注意:

［開始］アクティビティに関連付けられるイベントはユーザイベントです。

次の2つのタイプの［ユーザ］イベントがあります（イベントの詳細画面上部の［タイプ］（SQL名：seMonitoringType）フィールドで指定します）。

- データベース
- 定期的

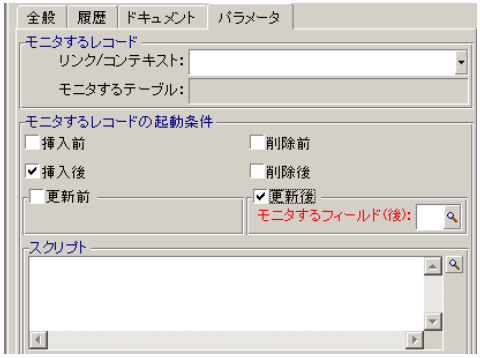
［データベース］タイプのイベント

［データベース］タイプのイベントでは、特定のレコードでワークフローを起動できます。

［データベース］タイプのイベントは、次の場合に発生します。

- ［全般］タブページで指定した一般的な起動条件が満たされた場合
- モニタしているレコードで特定のトリガパラメータが確認された場合

図 9.5. ［データベース］タイプアクションの［パラメータ］タブページ



［データベース］タイプのイベントをトリガするパラメータは、イベントの詳細画面の［パラメータ］タブページに入力します。次の情報を指定します。

- モニタするレコード（これらのレコードは、コンテキストで指定されているテーブル内のレコードでも、またはそのテーブルからリンクされているレコードでも構いません）。モニタするレコードが、コンテキストとして指定されているテーブルにリンクされているレコードの場合は、対応するリンクを［リンク/コンテキスト］（SQL名：LinkToMonitTable）フィールドで指定します。
- モニタ中のレコードに関するイベントの起動条件。起動の条件を指定するには、以下をおこないます。
  - ［挿入前］オプション（SQL名：bPreInsert）では、新規レコードを挿入する前に遷移を起動することができます。

    警告:

    ワークフロースキームがテーブルにレコードを作成し、かつこのレコードにリンクされているアクションやレコードを実行する場合、このオプションは使用できません。

    たとえば、ワークフロースキームがポートフォリオ品目を作成し、ポートフォリオ品目と一緒に自動的に作成された資産のフィールドを変更する場合などです。

    代わりに［挿入後］オプションを使用します。

  - ［挿入後］オプション（SQL名：bInsert）では、新規レコードを挿入した後に遷移を起動することができます。

> **警告:**
>
> **ワークフロースキームがスクリプトタイプアクションを実行して、set()関数でフィールドやリンクを入力する必要がある場合、このオプションは使用できません。**
>
> **代わりに［挿入前］オプションを使用します。**

- ［**削除前**］オプション(SQL名：bDelete)では、レコードを削除する前に遷移を起動できます。
- ［**削除後**］オプション(SQL名：bPostDelete)では、レコードを削除した後に遷移を起動できます。
- ［**更新前**］オプション(SQL名：bPreUpdate)を選択すると、フィールドを指定して、そのフィールドの更新前に遷移が起動されるようにすることができます。これらのフィールドは［**モニタするフィールド(前)**］フィールドで定義します。
- ［**更新後**］（SQL名：bUpdate）オプションを選択すると、［**モニタするフィールド（後）**］フィールド（SQL名：MonitFields）を指定して、指定したフィールドが変更されたときにだけ遷移を起動するようにすることができます。複数のフィールド名を指定する場合は、カンマを使って区切ります。フィールドを空のままにすると、どのフィールドが変更された時でも遷移が起動するようになります。

> **警告:**
>
> **コンテキストが参照するオブジェクトの破棄をイベントの起動条件にすることはできません。**

- ［**スクリプト**］（SQL名：memScript）枠にスクリプトを入力します。スクリプトを入力し、［**挿入**］、［**更新**］、［**削除**］のチェックボックスを必要なだけオンにすると、スクリプトによって起動条件が制限されます。

例：既存の依頼の総額が変更されたときにイベントをトリガする場合は、［**パラメータ**］タブに次のように入力します。

［**データベース**］タイプのイベントが発生すると、そのイベントは、発生したAsset Managerクライアントマシンによって処理されます。処理方法は、イベントの詳細画面の［**全般**］タブページにある［**処理**］（SQL名：seProcessingMode）フィールドで選択したオプションによって決まります。

詳しくは、このガイドの「ワークフロー」の章の「イベントの処理」のセクションを参照してください。

古いリンク

「古い」リンクは、イベントによってモニタされ、いずれかの構成要素が変更される前は現在のレコードを指します。書式[Old.<フィールドのSQL名>]内のスク

リプトで使用され、変更前のレコードフィールドを返します。この方法で、次のようにテストを実行できます。

```
If [Old.lUserId] <> lUserId Then....
```

［定期的］タイプのイベント

［定期的］タイプのイベントは、テーブルで選択された複数のレコードを対象とします。このタイプのイベントを使うと、選択した各レコードに対して定期的にワークフローをトリガできます。

例：毎月、属性が「PC」の資産の残存価額を更新します。

［定期的］タイプのイベントは、**［全般］**タブページに指定されている起動条件が満たされたときに発生します。

この場合、Asset Manager Automated Process Managerによってイベントがトリガされます。

Asset Manager Automated Process Managerが**［定期的］**タイプのイベントをトリガする頻度は、イベントの詳細画面の**［パラメータ］**タブのプランナで定義します。

イベントの処理方法については、このガイドの「ワークフロー」の章、「イベントの処理」のセクションを参照してください。

起動の一般的な条件

すべてのタイプのイベントの起動条件は、**［全般］**タブページで定義できます。

［AQL条件］（SQL名：AQLCond）

［AQL条件］フィールドでは、ワークフロースキームに使われるレコードを指定します。

［進行中のワークフローがある場合にワークフローを再度初期化］（SQL名：bReinitialize）

注意:

［進行中のワークフローがある場合にワークフローを再度初期化］オプションは、「開始」アクティビティの結果起動されるイベントの詳細画面にのみ表示されます。

［進行中のワークフローがある場合にワークフローを再度初期化］オプションは、**［開始］**アクティビティの出力イベントが、同じワークフロースキームの別のワークフローで既に使われているデータベースオブジェクトを必要とする場合の処理を決定します。

このオプションだけでなく、ワークフロースキームの**［全般］**タブページにある**［1つの物件に付き1つのワークフローを起動］**（SQL名：bUniqueActive）オプションも、処理方法を決定します。

オプションの選択に応じた結果は、次の表のとおりです。

表 9.1. ワークフローを制限する場合

		ワークフロースキームの［全般］タブページの［1つのオブジェクトにつき1つのワークフローを起動］オプション	
		オン	**オフ**
「開始」アクティビティの出力イベントの［全般］タブページにある［進行中のワークフローがある場合にワークフローを再度初期化］オプション	**オン**	そのオブジェクトに対して既に実行中のワークフローインスタンスがある場合、そのインスタンスは停止し、新しいワークフローインスタンスが開始します。	
	オフ	そのオブジェクトに対して既に実行中のワークフローインスタンスがある場合、イベントは無視されます（新しいワークフローインスタンスは開始しません）。	新しいワークフローインスタンスが作成されます。

イベントの処理

一般的な起動条件が満たされた場合、イベントは、次の指定に従って処理されます。

- イベントの「タイプ」（イベントの詳細画面上部の［**タイプ**］（SQL名：seMonitoringType）フィールド）
- イベントの詳細画面の［**全般**］タブページにある［**処理**］（SQL名：seProcessingMode）フィールドで選択されるオプション

イベントの様々な処理方法は、次の表のとおりです。

表 9.2. イベントの様々な処理方法

	［イベントを保存してからサーバで処理］	［イベントを保存してすぐに処理］	［イベントを保存しないですぐに処理］
［定期的］タイプのイベント	起動条件が満たされたときに、Asset Manager Automated Process Managerによってイベントがトリガされます。トリガの頻度は、イベントの詳細画面の［パラメータ］タブで定義します。 発生するとすぐに、Asset Manager Serverにより、［amWfOccurEvent］というSQL名のテーブルにイベントが保存されます。 遷移は、Asset Manager Automated Process Managerによって後で起動されます（Asset Manager Automated Process Managerが遷移の起動をモニタする頻度は、Asset Manager Automated Process Managerのオプションで定義します）。	起動条件が満たされたときに、Asset Manager Automated Process Managerによってイベントがトリガされます。トリガの頻度は、イベントの詳細画面の［パラメータ］タブで定義します。 発生するとすぐに、Asset Manager Serverにより、［amWfOccurEvent］というSQL名のテーブルにイベントが保存されます。 遷移は、Asset Manager Automated Process Managerによって即座に起動されます。	起動条件が満たされたときに、Asset Manager Automated Process Managerによってイベントがトリガされます。トリガの頻度は、イベントの詳細画面の［パラメータ］タブで定義します。 イベントが発生しても、イベントはAsset Manager Automated Process Managerによって、［amWfOccurEvent］というSQL名のテーブルに記録されません。ただし、遷移は、Asset Manager Automated Process Managerにより即座に起動されます。

	［イベントを保存してからサーバで処理］	［イベントを保存してすぐに処理］	［イベントを保存しないですぐに処理］
［データベース］タイプのイベント、または**Asset Manager**によってトリガされるシステムイベント（［質問］または［ユーザのアクション］タイプのアクティビティの結果、**Asset Manager**によって実行される［自動アクション］または［テスト/スクリプト］タイプのアクティビティの結果）	イベントが発生するとすぐに、Asset Managerクライアントマシンにより、［**amWfOccurEvent**］というSQL名のテーブルにイベントが保存されます。 遷移は、Asset Manager Automated Process Managerによって後で起動されます（Asset Manager Automated Process Managerが遷移の起動をモニタする頻度は、Asset Manager Automated Process Managerのオプションで定義します）。	イベントが発生するとすぐに、Asset Managerクライアントマシンにより、［**amWfOccurEvent**］というSQL名のテーブルにイベントが保存されます。 遷移はAsset Managerクライアントマシンによって即座に起動されます。	イベントが発生しても、［**amWfOccurEvent**］というSQL名のテーブルにイベントは保存されませんが、遷移は、Asset Managerクライアントマシンによって即座に起動されます。
Asset Manager Automated Process Managerによってトリガされるシステムイベント（**Asset Manager Automated Process Manager**によって実行される［自動アクション］または［テスト／スクリプト］タイプのアクティビティの結果）またはアクティビティアラームに関するイベント	イベントが発生するとすぐに、Asset Manager Serverにより、［**amWfOccurEvent**］というSQL名のテーブルにイベントが保存されます。 遷移は、Asset Manager Automated Process Managerによって後で起動されます（Asset Manager Automated Process Managerが遷移の起動をモニタする頻度は、Asset Manager Automated Process Managerのオプションで定義します）。	イベントが発生するとすぐに、Asset Manager Serverにより、［**amWfOccurEvent**］というSQL名のテーブルにイベントが保存されます。 遷移は、Asset Manager Automated Process Managerによって即座に起動されます。	イベントが発生しても、イベントは［**amWfOccurEvent**］というSQL名のテーブルに記録されません。ただし、遷移は、Asset Manager Automated Process Managerにより即座に起動されます。

表 9.3. イベントの様々な処理方法

	［イベントを保存してからサーバで処理］	［イベントを保存してすぐに処理］
［定期的］タイプのイベント	起動条件が満たされたときに、Asset Manager Automated Process Managerによってイベントがトリガされます。トリガの頻度は、イベントの詳細画面の［**パラメータ**］タブで定義します。 発生するとすぐに、Asset Manager Serverにより、［**amWfOccurEvent**］というSQL名のテーブルにイベントが保存されます。 遷移は、Asset Manager Automated Process Managerによって即座に起動されます。	起動条件が満たされたときに、Asset Manager Automated Process Managerによってイベントがトリガされます。トリガの頻度は、イベントの詳細画面の［**パラメータ**］タブで定義します。 イベントが発生しても、イベントはAsset Manager Automated Process Managerによって、［**amWfOccurEvent**］というSQL名のテーブルに記録されません。ただし、遷移は、Asset Manager Automated Process Managerにより即座に起動されます。
［データベース］タイプのイベント、または**Asset Manager**によってトリガされるシステムイベント（［質問］または［ユーザのアクション］タイプのアクティビティの結果、**Asset Manager**によって実行される［自動アクション］または［テスト/スクリプト］タイプのアクティビティの結果）	イベントが発生するとすぐに、Asset Managerクライアントマシンにより、［**amWfOccurEvent**］というSQL名のテーブルにイベントが保存されます。 遷移はAsset Managerクライアントマシンによって即座に起動されます。	イベントが発生しても、［**amWfOccurEvent**］というSQL名のテーブルにイベントは保存されませんが、遷移は、Asset Managerクライアントマシンによって即座に起動されます。
Asset Manager Automated Process Managerによってトリガされるシステムイベント（**Asset Manager Automated Process Manager**によって実行される［自動アクション］または［テスト／スクリプト］タイプのアクティビティの結果）またはアクティビティアラームに関するイベント	イベントが発生するとすぐに、Asset Manager Serverにより、［**amWfOccurEvent**］というSQL名のテーブルにイベントが保存されます。 遷移は、Asset Manager Automated Process Managerによって即座に起動されます。	イベントが発生しても、イベントは［**amWfOccurEvent**］というSQL名のテーブルに記録されません。ただし、遷移は、Asset Manager Automated Process Managerにより即座に起動されます。

これらの様々なモードを使って、ワークフローの実行方法を正確に指定できます。

次のレベルで行う選択に従って、

- イベントタイプ
- イベント処理モード
- アクティビティ

同期および非同期ワークフロースキームを定義できます。また、両方のアプローチを組み合せることもできます。

応用：同期ワークフロースキームを実行する

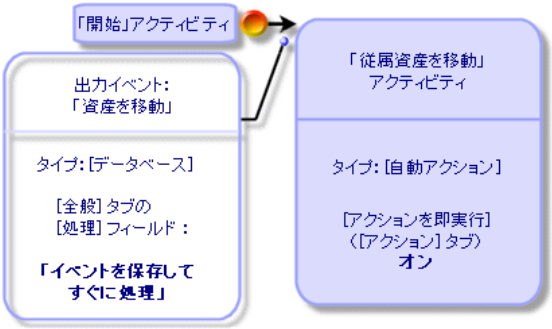
同期ワークフロースキームを作成するには、次の内容を定義する必要があります。

- ［**イベントを保存してすぐに処理**］（イベントの詳細画面の［**全般**］タブページ／［**処理**］（SQL名：seProcessingMode）フィールド）を指定した［**データベース**］タイプのイベント
- そのイベントによってトリガされる、［**アクションを即実行**］（SQL名：bExecImmediately）オプションを指定した［**自動アクション**］、または［**テスト/スクリプト**］タイプのアクティビティ。

例：

下記のワークフロースキームを使うと、資産の設置場所が変更されるのと同時に、その従属資産が自動的に同じ場所へ移動するようになります。

図 9.6. 同期ワークフロースキームの例



この場合、資産の設置場所を変更し［**変更**］をクリックすると、次の処理が行われます。

1. データベーストランザクションが開始します。
2. 資産の設置場所が変更されます。
3. ワークフローが開始します。
4. ワークフローの遷移が起動します。
5. 従属資産の設置場所が変更されます。

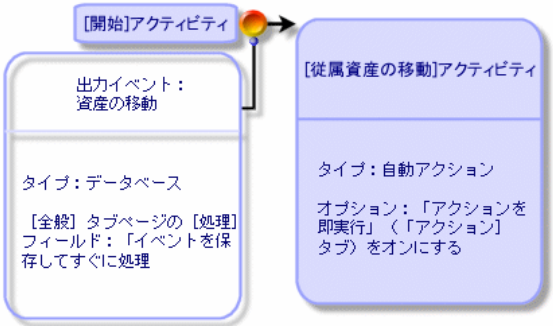
6 トランザクション全体が確認されます。

1つの手順にエラーが発生すると、資産と従属資産とも設置場所は変更されません。

処理が正常に終了すると、すべての設置場所が変更されます。

一方、下に示すように、非同期ワークフロースキームを使って同じ処理を実行すると、エラーが発生した場合は、従属資産の設置場所を変更せずに、資産の設置場所を変更することができます。

図 9.7. 非同期ワークフロースキームの例



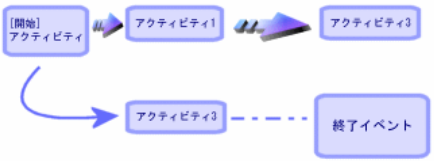
終了イベント

定義

終了イベントが発生すると、実行するタスクが残っている場合もワークフローは終了します。

例：

図 9.8. 終了イベント付きのワークフロースキーム



上記のワークフローが発生し、次の状態になったとします。

- アクティビティ1の出力イベントが発生してアクティビティ2がトリガされ、実行するタスクが作成された。
- アクティビティ3の終了イベントが発生した。

この場合、アクティビティ2のタスクが実行されていない場合も、ワークフローは終了します。

イベントを終了イベントとして指定する

ワークフロースキームの［**アクティビティ**］タブページにあるグラフィカルエディタを使って、ワークフロースキームを作成する場合は、次の方法で終了イベントを指定します。

1. ワークフロースキーム内のイベントをマウスの右ボタンでクリックします。
2. ショートカットメニューから［**終了イベント**］を選択します。

ワークフローの遷移

遷移は、あるアクティビティから別のアクティビティへ出力イベントをリンクします。

イベントは複数の遷移にリンクできます。

必要に応じて、遷移の詳細画面の［**AQL条件**］（SQL名：AQLCond）フィールドで、遷移の起動条件を指定できます。

ワークフローのアラームと制限時間

ワークフローのアクティビティごとに、次の要素を定義できます。

- 実施の期限
- 実施の期限またはデータベースに格納されている日付にリンクするアラーム。このアラームによりアクションが起動します。

警告:

アクティビティの詳細画面の［全般］タブページにある［タスクを保存］（SQL名：bLogWorkItem）オプションをオンにしないと、制限時間とアラームは定義できません。

制限時間

ワークフローのアクティビティ実施の制限時間は、アクティビティの詳細画面の［**制限時間**］タブページで定義します。

注意:

アクティビティの詳細画面の［制限時間］タブページは、同じ画面の［全般］タブページにある［タスクを保存］（SQL名：bLogWorkItem）オプションをオンにしないと表示されません。

この制限時間は、アクティビティがトリガされる時間に基づいて定義されます。

制限時間は、業務用カレンダに関連付けられます。

期間を指定するか、または次の3つの定義済みオプションから選択します。

- ［翌日］
- ［今週末］
- ［今月末］

警告:

期間を入力すると、Asset Managerは業務時間で指定していると認識し、時間単位に変換します。例えば、「2日」と入力すると、48時間と解釈されます。

ワークフローのアラーム

アクティビティの詳細画面の［**アラーム**］タブページで、各アクティビティにアラームを関連付けることができます。

注意:

アクティビティの詳細画面の［アラーム］タブページは、同じ画面の［全般］タブページにある［タスクを保存］（SQL名：bLogWorkItem）オプションをオンにしないと表示されません。

制限時間

アラームをトリガする制限時間は、次のいずれかの方式で定義できます。

- データベースに格納されている日付から経過した期間で指定（［**タスクを開始してから経過した時間**］タイプ）
- データベースに格納されている日付前の期間で指定（［**タスク終了前の時間**］タイプ）
- アクティビティを実行する制限時間に対する割合で指定（［解決時間比］タイプ。［**制限時間**］タブページの［**制限時間**］（SQL名：tsResolDelay）フィールドを使って計算）

注意:

ワークフローの制限時間を定義する期間は、業務日で指定します。

タスクを作成するとすぐに、関連するワークフローのアラームが生成されます。

ワークフローの制限時間は、Asset Manager Automated Process Managerによってモニタされます。モニタの頻度は、Asset Manager Automated Process Managerのオプションで定義します。

アラームの効果

アラームで次のインスタンスをトリガできます。

- Asset Managerで定義したアクション
- イベント。アラームによってトリガされるイベントは、［**アラーム**］タイプのイベントです。名前は、これらのイベントを定義したアラームと同じです。

ワークフローの実行グループ

ワークフローの実行グループを使って、ワークフロースキームを分類できます。ワークフロースキームが属する実行グループは、ワークフロー詳細の［**全般**］タブの［**実行グループ**］（SQL名：GroupName）フィールドで指定します。

Asset Manager Automated Process Managerは、新しいワークフロー実行グループの作成をモニタします。

Asset Manager Automated Process Managerは、新しいワークフロー実行グループGを検出し次第、［**実行グループ 'G'の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。
- 異なるワークフローの実行グループごとに、異なるインスタンスAsset Manager Automated Process Managerでモニタできる。

ワークフローの実行グループが検出されると、Asset Manager Automated Process Managerは対応するワークフローの規則（アラームのモニタ、［**定期的**］タイプのイベントの処理、遷移の起動、タスクの実行など）をモニタし、実行します。

優先度

各グループには、優先度を定義する必要があります。この優先度は、ワークフローグループの実行順序に相当し、一定期間内に起動するワークフローを定義することができます。

優先度レベルは［**優先度**］フィールドで定義します。優先度レベル1は優先度レベル3よりも重要です。

ワークフローのトラッキング

レコードのワークフローインスタンスを表示します。

Asset Managerのあるテーブルを、ワークフロースキームの開始オブジェクトのコンテキストとして定義すると、［**ワークフロー**］タブがそのテーブルの詳細画面に表示されます。

この［**ワークフロー**］タブページには、選択しているレコードを開始オブジェクトとして使う進行中のワークフローのステータスが表示されます。

Windowsクライアント：ワークフローインスタンスのリストをサブタブまたはリストで表示できます。

Webクライアント：リストビューのみが使用できます。

各ワークフローインスタンスは、インスタンスの処理方法を指定します。

- Windowsクライアント：

  発生したイベントは左側に表示されます。

  右側には、ワークフロースキームが表示されます。実行するアクティビティが点滅表示されます。その後の手順は淡色表示されます。

- Webクライアント：

  発生したイベントは上部に表示されます。

  ワークフロースキームは下部に表示されます。実行するアクティビティを表すボックスは**明るい青色**、それ以外のボックスは**暗い青色**です。

ワークフローインスタンスの各手順の操作中の情報の表示

- 各ワークフロースキームボックス上にマウスのアイコンを置くことによって（このように操作するとポップアップ情報が表示される）、ワークフローインスタンスの完了済みの手順または進行中の手順についての情報（タスクが割り当てられた担当者など）を表示できます。
- A アイコンをクリックすると、完了したワークフローのテキスト説明とイベントを表示できます。
- Windowsクライアント：拡大機能を使用してスキームを拡大できます。この機能はWebクライアントでは使用できません。

ワークフロースキームとインスタンスの更新

Windowsクライアント

ワークフロースキームとインスタンスの更新は動的に実行されます。この処理は、ワークフロースキームの処理がある手順から次の手順に移行したとき、またはスキーム自身に対して変更を行ったときに自動的に実行されます。

Webクライアント

次の2種類の情報があります。

- ワークフローインスタンスに関連する情報。イベントの完了済み、進行中ステータス（色分けされた矢印）各アクティビティの完了日などの情報。この情報は動的に更新されます。変更されると、ワークフローインスタンスのグラフィカル表示は即座に更新されます（［**F5**］キーを押して画面を更新してください）。
- ワークフロースキームに関連する情報。ボックスとリンクのリストや位置などの情報。この情報は、Asset Manager Web Tierサーバ上のメモリに格納された**キャッシュ**で管理され、定期的に（デフォルトで**24時間**に1回）更新されます。キャッシュに格納されている各アイテムは個別に管理されます。
  1. ワークフロースキームを最初に開いたときに、ワークフロースキームはキャッシュに格納され、格納日が記録されます。
  2. 次にこのスキームを開いたときに、Asset Manager Web Tierは現在の日付と、キャッシュ内のワークフロースキームの最終格納日との差をチェックします。
     - この時間の差が、事前に定義したキャッシュの更新間隔より小さい場合、スキームの画像は直接キャッシュから取得されます。これにより、情報の表示に必要な時間が短縮されます。
     - 反対に、この時間の差が事前に定義したキャッシュの更新間隔より大きい場合は、ワークフロースキームの情報は更新され、新しい格納日がキャッシュに保存されます。

キャッシュの更新間隔を変更する方法については、**『Tailoring』**ガイドの**「Customizing Web clients」**セクションの**「Modifying the Web client's default behavior」**の章の「**To modify how often the cache managing the workflow schemes and instances is refreshed**」セクションを参照してください。

終了済みワークフローインスタンスの削除

終了したワークフローインスタンスを削除する理由

ワークフロースキームを実行するとワークフローインスタンスが作成されます（［**ワークフローインスタンス**］（amWfInstance）テーブル）。

これらのワークフローインスタンスは、実行終了後（［**ステータス**］（seStatus）フィールド）でも自動的には削除されません。

ワークフロースキームを頻繁に使用する場合、作成されるワークフローインスタンスの数は莫大になります。

これはデータベースのサイズを無用に増大させるだけでなく、Asset Managerの性能を低下させる原因にもなります。

このため、実行が終了したワークフローインスタンスを、定期的に削除するように設定することをお勧めします。

終了済みワークフローインスタンスの自動削除

本番用データベースで、終了済みワークフローインスタンスの削除を自動化するには、

1. ［**ワークフロースキーム**］（amWfScheme）テーブルにフィールドを1つ追加し、期限を指定できるようにします。これを使って、ワークフロースキームの古いインスタンスが、一定の期限を経過した後に削除されるように指定します。
2. 古いワークフローインスタンスを削除するアクションを作成します。
3. アクションを自動的に実行するワークフロースキームを作成します。
4. ワークフロースキームの実行を自動化するために、Asset Manager Automated Process Managerを設定します。

［ワークフロースキーム］テーブルにフィールドを 1 つ追加する

［**ワークフロースキーム**］（amWfScheme）テーブルに次のフィールドを追加します。

パラメータ	値
SQL名	AutoCleaningDelay
ラベル	インスタンス削除の期限
説明	この期限を過ぎると、終了済みのワークフローインスタンスは削除される。
データ型	期間
このフィールドのインデックスを作成	このオプションは選択しません。

パラメータ	値
説明（［ヘルプの本文］タブ）	<LI>この期限を過ぎると、終了したワークフローインスタンスは削除されます。
例	<LI>-1：ワークフローインスタンスは決して削除されません。 <LI>正の値またはNULL値：期限が過ぎるとワークフローインスタンスは削除されます。

既存のテーブルにフィールドを追加する方法については、『**Tailoring**』ガイドの「**Customizing the database**」の章、「**Creating new objects**」セクションの「**Creating a field, link or index**」を参照してください。

［インスタンス削除の期限］フィールドに入力する

使用する各ワークフロースキームの、［**インスタンス削除の期限**］（AutoCleaningDelay）フィールドに値を入力します。

本番用データベースに接続する

1 Asset Managerを起動します。
2 本番データベースに接続します。

古いインスタンスを削除するアクションを作成する

1 アクションのテーブルを開きます（［**ツール／アクション／編集**］メニュー）。
2 ［**新規作成**］をクリックします。
3 次のフィールドに値を入力します。

フィールド名	SQL名	値
名前	Name	終了済みワークフローインスタンスを削除する
コンテキスト	ContextTable	ワークフロースキーム（amWfScheme）
タイプ	seActionType	スクリプト
SQL名	SQLName	DeleteFinishedWfInstances
アクションのスクリプト	Script	以下（*）参照

（*）アクションのスクリプト

```
Const NumberOfInstanceToDelete = 50

Dim lRc As Long
Dim i As Long

i = 0

If [AutoCleaningDelay] >= 0 Then
Dim hqWfInstance As Long
hqWfInstance = AmQueryCreate()
```

```
lRc = AmQueryExec(hqWfInstance, "SELECT lWfInstanceId FROM amW
fInstance WHERE lWfSchId = "& [lWfSchId] & " And seStatus = 1 AND A
DDSECONDS(dtCompleted, " & [AutoCleaningDelay] & ") < GetDate()" )

Do While (lRc = 0 And i < NumberOfInstanceToDelete)
Dim hrWfInstance As Long
hrWfInstance = AmGetRecordHandle(hqWfInstance)
lRc = AmDeleteRecord(hrWfInstance)
lRc = AmReleaseHandle(hrWfInstance)
lRc = AmQueryNext(hqWfInstance)
i = i + 1
Loop
End If
```

4 ［**作成**］をクリックします。

アクションを自動的に実行するワークフロースキームを作成する

1 ワークフロースキームのリストを表示します（［**ツール／ワークフロー／ワークフロースキーム**］メニュー）。
2 ［**新規作成**］をクリックします。
3 次のフィールドに値を入力します。

フィールド名	SQL名	値
名前	Name	終了済みワークフローインスタンスの削除
リファレンス	Ref	ADM_CLEAN_WF_INSTC
開始オブジェクトのコンテキスト	StartContextTable	ワークフロースキーム（amWfScheme）
実行グループ	GroupName	Asset Manager Automated Process Manager内でワークフロースキームの実行を自動化できるように、名前を記入します（**ADMIN**など）。
１つのオブジェクトにつき1つのワークフローを起動	bUniqueActive	このオプションを選択します。
データベースにインスタンスを保存しない	bTransient	このオプションは選択しません。

4 ［**作成**］をクリックします。
5 ［**アクティビティ**］タブを選択します。
6 「**開始**」アクティビティで右クリックし、［**イベントの追加**］メニューを選択します。
7 次のフィールドに値を入力します。

フィールド名	SQL名	値
名前	Name	Timer
タイプ	seMonitoringType	定期的
進行中のワークフローがある場合にワークフローを再度初期化	定期的	このオプションは選択しません。
AQL条件	AQLCond	AutoCleaningDelay >= 0

8 必要に応じて［**パラメータ**］タブにデータを入力します。
9 ［**追加**］をクリックします。
10 右クリックし、［**アクティビティの追加**］メニューを選択します。
11 次のフィールドに値を入力します。

フィールド名	SQL名	値
名前	Name	Clean W/F instances
タイプ	seType	自動アクション
タスクを保存	bLogWorkItem	このオプションを選択します。
コンテキスト	ContextTable	ワークフロースキーム（amWfScheme）
入力条件	seInCond	OR

12 ［**追加**］をクリックします。
13 ［**パラメータ**］タブを選択します。
14 次のフィールドに値を入力します。

フィールド名	SQL名	値
アクションを即実行	bExecImmediately	このオプションを選択します。
アクション	Actions	終了済みワークフローインスタンスを削除する

15 ［**閉じる**］をクリックします。
16 ［**Clean W/F instances**］アクティビティをダブルクリックします。
17 ［**実行済**］イベントを右クリックし、［**イベントの詳細**］メニューを選択します。
18 次のフィールドに値を入力します。

フィールド名	SQL名	値
処理	seProcessingMode	イベントを保存してすぐに処理

19 ［**閉じる**］をクリックします。
20 マウスを使って、［**開始**］アクティビティと［**Clean W/F instances**］アクティビティ間にリンクを作成します。

21 ［**実行済**］イベントを右クリックし、［**終了イベント**］メニューを選択します。

22 ［**変更**］をクリックします。

ワークフロースキームの実行が自動化されるように、**Asset Manager Automated Process Manager**を設定する

1 Asset Manager Automated Process Managerを起動します。

2 「**終了済みワークフローインスタンスの削除**」ワークフローの実行をトリガするモジュールを設定します（［**ツール／モジュールの設定**］メニュー）。

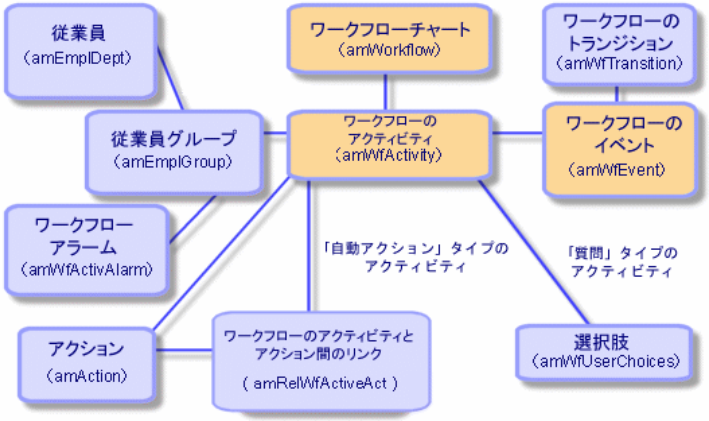
モジュールには「**実行グループ 'X' にワークフロー規則を適用**」という名前が付いています。「**X**」は、ワークフロースキームの［**実行グループ**］（GroupName）フィールドで指定した値になります。

3 ワークフローを自動的に実行するには、Asset Manager Automated Process Managerを稼動状態にしておきます。

技術情報：データモデル

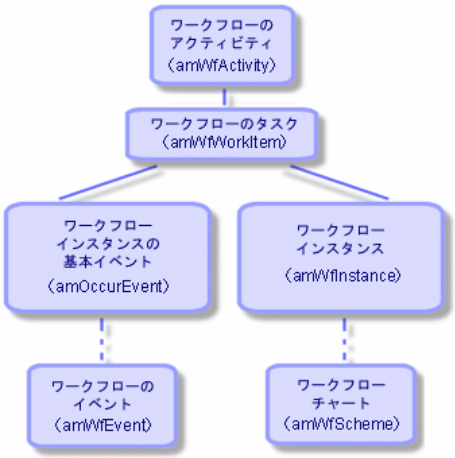
ワークフローに関連するメインテーブルとこれらの関係は、次の図のとおりです。テーブルは、ラベルとSQL名で表記されています。

図 9.9. ワークフロースキームを定義できるメインテーブル



進行中のワークフローに関連するメインテーブルは、次の図のとおりです。

図 9.10. 実行中のワークフローに関連するメインテーブル



10 データの出力およびSQLビューの作成

この章では、Asset Managerデータの出力方法とデータベースのSQLビューの管理方法について説明します。

定義

出力スクリプト

エクスポートスクリプトを使うと、データのエクスポート、Asset Manager Export Toolまたは**amexp.exe**を使ったSQLビューの作成、再作成、または削除を行うことができます。エクスポートスクリプトを保存すると、後で繰り返し使用できます。

出力スクリプトは次の要素で構成されます。

- 出力クエリ。データベースから抽出するデータを定義します。
- データを出力するときの書式設定オプション
- SQLビューの作成および削除で実行するアクション

出力スクリプトは次のモードで動作します。

- データを出力する場合は「出力モード」
- データベースからSQLビューを作成（再作成）または削除する場合は「ビューモード」

出力クエリ

エクスポートクエリは、Asset Manager Export Toolを使って定義します。

出力クエリは、次の要素を使って定義します。

- 名前
- 生成される出力ファイル（「出力モード使用時」）
- コメント（出力されません）
- 開始テーブル
- 抽出する列のリスト（開始テーブルからのフィールド、リンク、任意管理項目、および特殊フィールド）および関連付けられている並べ替え基準
- WHERE句を含む、抽出条件を定義するフィルタ
- HAVING句を含む、抽出条件を定義するフィルタ
- クエリタブページ（［**フィルタ（WHERE句）**］およびHAVING句の各タブページ）のテキスト
- プレビュータブページ

Asset Managerデータベースからデータを出力する

Asset Managerデータベースからテキストファイルに、次の方法でデータを出力できます。

- 出力スクリプトを使う。
- ［**ユーティリティ／リストの出力**］ショートカットメニューから。このメニューは、少なくとも1つのリストまたはタブページ内のリストが表示されている場合に表示されます。アクティブなリストを出力できます。

出力スクリプトを使ってデータを出力する

1. ［**スタート**］メニューから、またはAsset Managerプログラムグループから Asset Manager Export Toolを起動します。
2. 「出力モード」に設定した出力スクリプトを定義します。
   1. ［**クエリ**］タブページで、出力するデータを定義するクエリを作成します。
   2. ［**書式設定**］タブページで、データを出力するテキストファイルの書式を指定します。
   3. ［**ファイル／保存**］または［**ファイル／名前を付けて保存**］メニューを使って、スクリプトを保存します。
3. 次のいずれかの方法で出力スクリプトを実行します。
   - Asset Manager Export Toolで<スクリプトの実行> アイコンから直接実行する。

- コマンドプロンプトから**amexpl.exe**を使う。

注意:

Asset Managerで定義したアクセス制限の整合性を維持する理由から、Asset Manager Export Toolを起動またはamexpl.exe**を実行できるのは管理者（「Admin」ログインまたは管理者権限を持つユーザ）だけです。**

［リストの出力］状況依存メニューを使ってデータを出力する

［リストの出力］状況依存メニューにはすべてのAsset Managerユーザがアクセスできます。ユーザは、自分の権限に応じて表示できるデータのみを出力できます。

1 出力するリスト（メインリストまたはタブページ内のリスト）を表示します。画面に複数のリストを表示している場合は、カーソルが出力するリスト内にあることを確認してください。
2 **［リストの出力］**状況依存メニューを選択します。
3 表示されるウィンドウに入力し、**［出力］**ボタンをクリックします。

注意:

［ファイル／リストのエクスポート］メニューの詳細については、『はじめに』ガイドの「Asset Managerを初めて使用する」の章、「レコードリスト」のセクションの「リストをエクスポートする」を参照してください。

Asset ManagerデータベースでSQLビューを管理する

Asset Manager Export Toolを使うと、Asset ManagerデータベースでSQLビューを作成、再作成、または削除できます。その後、外部ツールでテキストファイルの代わりにこれらのビューを使用できます。

注意:

注意：出力スクリプトで作成、変更、および削除できるSQLビューは、Asset Managerで使われている意味でのビューとは異なります。SQLビューは、SQLの「CREATE VIEW」ステートメントに当たります。

Asset ManagerデータベースでSQLビューを作成、再作成、または削除するには：

1 Asset Manager Export Toolを起動します。
2 「ビューモード」に設定した出力スクリプトを定義します。

1 ［**クエリ**］タブページで、抽出するデータを定義するクエリを作成します。
2 ［**ビュー**］タブページで、実行するアクションを指定します。ビューの作成、変更または削除、生成されるSQLスクリプトの直接実行、ファイルへの保存などのアクションを指定できます。
3 出力スクリプトを保存します。

3 次のいずれかの方法で出力スクリプトを実行します。
- Asset Manager Export Toolで直接実行する。
- **amexpl.exe**を実行する。

推奨事項

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

出力スクリプトを定義する

データのエクスポートまたはデータベースのSQLビューの作成を行うには、エクスポートスクリプトとその中で使うエクスポートクエリを定義する必要があります。これを行うには、Asset Manager Export Toolを使います。

出力スクリプトは次の要素で構成されます。
- 出力クエリ。データベースから抽出するデータを定義します。
- データを出力するときの書式設定オプション

SQLビューの作成および削除で実行するアクション

出力スクリプトは次のモードで動作します。
- データを出力する場合は「出力モード」
- データベースからSQLビューを作成（再作成）または削除する場合は「ビューモード」

ここでは、出力スクリプトの作成方法について説明します。
- 出力スクリプトの作成方法
- 出力クエリを定義する
- 出力スクリプトの出力書式
- SQLビューに関するアクション

出力スクリプトの作成方法

出力スクリプトを作成または変更するには：

1 Asset Manager Export Toolを起動します。
2 適切なデータベースを開きます。注意：「Admin」ログインまたは管理者権限を持つログインを使った場合のみ接続できます。
3 ［**ファイル／新規作成**］メニューを使って新しいスクリプトを作成するか、［**ファイル／スクリプトを開く**］メニューを使って変更するスクリプトを開きます。
4 Asset Manager Export Tool画面の上部で、データを出力するか（エクスポートモード）、データベースのSQLビューを管理するか（ビューモード）を定義します。
5 ［**クエリ**］タブページで出力スクリプトのクエリを作成します。
6 データを出力する場合は、［**書式設定**］タブページで出力データの書式を指定します。
7 SQLビューを管理する場合は、［**ビュー**］タブページで実行する操作を指定します。
8 ［**ファイル／保存**］メニューまたは［**ファイル／名前を付けて保存**］を使ってスクリプトを保存します。

出力クエリを定義する

エクスポートスクリプトのクエリは、Asset Manager Export Toolの［**クエリ**］タブページで定義できます。

- 出力クエリを追加する場合は、［**新規作成**］ボタンをクリックします。
- 選択した出力クエリを削除するには、［**削除**］ボタンをクリックします。

出力スクリプトでクエリを作成する

1 ［**クエリ**］タブページで［**新規作成**］をクリックします。
2 クエリ名を定義します。この名前は、出力スクリプトの詳細の［**メッセージ**］タブページの実行ログで使われます。
3 コメントを入力することは可能ですが、出力されません。
4 ［**クエリ**］フィールドで、出力するデータをスクリプトの書式で定義します。
5 ビューの作成、変更または削除ではなく、データーを出力する場合は、［**ファイル**］フィールドに、クエリによって選択された出力データを書き込むテキストファイルのパスと名前を指定します。出力スクリプトに複数の出力クエリが含まれている場合は、複数のテキストファイルが作成されます。

注意:

［SQLビューの作成／破棄］オプションを選択した場合は、［ファイル］フィールドは表示されません。

出力するデータ

出力するデータを指定するには、［クエリ］タブページの［**クエリ**］フィールドに入力します。クエリは、Asset Managerデータベースのテーブルに適用されます。

クエリは、直接入力するか、ボタンをクリックしてクエリを簡単に定義できるウィンドウにアクセスし、次のオプションを使って指定することもできます。

出力する列と並び順

［**列／並べ替え**］タブ(クエリの詳細)では、出力するフィールド、リンク、任意管理項目、および特殊フィールドのリストと、その並べ替え基準を定義します。

左側のリストから出力に使うすべてのフィールド、リンク、任意管理項目、および特殊フィールドを1つずつ選択し、右矢印ボタンをクリックして、右側のリストに挿入します。

右側のリストの各列で次の操作を行います。

- 列を出力するには、［**表示／非表示**］チェックボックスをオンにします。［**表示／非表示**］チェックボックスをオンにしないと、列は出力されません（出力データの並べ替えなどに列を使用することは可能です）。
- ［**グループ化**］ボックスをチェックして、列に対応するフィールドでデータをグループ化します。これは、SQLクエリにGROUP BY <フィールド名>句を追加するのと同じです。

  例：

  ```
  SELECT Brand, Count(lModelId) FROM amModel GROUP BY Brand ORDER BY Brand
  ```

  警告:

  ［グループ化］チェックボックスをオンにすると、クエリにはGROUP BYが追加されますが、クエリを有効にするには、SELECT句に適切な集合関数を追加する必要があります。

出力データの並び順を定義します。

1. インデックスで並べ替えるには、［**インデックスで並べ替え**］フィールドでインデックスを選択します。
2. それ以外の場合は、並べ替える順番で項目の［**並べ替え**］チェックボックスをオンにします。

注意:

クエリで指定されたインデックスを強制的に使用するためには、［インデックスの強制使用］オプションを選択します。

詳細については、このガイドの「AQLクエリ」の章、「並べ替えとインデックス」のセクションを参照してください。

［**固有のレコードのみ**］チェックボックスをオンにすると、まったく同じ行が複数ある場合は1度だけ出力されます。これは、SQLクエリにDISTINCT句を追加するのと同じです。

［**固有のレコードのみ**］チェックボックスをオフにした場合のクエリ

```
SELECT Brand FROM amModel
```

［**固有のレコードのみ**］チェックボックスをオンにした場合のクエリ

```
SELECT DISTINCT Brand FROM amModel
```

フィルタ

次の2種類のフィルタを使って、抽出するデータを選択できます。

- ［**フィルタ（WHERE句）**］タブページでWHERE句を使うAQLクエリ
- ［**HAVING句**］タブページでHAVING句を使うAQLクエリ

クエリを表示する

［**列／並べ替え**］、［**フィルタ（WHERE句）**］、および［**HAVING句**］タブページで定義したAQLクエリは、［**クエリ**］タブページに表示されます。

クエリ結果をプレビューする

クエリをテストし、その結果を［**プレビュー**］タブページにSQL言語のシンタックスで表示できます。

をクリックするだけで、クエリ結果をレコードのリストとしてプレビューできます。ウィンドウの右下に、クエリに一致するレコード数が表示されます。

出力スクリプトの出力書式

［**データの出力**］を選択した場合は、［**書式設定**］タブページで、出力テキストファイルの書式を定義できます。この書式設定はすべての出力クエリに適用されます。

注意:

ビューの削除、作成、再作成を選択した場合は、［書式設定］タブページは表示されません。

列タイトル

出力ファイルの1行目に含める要素を次の中から選択します。

- 出力クエリで指定した列のエイリアス
- 列に対応するフィールドまたはリンクの「SQL名」
- 列に対応するフィールドまたはリンクの「説明」

列の区切り

各列のデータ間に挿入する記号を指定します。

テキスト識別子

テキスト文字列を囲む識別子を指定します。一重引用符（'）を使うと、二重引用符（"）として出力されます。逆に二重引用符（"）は、一重引用符（'）として出力されます。

文字セット

ANSI、OEM(DOS)、UFT-8、UNICODE と Latin1文字セットのどれかを選択できます。

十進法の区切り

出力された数字の小数部を分離する文字を指定します。

日付の区切り

出力された日付の日、月、および年の間に挿入する文字を指定します。

日付の表示形式

出力される日（DD）、月（MM）、および年（YY）の順序を定義します。

西暦年の桁数

西暦年を2桁と4桁のどちらで出力するかを定義します。

時刻の区切り

時間、分、および秒の間に挿入する文字を指定します。

秒数表示

出力された時刻で秒を表示するかどうかを指定します。

SQLビューに関するアクション

出力クエリに対応するSQLビューを削除または（再）作成する場合は、［ビュー］タブページを使って、実行するアクションを定義できます。

注意:

［ビュー］タブページは、［データの出力］オプションを選択した場合は表示されません。

［アクション］オプションで、実行するアクションとして次のどちらかを選択します。

- ビューの作成または変更
- ビューの作成

「SQLビューの操作スクリプト」セクションで、クエリの処理方法を指定します（［**クエリ**］タブページ、**「アクション」**フレーム）。

- 出力スクリプトの実行時に直接SQLビューを作成または再作成するには、［**直接SQLを実行**］オプションを選択します。
- ビューを作成（CREATE VIEWステートメント）または削除（DROP VIEWステートメント）するSQLビュースクリプトを作成するには、［**SQLコードをファイルに保存**］オプションを選択してから、次の操作を実行します。
  1. ボタンをクリックして、ファイルの名前とパスを指定します。
  2. セミコロン（;）（Oracleの場合）と「GO」（他のすべてのDBMS）のどちらかをSQLステートメントの区切り文字として指定します。

出力スクリプトを実行する

出力スクリプトを使うと、データの出力やSQLビューの管理を行うことができます。

ここでは、出力スクリプトを実行する2つの方法について説明します。

- Asset Manager Export Toolからエクスポートスクリプトを実行する
- DOSから出力スクリプトを実行する

Asset Manager Export Toolからエクスポートスクリプトを実行する

Asset Manager Export Toolからエクスポートスクリプトを実行するには：

1. Asset Manager Export Toolを起動します。
2. 出力スクリプトを定義し、保存します。
3. 次のいずれかの方法でスクリプトを実行します。
   - ［**アクション／スクリプトの実行**］メニューを使う。
   - ［F8］キーを押す。
   - をクリックする。

出力処理の進行状況は、［**メッセージ**］タブページに表示されます。

出力処理が正常に終了すると、最後に「スクリプトが問題なく実行されました。」というメッセージが表示されます。エラーが発生すると、「スクリプトを実行中にエラーが発生しました。」というメッセージが表示されます。

すべてのメッセージの前には、次のアイコンが表示されます。

一般情報

エラー

問題なく出力

⚠ 注意

DOSから出力スクリプトを実行する

動作

DOSソフトウェアをオンラインで実行するには、Asset Manager Export Toolを使って、事前に実行スクリプトを作成しておく必要があります。

その後、Asset Managerのインストール先フォルダの**bin**サブフォルダにある**amexp.exe**プログラムを使って、手動または自動(バッチファイルなど)で出力コマンドを実行できます。

シンタックス

```
amexpl [-verbose] [-?|h|H] -script:<スクリプト名>
-cnx:<cnx> [-login:<ログイン名>]
[-password:<パスワード>]
```

-verbose：出力処理中にメッセージを表示します。

-?、-hまたは-H：ヘルプメッセージを表示します。

-script：実行する出力ファイルのパスと名前。

-cnx：Asset Managerデータベースへの接続名（［**ファイル／接続の管理**］メニューに表示される名前）。

-login：データベース管理者のログイン名（「Admin」または管理者権限を持つユーザ）。

-password：ログインのパスワード

<>内に入れる文字列にスペースを入れることはできません。

例：

```
amexpl32 -verbose -script:ibmassets.scx -cnx:GeneralDatabse -login:Gerald -p
assword:PAssword
```

11 スクリプト

この章では、スクリプトの使用方法について説明します。

スクリプトの定義

概要

「スクリプト」という用語は、一般に高級言語で書かれたプログラムを指します。Asset Managerでは、次の3種類のスクリプトを使います。

- プロシージャスクリプト
  - フィールド値の計算、Asset Managerのデータベースオブジェクトのプロパティの条件付けなどに使われる、Basicで書かれた計算スクリプト
  - 特にアクションでのタスクを実行するBasicスクリプト

  注意:

  これらのBasicプログラムには関数を組み込むことができます。ここでは主にこのタイプのスクリプトについて説明します。

- 宣言スクリプト：Basicとは異なる独自のスクリプト言語を使うインポートおよび出力スクリプトです。このタイプのスクリプトについては、『**管理**』ガイドの「**インポート**」の章、および「**データの出力およびSQLビューの作成**」の章を参照してください。

- 宣言スクリプトとプロシージャスクリプトの混合：このタイプのスクリプトは、Asset Managerのウィザードで使われます。

このバージョンのBasicに関する情報

Asset Managerで使われているBasicのバージョンは、Cypressによって開発されたサブセットで、Visual Basic for ApplicationsTMと互換性があります。この言語、構造、およびシンタックスの詳細については、Basicのドキュメントを参照してください。

特定の**Visual Basic for Applications**関数だけをサポートしています。

- ファイルアクセス関数はサポートしていません。
- 日付および時刻関数のサポートに制限があります。
- **Visual Basic for Applications**コントロールは使用できません。

注意:

関数やBasicキーワードのプログラミングの参照を表示するには、調べる語の上にカーソルを置き［F1］キーを押すと、状況依存ヘルプが表示されます。

データアクセス表記

Asset Managerで使われているBasicシンタックスは、現在のレコードからのデータアクセス関数を除き、標準のシンタックスと似ています。データアクセスでは、次の形式を使います。

```
[Link.Link.Field]
```

例：モデルのテーブルから

```
[Category.FullName]
```

注意:

次のシンタックスを使って、リンクのID番号を取得できます。

```
[Link.Link]
```

リンクを参照する場合は、リンクのSQL名またはリンクのキー名を使います。

例：

```
RetVal=[Contact.Location] or RetVal=[Contact.lLocaId]
```

両方の例とも、同じ結果、つまりリンクのIDが返されます。

スクリプトの適用

Asset Managerでは、次のプロパティにBasicスクリプトを関連付けることができます。

- フィールドのデフォルト値の設定（ショートカットメニューの［**オブジェクトの設定**］コマンドを使って設定）
- テーブルに関連付けられた任意管理項目のデフォルト値
- Basicタイプの特殊フィールド
- フィールドの設定（ショートカットメニューの［**オブジェクトの設定**］コマンドまたはAsset Manager Application Designerを使って設定）
  - **デフォルト値**
  - **必須属性**
  - **履歴**
  - **読取り専用**
- テーブルに関連付けられている任意管理項目のパラメータ
  - ［**デフォルト値**］（SQL名：DefValScript）
  - ［**使用可能**］（SQL名：seAvailable）
  - ［**デフォルトで表示**］（SQL名：seForceDisplay）
  - ［**必須**］（SQL名：seMandatory）
  - ［**履歴の保持**］（SQL名：seKeepHistory）
- ［**スクリプト**］アクション
  - ［**スクリプト**］アクションの［**アクションのスクリプト**］（SQL名：Script）
- ウィザード
  - ウィザードスクリプトの開始および終了
  - ノードプロパティを定義するスクリプト
- Basicタイプの特殊フィールド
- ワークフロー
  - ［**テスト／スクリプト**］ワークフローのアクティビティ
  - ［**データベース**］ワークフローのイベント
  - 「**計算された**」ワークフローの担当者

関数の概要

ここでは、次の内容について説明します。

- 関数の定義

- 組み込み関数とプログラマブル関数
- 関数とパラメータのデータ型

関数の定義

「関数」とは、なんらかの処理を実行し、値をユーザに返すプログラムです。返される値を「戻り値」または「戻りコード」といいます。

関数は次のような構造になっています。

```
Function <関数名> (<パラメータ> As <パラメータの種類>[, ..., <パラメータ> As <パラメータの種類>]) As <関数の種類>

<関数によって実行されるプログラム（スクリプト）。このプログラムで戻り値を定義します>

End Function

End Function
```

この構造は、組み込み関数とプログラマブル関数の両方に当てはまります。

組み込み関数とプログラマブル関数

Asset Managerでは、主に組み込み関数とプログラマブル関数を使うことができます。

組み込み関数

組み込み関数は、ユーザ用に事前に作成されたソフトウェアアイテムに似ています。これらのソフトウェアアイテムは、すべてのタイプのタスク（ユーザから要求されるデータの計算、変換）を実行し、結果を返します。ユーザは、名前を使って関数を呼び出し、結果を得るために必要な情報を指定します。ユーザが指定する情報項目を「パラメータ」と呼びます。

例えば、**AmConvertCurrency()**関数は、特定の日付に定義された為替レートを使って、通貨Aの金額を通貨Bの金額に換算します。この場合は、次のような情報が必要になります。

- 関数名は**AmConvertCurrency**
- ユーザが指定するパラメータ
  - 通貨A
  - 通貨B
  - 換算する金額
  - 換算を実行する日付（使用する為替レートの識別に使います）

この関数は換算を実行し、換算結果に対応する戻り値をユーザに返します。

プログラマブル関数

プログラマブル関数は、ユーザ自身が作成できるソフトウェアアイテムです。ユーザは、プログラマブル関数によって**RetVal**変数（「戻り値」とも呼ばれる）に戻される値を、次の形式で明示的に定義する必要があります。

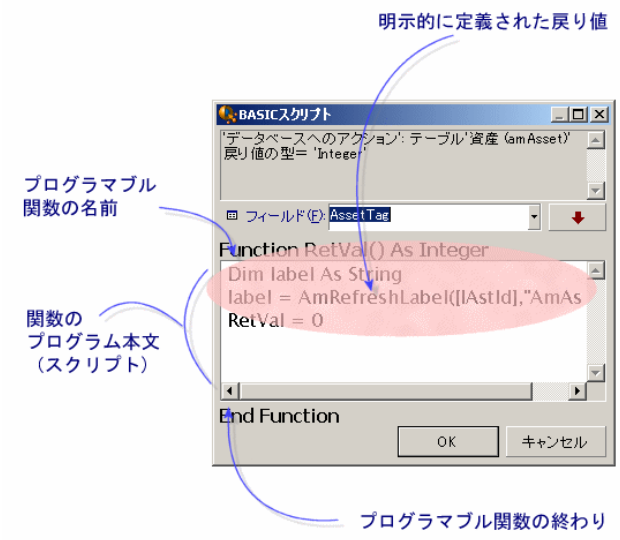
```
RetVal=<式>
```

注意:

Asset Managerでは、戻り値が定義されていない関数のスクリプトはコンパイルできません。

プログラマブル関数には、スクリプトビルダからアクセスできます（スクリプトを作成できるフィールドの ボタンをクリックすると表示されます）。スクリプトビルダは、関数に対応するソフトウェアアイテムを、ユーザが簡単に作成できるように設計されています。スクリプトビルダには、プログラマブル関数を作成するためのテンプレートがあります。

図 11.1. スクリプトビルダ



プログラマブル関数の説明は、スクリプトビルダウィンドウの上部に表示されます。関数に関連するオブジェクト（例えば、資産テーブルの［**バーコード**］（SQL名：BarCode）フィールドのデフォルト値）と、予測される戻りコード（前の例では「文字列」）を確認できます。

関数とパラメータのデータ型

関数のデータ型

組み込み関数のデータ型は、関数から戻される値のデータ型になります。Basicスクリプトでは、コンパイルエラーや実行エラーの原因となることがあるので、この点によく注意してください。

例えば、あるデータ型のフィールドのデフォルト値を定義しているときに、別のデータ型の値を戻す関数は使用できません。例えば、下のデフォルトスクリプトに「日付」または「日付＋時刻」型のフィールドを割り当てようとしたとします。

```
RetVal=AmLoginName()
```

AmLoginName()関数は接続ユーザ名を文字列で返します（文字列型）。そのため、この場合の戻り値の形式は、「日付」フィールドの形式とは互換性がないので、次に同じテーブルでレコードを作成すると、エラーメッセージが表示されます。

パラメータのデータ型

組込み関数で使われるパラメータにもデータ型があります。実行する関数に適したデータ型を使う必要があります。パラメータのデータ型にエラーがある場合は、関数の実行時にエラーメッセージが表示されます。

使用可能なデータ型

関数とパラメータで使えるデータ型とその説明は、次の表のとおりです。

表 11.1. 使用可能なデータ型

タイプ	説明
Integer	-32,768から+32,767の整数
Long	-2,147,483,647から+2,147,483,646の整数
Double	8バイト浮動小数点数
String	あらゆる文字を含むテキスト
日付	日付または日付+時刻
Variant	汎用の任意の型

プログラマブル関数の戻り値のデータ型を決定する

スクリプトを編集する前に、使用する関数とそのデータ型を決定する必要があります。この情報は、すべての「BASICスクリプト」ウィンドウに次の形式で太字で表示されます。

```
Function <関数名>() As <関数のデータ型>
```

最も一般的な関数のデータ型は、「ブール」、「整数」、および「文字列」です。

- 「ブール」関数は、TRUE（真）またはFALSE（偽）を返します。他の値は、Basicスクリプトのコンパイル時にエラーの原因となります。
- 「整数」関数は、整数値だけを返します（0、1、8、12など）。
- 「文字列」関数は、引用符で囲まれた文字列だけを返します（"Building21"など）。

注意:

関数のデータ型に従わない場合は、Basicプログラムのコンパイル中にエラーが発生します。常に使用する関数のデータ型に注意してください。

関数名とデータ型を使って、スクリプトで使う戻りコードを次の形式で指定できます。

```
RetVal=<関数のデータ型に従った式>
```

Basic関数を分類する

スクリプトで使われているBasicでは、様々な種類の関数が使われます。

- Visual Basic for Applications TM標準に従った従来のBasic関数
- Asset Managerに固有の汎用関数。スクリプトを使う機能で使用できます。
- 専用関数。Asset Managerの特定の機能で使用できます。

スクリプト作成の基本

ここでは、スクリプトの機能について、事例を紹介しながら説明します。

- スクリプト例の内容
- 手順1：任意管理項目「チュートリアル」を作成する
- 手順2：編集ウィンドウを開く
- 手順3：アルゴリズムを分析および定義する
- 手順4：Basicスクリプトを作成する
- 手順5：Basicスクリプトをテストする

スクリプト例の内容

目的

「チュートリアル」任意管理項目が「コンピュータ/マザーボード/」モデルだけで使用できることを確認します。

方法

「チュートリアル」任意管理項目の［**使用可能**］（SQL名：seAvailable）パラメータにBasicスクリプトをリンクします。

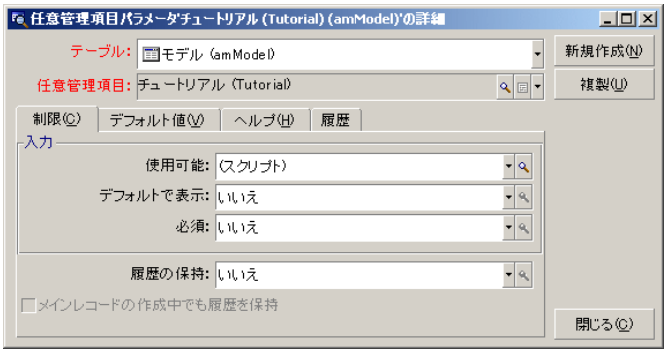
手順1：任意管理項目「チュートリアル」を作成する

［**管理／任意管理項目**］メニューを選択します。［**新規作成**］をクリックして、新しい任意管理項目を作成します。この任意管理項目に次のように入力します。

フィールド名	フィールドの値
［**タイトル**］（SQL：TextLabel）	チュートリアル
［**SQL名**］（SQL名：SQLName）	チュートリアル
［**入力タイプ**］（SQL名：seDataType）	［**数値**］

［**作成**］をクリックして、任意管理項目を作成します。

［**パラメータ**］タブページに移動し、⊞をクリックして、「チュートリアル」任意管理項目のパラメータを編集します。［**制限**］タブページに次のように入力します。



手順2：編集ウィンドウを開く

［制限］タブページ、［使用可能］（SQL名：seAvailable）パラメータを「(Script)」に設定します。 ボタンをクリックします。スクリプト編集ウィンドウが開きます。

手順3：アルゴリズムを分析および定義する

次のタスクを実行するアルゴリズムを作成する必要があります。

- モデルが「/コンピュータ/マザーボード」または従属モデルの場合に、［使用可能］（SQL名：seAvailable）フィールドを［はい］に設定する。
- それ以外の場合はすべて、［使用可能］フィールドを［いいえ］に設定する。

アルゴリズムは次のようになります。

```
モデルのフルネームが "/コンピュータ/マザーボード/"で始まる場合、
任意管理項目が使用できます。
それ以外の場合、
任意管理項目は使用できません。
```

つまり、任意管理項目の［使用可能］フィールドの値は、モデルのテーブルの［完全名］（SQL名：FullName）フィールドの値によって決まります。アルゴリズムでは、このフィールドしか使いません。

［使用可能］フィールドの横の ボタンをクリックして、Basicスクリプトの編集を開始します。編集ウィンドウのドロップダウンリストから、モデルのテーブルから［完全名］（SQL名：FullName）フィールドを選択できます。

フィールドを選択したら、 ボタンをクリックして、編集ウィンドウに値を移します。

手順4：Basicスクリプトを作成する

Basicスクリプト作成用のウィンドウで、スクリプトを作成します。このスクリプトは、手順3で作成したBasicアルゴリズムを変換したものです。

```
If Left([FullName], Len("/コンピュータ/マザーボード/"))="/コンピュータ/マザーボード/" Then
RetVal=1
Else
RetVal=0
End If
```

注意:

スクリプトでは、大文字と小文字は区別されません。

［**OK**］をクリックして、スクリプトの作成を確定します。

手順5：Basicスクリプトをテストする

この手順では、スクリプトが正常に機能することを確認します。

1 ［**ポートフォリオ／モデル**］メニューを選択して、モデルのテーブルを開きます。［**新規作成**］をクリックして、新しいモデルを作成します。
2 必須フィールドだけに入力します。
   1 ［**名前**］
   2 ［**親モデル**］（SQL名：Parent）フィールドに「コンピュータ/マザーボード」と入力します。
   3 ［**属性**］（SQL名：Nature）
   4 ［**バーコード**］（SQL名：BarCode）
3 ［**作成**］をクリックして、この新しいモデルを作成します。
4 ［**任意管理項目**］タブページを表示し、⊞ボタンをクリックして任意管理項目を追加します。選択画面に、スクリプトを編集した任意管理項目の名前が表示されます。
5 ［**親モデル**］の値を「/コンピュータ/」に変更します。［**変更**］をクリックして変更を確定します。
6 モデルの詳細画面の［**任意管理項目**］タブページを選択し、⊞ボタンをクリックして任意管理項目を追加します。選択画面には、スクリプトを編集した任意管理項目の名前が表示されません。

このスクリプトは正常に機能していることになります。

スクリプトライブラリ

Asset Managerでは、スクリプトへのアクセスを統括するために、スクリプトライブラリを保存できるようになっています。

［**管理／スクリプトライブラリ**］メニューを使うと、スクリプトライブラリにアクセスできます。

保存されたライブラリはAPIコマンド**amEvalScript**により呼び出されます。

APIの**amEvalScript**に関する詳細は、『Programmer's Reference』ガイドの「Alphabetic reference」のセクションを参照してください。

概念

Asset Managerでは、1つのスクリプトは1つの関数を定義します。

スクリプトライブラリを作成するということは、複数の関数の集まりを定義することに当たります。

スクリプトライブラリを作成する

スクリプトライブラリを作成するには：

1 スクリプトライブラリのリストを開きます。
2 ［**名前**］フィールドにライブラリの名前を入力します。
3 ［**スクリプト**］フィールドにスクリプトを入力します。
4 ［**作成**］をクリックしてスクリプトを確定します。

例えば「biblio」という名のライブラリを作成し、次のスクリプトを入力します。

```
function FullName(strName As String, strFirstName As String) As String
FullName = strFirstName & ", " & strName
end function
```

この関数は、従業員の姓と名から構成される文字列を戻します。

警告:

各関数の名前は、複数のスクリプトライブラリ全体の中で、固有の名前でなければなりません。

スクリプトライブラリ内に記録されているスクリプトを呼び出す

ライブラリからスクリプトを呼び出すには、ライブラリの名前、スクリプト内で定義された関数と、関数に関連付けられたパラメータを定義する必要があります。

例えば、先ほど作成したライブラリを使用するスクリプトタイプの、「callEvalScript」アクションを作成するとします。

1 ［**コンテキスト**］（SQL名：ContextTable）フィールドに［**部署と従業員**］（SQL名：amEmplDept）テーブルを入力します。
2 ［**スクリプト**］タブに以下のスクリプトを入力します。

```
Dim strFullName As String
strFullName = amEvalScript("biblio", "FullName", "", [Name], [FirstName
])
amMsgBox (strFullName)
```

このスクリプトは「biblio」ライブラリから「FullName」関数を呼び出し、ダイアログボックス内に従業員の姓名を表示します。

3 ［**作成**］をクリックして作成を確定します。

注意:

通常「amEvalSript」APIと共に使用されるコンテキストパラメータは、スクリプトライブラリを呼び出す場合には使用されません。

ヒントと注意事項

ここでは、スクリプトを作成する時に役立ついくつかのヒントを紹介します。

プログラマブル関数を使うときの注意事項

スクリプトを作成するときは、次の点に注意してください。

- フィールドやリンクのデフォルト値を定義する関数などのプログラマブル関数は、関数の戻り値を設定することが目的です。プログラマブル関数内で他の操作を実行しないでください。関数が正常に機能しているときは一般的にパフォーマンスが向上しますが、関数を誤って使うとデータベースが破損するおそれがあります。
- プログラマブル関数は、Asset Managerの様々な機能で使用します。可能な限りスクリプトを最適化して、Asset Managerの全体的なパフォーマンスを維持してください。

スクリプトで使用する「日付＋時刻」定数の形式

スクリプトで参照する日付は、ユーザが定義した表示形式に関係なく、次の国際標準形式で指定します。

yyyy/mm/dd hh:mm:ss

例：

```
RetVal="2001/07/12 13:05:00"
```

注意:

日付の区切り文字としてハイフン（-）を使うこともできます。

Basicでの日付形式

Basicでは、国際標準形式、またはDouble（倍精度）型の浮動小数点数で日付を記述します。浮動小数点数の場合は、整数部は1899年12月30日の午前0時から数えて現在まで経過した日数、小数部は本日の午前0時から現在まで経過した時間の1日（86400秒）に対する割合（現在までの経過秒数を86400で割ったもの）を示します。

スクリプトでの「Duration（時間）」の形式

スクリプトでは、時間を秒単位で記述して保存します。例えば、Duration（時間）型フィールドのデフォルト値を3日に設定するには、次のスクリプトを使います。

```
RetVal=259200
```

同様に、**AmWorkTimeSpanBetween()**などの時間を計算する関数も、秒単位の時間を返します。

注意:

Asset Managerでは、1年は12ヶ月、1月は30日として計算するので、1年は360日になります。

システムリストデータの読取りと書込み

Asset Managerでは、リストデータ内のそれぞれの値に整数を割り当てて、システムリストデータを管理しています。

資産の詳細画面の［**全般**］タブページの［**標準割当**］枠内の［**割当**］（SQL名：seAssignment）フィールド用の、リストデータの例は以下のとおりです。

リストデータの値に対応する整数は次のようになります。

リストデータの値	対応する整数
使用中	0
在庫中	1
除却済(または消費済)	2
納品待ち	3

この原理から、リストデータのデフォルト値を定義するには、次の操作を行う必要があります。

1 デフォルト値にふさわしい値に対応する整数を確認する。
2 次の文字列を編集する。

```
RetVal=<デフォルト値にふさわしい値に対応する整数>
```

例えば、［**割当**］フィールドで使うシステムリストデータのデフォルト値を［**納品待ち**］に設定するには、文字列を次のように編集する必要があります。

```
RetVal=3
```

注意:

システムリストデータを、ユーザ定義のクローズ済リストデータと混同しないでください。

注意:

システムリストデータ値の完全なリストは、Asset Managerのインストール先フォルダ内にある「**doc\infos**」サブフォルダの「Database.txt」にあります。「Data display and entry type」と「Additional information on data display and entry type」という2つの列に、リストデータのデータ型と各リストデータの値の一覧があります。

CurrentUser仮想リンク

定義

CurrentUserは、先ず全テーブルを検索し、次に［部署と従業員］テーブル内にある現在のユーザのレコードに検索を絞るリンク、と見なされます。

- 「CurrentUser」という形式を使うと、現在のユーザのレコードを検索し、［部署と従業員］テーブルから表記文字列を返します。
- 「CurrentUser.<フィールドのSQL名>」という形式を使うと、現在のユーザのフィールドの値を返します。

注意:

仮想リンクは、フィールドとリンクのリストには表示されません。そのため、CurrentUserは、AssetCenter内のスクリプトビルダには直接表示されません。手動で「CurrentUser」と入力してください。

等価関数

AmLoginName()と**AmLoginId()**関数は、それぞれ現在のユーザ名とIDを返す関数で、CurrentUserから派生した関数と見なされます。これらの関数には、次のような関係が成り立ちます。

- AmLoginName()=[CurrentUser.Name]
- AmLoginId()=[CurrentUser.lPersId]

制限

CurrentUserは、コンテキストが定義されていないと動作しません（コンテキストはテーブル）。

コンテキストがない場合、他の関数を使用する必要があります。

例：

Asset Managerデータベースに接続しているユーザによって異なるパスのファイルを実行する、コンテキストを指定しないアクションを作成したいとします。

コンテキストを指定するアクションであれば、**実行可能ファイル**タイプのアクションを作成し、**フォルダ**フィールド（Folder）を例えば **c:\scripts\[CurrentUser.Name]**などに設定することができます。

しかし、**Executable**タイプのアクションにコンテキストがない場合、**[CurrentUser.Name]**は固定のテキストと見なされます。

このため、別の方法を見つける必要があります。例えば、次のような**スクリプト**タイプのコンテキストを指定しないアクションを作成します。

```
RetVal = amActionExec("program.exe","c:\scripts\" + amLoginName())
```

古い仮想リンク

このリンクは、ワークフロー内で使用されるアクションに属するスクリプト内で使用します。

定義

古いリンクは、レコードを変更する前にワークフローのコンテキストを参照することができます。

制限

このリンクは、ワークフローによって使用されるアクションに属するスクリプト内でしか使用できません。

シンタックス

ワークフローによってフィールドを更新する前にフィールドの値を取得するには、次のシンタックスを使用します。

```
[Old.Field]
```

Basicスクリプトにコメントを付ける

Basicスクリプトにコメントを付けスクリプトの機能を説明すると、ユーザのスクリプトの理解が深まり、ユーザが各自でスクリプトを変更できるようになります。Asset Managerでは、一重垂直引用符（'）文字を使って、スクリプトの本体にコメントを付けることができます。一重垂直引用符（'）で始まる行の文字はすべて、コンパイラによってコメントとして解釈され、無視されます。コメントを付けるには、次の2つの方法があります。

- 次に示すように、Basicスクリプト内にコメントだけの行を加える。

```
' 資産のテーブルのBarCodeフィールドの値をテストする
' この値がPC1の場合は戻りコードをTRUEに設定する
If [BarCode]="PC1" Then
RetVal=True
End If
```

- 行の終わりにコメントを追加し、Basicコンパイラにコメントとして解釈させる。

```
If [BarCode]="PC1" Then ' BarCodeがPC1の場合
RetVal=TRUE ' 戻り値をTRUEに設定
End If ' テストの終わり
```

エラーメッセージをトリガする

Err.Raise関数を使って、必要に応じてエラーメッセージをトリガすることができます。シンタックスは次のとおりです。

```
Err.Raise (<エラー番号>, <エラーメッセージ>)
```

注意:

ユーザがテーブルで作成または変更したレコードが、関連するテーブルの［有効］フィールドの値により無効になった場合などに、Err.Raise関数を使ってエラーメッセージをトリガさせると便利です。警告メッセージを表示させることにより、レコードが作成または変更できない原因をユーザに知らせることができます。

スクリプト例1

ここでは、Basicスクリプトを使って処理できる作業例を示します。スクリプトの記述方法の説明を読む前に、自分でスクリプトを検討してみてください。

スクリプトの目的

作業指示のテーブルに関連付けられている「例1」という任意管理項目は、作業指示がクローズした時に入力する必要があります。作業指示がオープンの時は、入力しなくても構いません。この例ではこの任意管理項目は作成済みで、下図の

ようにリンク先のテーブルは作業指示であり、［使用可能］と［デフォルトで表示］が制限パラメータとして設定されていることにします。

手順1：アルゴリズムを分析および定義する

次のタスクを実行するアルゴリズムを作成する必要があります。

- チケットがクローズドの場合、［**必須**］（SQL名：seMandatory）フィールドを［**はい**］に設定する。
- それ以外の場合はすべて、［**必須**］（SQL名：seMandatory）フィールドを［**いいえ**］に設定する。

アルゴリズムは次のようになります。

作業指示がクローズ済の場合、 任意管理項目の入力は必須 それ以外の場合は 任意管理項目の入力は必須ではない

作業指示の［**ステータス**］（SQL名：seStatus）フィールドが［**クローズ済**］の場合、作業指示は終了しています。

したがって、任意管理項目の［**必須**］（SQL名：seMandatory）フィールドの値の条件となるのは、作業指示テーブルの［**ステータス**］（SQL名：seStatus）フィールドの値です。アルゴリズムでは、このフィールドしか使いません。

［BASICスクリプト］画面のドロップダウンリストから、作業指示テーブルの［**ステータス**］フィールドを選択します。

フィールドを選択したら、［↓］ボタンをクリックして、編集ウィンドウに値を移します。

このフィールドには、システムリストデータの値が入ります。

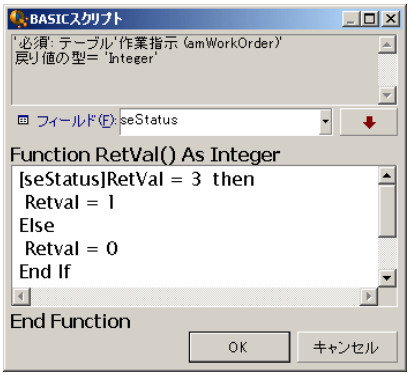
リストデータの値	対応する整数
通知済	0
作業日確定済	1
進行中	2
クローズ済	3

ここでは以下のリストデータの値を使用します。

- 値「3」に対応する［**クローズ済**］

手順2：Basicスクリプトを作成する

手順1のアルゴリズムをBasicで作成します。



［**OK**］をクリックして、スクリプトの作成を確定します。

手順3：Basicスクリプトをテストする

この手順では、スクリプトが正常に機能することを確認します。

1. ［**ポートフォリオ／作業指示**］メニューを使って［作業指示］テーブルを表示し、ステータスが［**クローズ済**］の作業指示を選択します（ない場合は、このステータスの作業指示を作成します）。
2. ［**任意管理項目**］タブを選択します。関連する作業指示には任意管理項目が追加され、この任意管理項目は必須になっています。
3. ステータスが［**クローズ済**］以外の作業指示を選択します。この作業指示の［**任意管理項目**］タブページに移動します。任意管理項目「例1」は表示されますが、この任意管理項目はオプションフィールドになっています。

このスクリプトは正常に機能していることになります。

スクリプト例2

ここでは、Basicスクリプトを使って処理できる作業例を示します。スクリプトの記述方法の説明を読む前に、自分でスクリプトを検討してみてください。

スクリプトの目的

従業員の詳細画面の［**フィールド1**］（SQL名：Field1）フィールドに、従業員の名前と姓がわかる場合はその両方を、名前がわからない場合は姓だけをデフォルトで表示するようにします。

手順1：アルゴリズムを分析および定義する

次のタスクを実行するアルゴリズムを作成する必要があります。

- 従業員の姓と名前の両方がわかる場合は、従業員の詳細画面の［**フィールド1**］（SQL名：Field1）に従業員の姓名をデフォルトで表示する。
- 名前がわからない場合は、従業員の詳細画面の［**フィールド1**］（SQL名：Field1）に姓だけをデフォルトで表示する。

アルゴリズムは次のようになります。

```
If the first name of the employee does not exist Then
The default value of "Field1" is the name of the employee
Else
The default value of "Field1" is "Last name," "First name"
```

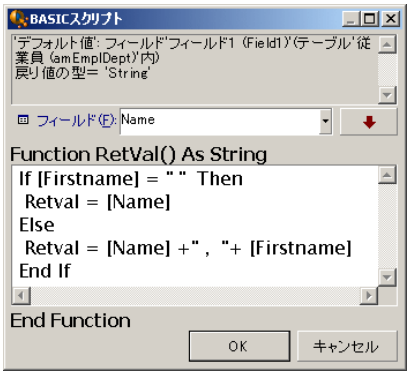
従って、従業員の詳細画面の［**フィールド1**］（SQL名：Field1）のデフォルト値の条件となるのは、従業員のテーブルの［**部署名/姓**］（SQL名：Name）フィールドと［**名前**］（SQL名：FirstName）フィールドの値です。アルゴリズムでは、これら2つのフィールドしか使いません。

従業員の詳細画面の［**フィールド1**］（SQL名：Field1）をマウスの右ボタンでクリックして、ポップアップメニューから［**オブジェクトの設定**］を選択します。

［**デフォルト**］フィールドの横にある ボタンをクリックして、Basicスクリプトを編集します。

手順2：Basicスクリプトを作成する

手順1のアルゴリズムをBasicで作成します。



［**OK**］をクリックして、スクリプトの作成を確定します。

手順3：Basicスクリプトをテストする

この手順では、スクリプトが正常に機能することを確認します。

1 ナビゲーションバーの［**組織／部署と従業員**］リンクを選択して［部署と従業員］テーブルを開き、新しい従業員を作成します。

2 ［**部署名/姓**］（SQL名：Name）フィールドと［**名前**］（SQL名：FirstName）フィールドに入力し、［**作成**］をクリックして作成を確定します。［**フィールド1**］（SQL名：Field1）に従業員の姓と名前が表示されます。

このスクリプトは正常に機能していることになります。

12 カレンダ

カレンダのリストにアクセスするには、［ツール／カレンダ］メニューを使います。

カレンダの概要

カレンダの詳細画面

カレンダの詳細画面には、次の情報が表示されます。

- カレンダを固有に識別できる一般情報
  - カレンダの［**名前**］（SQL名：Name）
  - カレンダがリンクしている［**タイムゾーン**］（SQL名：TimeZone）
- ［**予定表**］タブページ：通常の業務時間
- ［**例外**］タブページ：［予定表］タブページで指定した予定表を適用しない期間のリスト
- ［**予定表**］および［**例外**］タブページで定義した規則を適用した業務時間のプレビュー

カレンダの使用方法

カレンダは次の項目に関連付けられます。

- ワークフローのアクティビティ

ワークフローのアクティビティで定義したアラームをいつトリガするかを設定できます。これらのアラームで定義する［**制限時間**］は、実際の業務時間で指定します。

警告:

データベースでカレンダを変更すると、ユーザがAsset Managerを終了し再起動した時に、カレンダにリンクしているフィールドにのみ変更が適用されます。

カレンダに影響される機能

カレンダは、Asset Managerの特定の機能に影響を与えます。カレンダを変更すると、データベースの特定のレコードのデータが直接および間接的に変更されます。カレンダは次の要素に関連しています。

- ワークフローのタスクの実行時間
- ワークフローのアクティビティに関連付けられているアラーム

カレンダの作成方法

次の手順でカレンダを作成します。

1. カレンダに［**名前**］（SQL名：Name）を付けます
2. 必要に応じて、［**タイムゾーン**］（SQL名：TimeZone）フィールドに入力して、カレンダをタイムゾーンに関連付けます。
3. カレンダの詳細画面の［**予定表**］タブページで、通常の業務時間を定義します。
4. カレンダの詳細画面の［**例外**］タブページで、これらの業務時間以外の予定表を定義します。
5. 最後に、［**プレビュー**］タブページでカレンダの機能を確認できます。

カレンダの作成手順

カレンダの作成手順は以下の通りです。

1. 一般情報を入力する
2. ［予定表］タブページに入力する
3. ［除外］タブページに入力する
4. カレンダをチェックする

一般情報を入力する

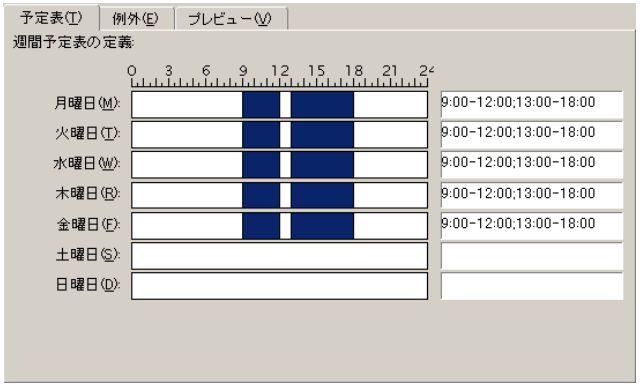
業務時間と除外項目の詳細情報を入力する前に、詳細画面の［**名前**］（SQL名：Name）フィールドに、カレンダを識別する名前を入力します。

また、［**タイムゾーン**］（SQL名：TimeZone）フィールドに入力して、カレンダに使うタイムゾーンを指定することもできます。

［予定表］タブページに入力する

カレンダの詳細画面の［**予定表**］タブページでは、カレンダの週単位のカレンダを定義します。このタブページで定義した業務時間によって週単位のカレンダが決まります。業務時間には一般的な規則を定義します。休日などはこの規則の適用外として、［**例外**］タブページで定義します。

図 12.1. カレンダの詳細画面の［予定表］タブページ



各曜日ごとに、複数の業務時間を定義できます。次の2つの方法で定義できます。

- 週の各曜日を表す目盛スライダーコントロールを使って定義する。
  1. 業務時間の開始点でコントロールをクリックします。
  2. 業務時間の終点までマウスをドラッグします。目盛コントロールの右側のテキストボックスに自動的に時間が入力されます。
  3. 必要に応じて、以上の操作を繰り返します。
- テキストフィールドに業務時間を直接数値で指定する。次のシンタックスで入力します。

```
<業務時間の開始時間>-<業務時間の終了時間>;<業務時間の開始時間>-<業務時間の終了時間>;...
```

時刻は次の形式で入力します。

<hh:mm[{AM|PM}]>

オプションの[午前|午後]パラメータを定義しない場合は、24時間形式を使うものと見なされます。

テキストフィールドの左側の目盛スライダコントロールに自動的に入力されます。

注意:

グラフィカルコントロールを使う場合は、30分単位で指定できます。直接時間を数値で入力する場合は、1分単位で指定できます。

［除外］タブページに入力する

カレンダの詳細画面の［**例外**］タブページでは、［**予定表**］タブページで定義した週単位の業務時間を適用しない期間を定義します。

出力スクリプトの作成方法

例外の期間を作成するには：

1 例外の期間に［名前］を付けます。
2 ［**日数**］フィールドに、業務時間を適用しない日数を指定します。日数は日、月、または年単位で定義できます。
3 ［**作業時間**］フィールドを使って、例外期間中の業務時間を定義することもできます。このフィールドを使うと、「各月の最後の金曜日、チームは午前８時30分から午前10時30分までと、午後5時30分から午後6時30分まで勤務する」など、より細かく期間を定義できます。

［例外］タブページ

このタブページは、次の2つのセクションで構成されています。

- タブページの上部には例外期間のリストが表示されます。ツールバーのボタンを使って、例外期間の作成、複製、破棄、変更および取り消しを行うことができます。
  - 新しい例外期間を作成します。
  - 例外期間を削除します。
  - 例外期間を複製します。

注意:

［ランク］列を使うと、例外期間を優先順位で並べ替えることがきます。同じ指定の例外期間がある場合に、優先順位を確認できます。例外期間には自動的にランク（「P00」から「P15」）が割り当てられます。数字が小さいほど、例外期間の優先順位が高くなります。例えば、「P06」の例外期間は、「P10」の例外期間より優先されます。

- タブページの下部では、例外期間の詳細を指定します。

  ［日数］フィールドの値によって、次のように例外期間が定義されます。

［実行日］フィールドの値	例外期間
毎日	例外なくその年のすべての日が例外期間になります。
日付指定	［日］、［月］、および［年］チェックボックスを使って定義する日が例外期間になります。
第1	［月］および［年］チェックボックスをオンにした月および年の、［日］チェックボックスで定義した曜日が例外期間になります。 **例** 毎月「第1」金曜日
第2	［月］および［年］チェックボックスをオンにした月および年の、［日］チェックボックスで定義した曜日が例外期間になります。 **例** 9月の「第2」月曜日
最後から2番目	［月］および［年］チェックボックスをオンにした月および年の、［日］チェックボックスで定義した曜日が例外期間になります。 **例** 11月の「最後から2番目」の水曜日

［実行日］フィールドの値	例外期間
最後	［月］および［年］チェックボックスをオンにした月および年の、［日］チェックボックスで定義した曜日が例外期間になります。 **例** 2000年の毎月「最終」火曜日

例

Taltek社の従業員の休日は以下のように指定されています。

- 各月の第1金曜日は休日
- 8月中は、午前8時30分から午後12時30分までの午前中だけ勤務する。

規則1：各月の第1金曜日は休日

1 ［**新規作成**］をクリックします。
2 例外期間は、各月の第1金曜日です。この場合は、［**月**］および［**年**］チェックボックスをオフのままにして、例外期間が月や年に関係ないことを指定します。例外期間は金曜日だけなので、［**日**］チェックボックスはオンにします。
3 ［**日数**］フィールドを「第1」に設定し、その隣のフィールドに「金曜日」を指定します。

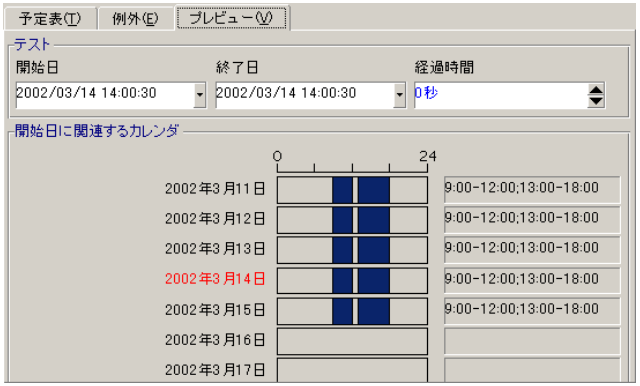
規則2：8月中は、午前8時30分から午後12時30分までの午前中だけ勤務する

1 ［**新規作成**］をクリックします。
2 例外期間は、8月だけです。この場合は、［**日**］および［**年**］チェックボックスはオフにしたまま、［**月**］チェックボックスをオンにして、「8月」に設定します。
3 例外期間は選択した日（8月いっぱい）だけになるので、［**日数**］フィールドを「日付指定」に設定します。
4 この期間中は、従業員は午前中だけ勤務します。［**作業時間**］フィールドで午前8時30分から午後12時30分までの時間を指定します。

カレンダをチェックする

［**プレビュー**］タブページでは、［**予定表**］および［**除外**］タブページで定義した規則を適用した結果を確認できます。確認する期間を**テスト**枠内の［**開始日**］および［**終了日**］フィールドを使って指定します。

図 12.2. カレンダの詳細画面の［プレビュー］タブページ



- ［**開始日に関連するカレンダ**］オプションには、デフォルトで、指定した［開始日］を含む週の業務時間のプレビューが表示されます。
- ［**経過時間**］フィールドには、選択した期間中の業務時間の合計時間が表示されます。

注意:

［開始日］を指定してから、［経過時間］フィールドに時間を入力すると自動的に［終了日］が計算されます。

13 タイムゾーン

この章では、タイムゾーン機能の使用方法について説明します。

注意:

この機能を使用するには、専用のAsset Managerライセンス契約を取得する必要があります。

タイムゾーンを管理する理由

クライアントマシンとデータベースサーバは地理的に別々の場所に配置できるので、Asset Managerではグリニッジ平均時（GMT）を基準にタイムゾーンと時差を管理します。次の規則が適用されます。

- クライアントマシンでは、「日付＋時刻」型のフィールドはすべて、クライアントマシンのタイムゾーンを反映して表示されます。
- サーバの場合、「日付＋時刻」型のフィールドはすべて、定義されたタイムゾーンを基準として格納されます。
- 日付と時刻に関するすべての計算には、時差も反映されます。

例

パリ（フランス）のタイムゾーンに従ってインデックス付けされたデータを持つサーバがニューヨークに置かれ、2台のクライアントマシンがロンドンとパリに

配置されているとします。まず、グリニッジ平均時に従ってこれらの各クライアントマシンのタイムゾーンを定義します。

- サーバのタイムゾーン = GMT-5
- パリのクライアントのタイムゾーン = GMT+1
- ロンドンのクライアントのタイムゾーン = GMT
- データのタイムゾーン = GMT+1

すべての「日付+時刻」型の値は、サーバ上でGMT+1形式で格納され、パリのクライアントではGMT+1、ロンドンのクライアントではGMTとして表示されます。例えば、ロンドンのクライアントマシンで作業指示を行う場合、作業指示の解決期限が2000年5月15日の17時30分に設定されているとすると、他のマシンには次のように表示されます。

- サーバ：1998年5月15日午後12時30分
- パリのクライアント：1998年5月15日午後6時30分
- ロンドンのクライアント：1998年5月15日午後5時30分

タイムゾーン機能を実装する

Asset Managerでタイムゾーンを正しく処理するには、次の操作を実行する必要があります。

1. Asset Manager Application Designerでデータベースを作成する時に［**タイムゾーンの使用**］オプションを使って、タイムゾーンを定義します。
2. タイムゾーンを作成します（例えば、タイムゾーンに関連する情報をインポートします）。
3. ［**ツール／タイムゾーン**］メニューを使って、マシンのタイムゾーンを定義します。
4. タイムゾーンに従ってカレンダを定義します。

タイムゾーンを作成する

Asset Managerのタイムゾーン機能は、Windowsと異なり、年間を通じて夏時間規則の変更を考慮します。これにより、経過した現地時間をより正確に表示できます。Asset Managerではタイムゾーン情報を使って、次の操作を行うことができます。

- 夏時間を反映したローカルの日付および時刻を表示します。
- 別の場所の時間を表示します。

タイムゾーンを手動で定義する手間を省くために、Asset Managerには、主要なタイムゾーンを含む記述ファイルが添付されています。このファイルは、次の手順でインポートできます。

1 ［ファイル／インポート］メニューを選択します。インポートの選択ウィンドウが開きます。

2 をクリックして［スクリプトの実行］を選択します。Asset Managerはデータベース更新画面を表示します。をクリックして実行するスクリプトを選択します。この場合、選択するファイルは、Asset Managerのインストール先フォルダの「**datasys**」サブフォルダにある「tz.scr」です。

3 ［インポート］をクリックします。スクリプトに従って、インポートが実行されます。

タイムゾーンを管理する

ここでは、［夏時間］（SQL名：memDaylightInfo）フィールドについて詳しく説明します。

夏時間フィールドのフォーマット

［夏時間］（SQL名：memDaylightInfo）フィールドは次のような構造をとります。

```
<Year>=<DaylightInfo>|<Year>
=<DaylightInfo>|<Year>=<DaylightInfo>|...
```

この節では、これらの用語を次のように表記します。

- <Year>=<DaylightInfo>をまとめて「パラメータ」と表記
- <Year>と<DaylightInfo>は、それぞれ「引数」と表記

<Year>および<DaylightInfo>引数に応じて変化する夏時間の概要は次の表のとおりです。

	<DaylightInfo>引数に値なし	<DaylightInfo>引数に値がある
<Year>引数に値なし（"<Year>="は表示されない）	指定しているタイムゾーン全体について夏時間は変更されません。	<Year>引数を持つパラメータで定義された年を除くすべての年について夏時間情報が有効になります。

	<DaylightInfo>引数に値なし	<DaylightInfo>引数に値がある
<Year>引数に値がある	適用できません。	<Year>引数で指定した年から次の<Year>引数までの各年について、指定しているタイムゾーンの夏時間情報が有効になります。

<Year>引数の値

<DaylightInfo>で定義された夏時間変更情報を適用する年を指定する<Year>引数には、4桁の年値を指定します（1990、1997、1998、2012など）。

<DaylightInfo>引数の値

<DaylightInfo>引数の完全な形式は次のとおりです。

```
<StdShift>,<DltShift>,<SDay>
,<SMonth>,<SDayPos>,<SHour>
,<DDay>,<DMonth>,<DDayPos>,<DHour>
```

この引数は、次に示す複数の副引数で構成されています。

副引数	説明	可能な値
<StdShift>	タイムゾーン内の標準時間と対象となるタイムゾーンの時間との時差を分単位で示します。 例えば、パリ（GMT+1タイムゾーン）の場合、<StdShift>を30（分）に設定すると、このタイムゾーン内の標準時間はGMT+1時間ではなく、GMT+1時間30分となります。	デフォルトでは、この副引数はNULLですが、任意の数値を設定できます。ユーザは、この副引数の一貫性を確認する必要があります。
<DltShift>	夏時間と対象タイムゾーンの時間との時差を分単位で示します。	デフォルトでは、この副引数は60に設定されます（夏時間と参照時間「GMT+」間の1時間の時差に対応する）が、任意の数値を設定できます。ユーザは、この副引数の一貫性を確認する必要があります。

副引数	説明	可能な値
<SDay>	夏時間から標準時間に変更する日	"Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday" "Sunday" 値なし（この場合、<SDayPos>を1から31の値に設定して、標準時間から夏時間に変更する日を指定する必要があります。）
<SMonth>	夏時間から標準時間に変更する月	"January" "February" "March" "April" その他 "November" "December"
<SDayPos>	夏時間から標準時間に変更する月内の日の位置	"First" "Second" "Third" "Fourth" "Last" "最後から2番目" <SDay>に値がない場合は、1から31の値
<SHour>	夏時間から標準時間に変更する時刻（夏時間で指定）	24時間形式（HH:MM:SS）の値
<DDay>	標準時間から夏時間に変更する日	"Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday" "Sunday" 値なし（この場合、<DDayPos>を1から31の値に設定して、夏時間から標準時間に変更する日を指定する必要があります。

副引数	説明	可能な値
<DMonth>	標準時間から夏時間に変更する月	"January" "February" "March" "April" その他 "November" "December"
<DDayPos>	夏時間から標準時間に変更する月内の日の位置	"First" "Second" "Third" "Fourth" "Last" "最後から2番目" <DDay>に値がない場合は、1から31の値
<DHour>	標準時間から夏時間に変更する時刻（標準時間で指定）	24時間形式（HH:MM:SS）の値

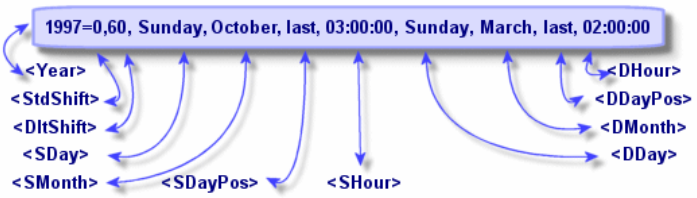
例

「（GMT+01:00）パリ、マドリード、アムステルダム」タイムゾーンの夏時間情報の例を見てみましょう。

```
2000=0,60,Sunday,October,last,03:00:00,Sunday,March,last,02:00:00|
0,60,Sunday,September,last,03:00:00,Sunday,March,last,02:00:00
```

使用されているパラメータは次のとおりです。

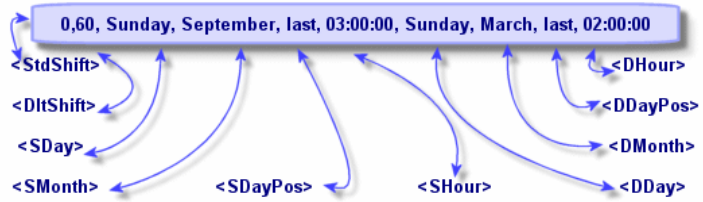
最初のパラメータ



- <Year> = 2000は、次の<DaylightInfo>パラメータが2000年から後に適用されることを意味します。

- <StdShift> = 0は、タイムゾーンとそのタイムゾーン内の標準時間に差がないことを意味します。つまり、冬時間はそのタイムゾーンの時間と等しくなります。
- <DltShift> = 60は、標準時間と夏時間の時差が60分、つまり1時間であることを意味します。夏時間は、タイムゾーンの時間に1時間を加えた値に等しくなります。
- <SDay> = Sundayは、標準時間への変更が日曜日に行われることを意味します。
- <SMonth> = Octoberは、標準時間への変更が10月中に行われることを意味します。
- <SDayPos> = Lastは、月内での日の位置を指定します。この例では、標準時間への変更は、10月の最後の日曜日に行われます。
- <SHour> = 03:00:00は、標準時間への変更が午前3時に行われることを意味します。
- <DDay> = Sundayは、夏時間への変更が日曜日に行われることを意味します。
- <DMonth> = Marchは、夏時間への変更が3月に行われることを意味します。
- <DDayPos> = Lastは、月内での日の位置を指定します。この例では、夏時間への変更は、3月の最後の日曜日に行われます。
- <DHour> = 02:00:00は、夏時間への変更が午前2時に行われることを意味します。

第2のパラメータ



- <Year>引数がないので、このパラメータは、前のパラメータで指定されていない年だけに有効であることを示します。
- <StdShift> = 0は、タイムゾーンとそのタイムゾーン内の標準時間に差がないことを意味します。つまり、冬時間はそのタイムゾーンの時間と等しくなります。
- <DltShift> = 60は、標準時間と夏時間の時差が60分、つまり1時間であることを意味します。夏時間は、タイムゾーンの時間に1時間を加えた値に等しくなります。
- <SDay> = Sundayは、標準時間への変更が日曜日に行われることを意味します。

- <SMonth> = Septemberは、標準時間への変更が9月中に行われることを意味します。
- <SDayPos> = Lastは、月内での日の位置を指定します。この例では、標準時間への変更は、9月の最後の日曜日に行われます。
- <SHour> = 03:00:00は、標準時間への変更が午前3時に行われることを意味します。
- <DDay> = Sundayは、夏時間への変更が日曜日に行われることを意味します。
- <DMonth> = Marchは、夏時間への変更が3月に行われることを意味します。
- <DDayPos> = Lastは、月内での日の位置を指定します。この例では、夏時間への変更は、3月の最後の日曜日に行われます。
- <DHour> = 02:00:00は、夏時間への変更が午前2時に行われることを意味します。

結果

注意:

2000年からは、標準時間への変更は、10月最後の日曜日の午前3時（時計を午前2時に戻す）、夏時間への変更は、3月最後の日曜日の午前2時（時計を午前3時に進める）に行います。

2000年より前の年は、標準時間への変更は9月最後の日曜日の午前3時、夏時間への変更は3月最後の日曜日の午前2時に行います。

Asset Manager Automated Process Managerでタイムゾーンを管理する

Asset Manager Automated Process Managerを使って、タイムゾーンのテスト方法を設定できます。［**ツール／モジュールの設定**］メニューを選択します。

実行するテスト

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の時間の検証

これらの2つのテストはいずれも、データベースサーバの時刻とAsset Manager Automated Process Managerをインストールしたマシンの時刻を比較します。時間差は[(n * 30分) + m]で表されます。ただし、mは-15から+15分です。

2つのテストに共通する特徴

時間差が5分を超えると、Asset Manager Automated Process Managerがインストールされているマシンのローカル時刻を更新するメッセージが表示されます。

例えば、サーバの時刻の更新が必要な場合にこの更新を拒否すると、接続が拒否されます。データベースサーバの時刻とAsset Manager Automated Process Managerをインストールしたマシンの時刻のいずれかを変更して、2つの時刻の差を5分以内にすると、再接続できるようになります。

［データベースサーバのタイムゾーンの検証］オプションの特徴

必要に応じて、Asset Managerのオプションテーブルで指定されているサーバのタイムゾーンに関する情報が更新されます（(n * 30分)の数値がサーバのタイムゾーンに対応しない場合）。

注意:

この機能を正しく実行するには、Asset Manager Automated Process Managerが稼動しているマシンで、時刻と夏時間の変更に関する情報を正しく設定しておく必要があります。

［サーバと比較したローカルマシン時間の検証］オプションの特徴

Asset Managerの内部操作に必要なサーバのタイムゾーンを取得します。

テストの頻度

テストは次の場合に実行されます。

1. まず、Asset Manager Automated Process Managerがデータベースに接続したときに実行されます。
2. その後は、Asset Manager Serverのモジュール設定画面（**［ツール／モジュールの設定］**）で定義するスケジュールに従って、定期的に実行されます。

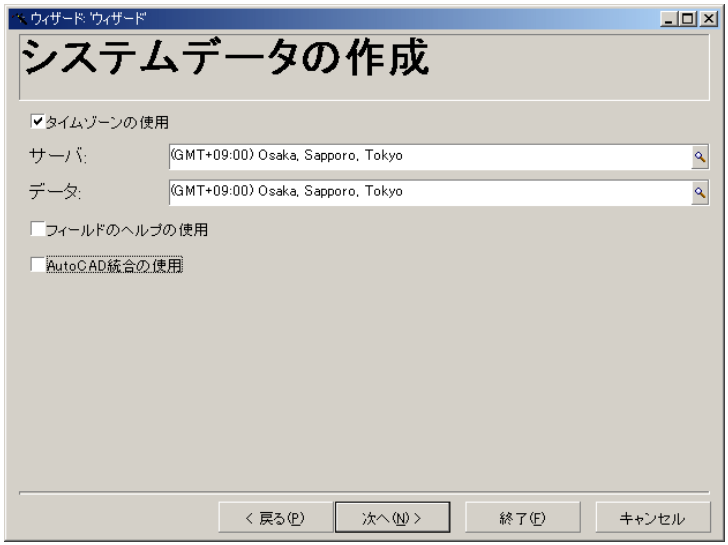
操作に与える影響

タイムゾーンは、様々な操作に影響を与えます。

- データベースの作成時
- データベースへの接続時
- インポートおよび出力時

データベースの作成時

データベースの作成時にタイムゾーンオプションを定義できます。Asset Manager Application Designerで［**アクション／データベースの作成**］メニューを選択します。［**システムデータの作成**］フレームでタイムゾーンに関するオプションを指定できます。



［**タイムゾーンの使用**］オプションを使って、データベースの作成時にタイムゾーン機能を使用するかどうかを指定します。

- チェックボックスをオンにすると、データベースの作成時にタイムゾーンが使われます。
- チェックボックスをオンにしない場合は、データベースの作成時にタイムゾーンは使用されません。

［**サーバ**］と［**データ**］フィールドでは、データベースを作成した時点で、サーバ上で有効にするタイムゾーンとデータを保存する時に使うタイムゾーンを指定します。

注意:

このオプションは、データベースを作成するときだけ使用できます。サーバとデータの基準時間を定義できます。これらの値を変更すると、データベース内の既存の「日付+時間」値は無意味になります。

データベースへの接続時

Asset Managerは、データベースに接続する時に、「am.ini」ファイルからクライアントマシンのタイムゾーンを定義する「LocalTimeZone」エントリを検索します。

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

この情報が見つからない場合は、Asset Managerはシステムのタイムゾーン（Windowsで定義）を使います。

次に、Asset Managerは、「am.ini」の「LocalTimeZone」エントリに対応するタイムゾーンまたはWindowsで定義されたタイムゾーンを、データベース内で検索します。

データベースへの接続時に「Am.ini」に加えられる変更は、次の表のとおりです。

「am.ini」に「LocalTimeZone」があるかどうか	タイムゾーンのテーブルに対応するタイムゾーンがあるかどうか	「am.ini」ファイルに保存される情報（「LocalTimeZone」エントリ）
可	可	データベースのタイムゾーン
	いいえ	変更なし
いいえ	可	データベースのタイムゾーン
	いいえ	システムのタイムゾーン

クライアントマシンで時間を修正する

リモートデータベースに接続すると、Asset Managerはクライアントマシンのクロック時刻の有効性をサーバクロックと比較して検証します。

クロック差は同期の差で、タイムゾーンの相違による転移である時間差と混同しないでください。Asset Managerは、クライアントクロックのタイムゾーンを計算してから、2台のマシン間のクロック差を決定します。次の方法で計算します。

クロック差 = 剰余((サーバとローカルマシン間の分単位の時差)/30)

注意:

剰余は、割算の余りです。

例えば、次のマシンがあるとします。

- マシンAのタイムゾーンはGMTで、18:02である。
- マシンBのタイムゾーンはGMT+1で、18:19である（つまりマシンAでは17:19で、マシンAとは43分の差がある）。

クロックの差 = 母数 (43/30)= 13分

この差が5分（固定値）を超えると、Asset Managerはクライアントマシンのクロック時間を修正するメッセージを表示します。

ユーザがこの修正を拒否すると、接続は失敗します。

Asset Managerはこのチェックを定期的に実行し、クライアントマシンで時間が変更された場合もチェックします。デフォルトでは、60分置きにテストを実行しますが、「am.ini」の［option］セクションにある**「g_lTimeZoneCheckInMns」**オプションでこの頻度を変更することもできます。

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

```
[option]
g_lTimeZoneCheckInMns = 30
```

上の例では、クロック差のチェックの頻度は30分に設定されています。

この頻度は、Asset Managerの［**データベースサーバのタイムゾーンの検証**］オプションでも設定できます。

注意:

この検証機能が使えるのは、タイムゾーン機能を有効にして作成したデータベースだけです。

インポートおよび出力時

この2つの機能の実行時は、データベース内のすべての「日付＋時刻」フィールドが、インポートまたは出力を行うマシンのタイムゾーンに対応しているものとしてデータが変換されます。

14 特殊フィールド

特殊フィールドの作成画面にアクセスするには、［**管理／特殊フィールド**］メニューを使います。

特殊フィールドの定義

特殊フィールドは、ユーザ定義の計算式を使って、他のフィールドや変数の値に応じて値が計算されるフィールドです。3種類の特殊フィールドがあります。

- AQL
- Basic
- 特殊フィールド

これらの特殊フィールドは、それぞれ異なる言語の計算式を使用しており、特定のオブジェクトにしかリンクできません。例えば、フィルタで使用できるのはAQLタイプの特殊フィールドだけです。

注意:

特殊フィールドは、読取り専用の仮想フィールド（データベースには計算式だけが格納される）です。必要な数の特殊フィールドを定義し、それぞれにユーザ権限を割り当てることができます。

特殊フィールドの用途

特殊フィールドを使うと、追加情報を定義したり、Asset Managerデータベースのテーブルのレコードに関する合成情報を計算したりすることができます。次のいくつかの点を除き、特殊フィールドは通常のデータベースフィールドと同じです。

- 通常のフィールドと異なり、特殊フィールドの値は、Asset Managerデータベースには格納されません。
- 特殊フィールドの値は、ユーザではなく計算式によって入力されます。
- 特殊フィールドを特定のフィールドの1個のレコードに関連付けることはできません。データベース内の他の通常のフィールドと同様、特殊フィールドは、テーブルのすべてのレコードに関連付けられ、そのテーブルの各レコードに対して値（NULLも可）を持ちます。
- 特殊フィールドはレコードの詳細画面には表示されません。リストにのみ表示できます。
- フィールドのタイプが［**特殊文字列**］または［**BASICスクリプト**］の場合は、特殊フィールドは標準フィールドのデフォルト値の計算だけに使用できます。

特殊フィールドを作成する

特殊フィールドを作成する前に、各特殊フィールドの特徴を理解することが大切です。

ここでは、次の内容について説明します。

- 特殊フィールドの概要
- 特殊フィールドの作成方法

特殊フィールドの概要

特殊フィールドのタイプはそれぞれ異なるプロパティを持ち、それによって使用方法も異なります。

3つのタイプの主な相違点は次の表のとおりです。

表 14.1. 特殊フィールドのタイプ

フィールドのタイプ	フィールドの特性			計算を実行するコンピュータ	計算式で使われる言語の特徴	
	表示可能	並べ替え可能	フィルタで使用可能		利点	欠点
AQL	可	可	可	データベース サーバ	強力 統合エディタ	限定的な言語 このタイプのフィールドはデフォルト値には使用できません。
特殊文字列	可	可	いいえ	クライアント	単純	強力ではない（文字列とフィールド値および文字列のみの単純な連結）
Basic	可	いいえ	いいえ	クライアント	多くの可能性 柔軟	このタイプのフィールドは表示のみが可能です。

注意:

この表から、AQLタイプのフィールドは他の2つの特殊フィールドに比べ、非常に用途が広いことが分かります。

AQLクエリでは、3つのプロパティ（表示可能、並べ替え可能、フィルタで使用可能）をすべて活用できます。

プロパティ	対応するAQL引数
表示可能	SELECT 句
並べ替え可能	SELECT ORDER BY GROUP BY 句

プロパティ	対応するAQL引数
フィルタで使用可能	SELECT ORDER BY GROUP BY WHERE HAVING 句

AQLクエリの詳細については、このガイドの「**AQLクエリ**」の章を参照してください。

サーバ／クライアントマシンによる計算

AQLタイプのフィールドの場合、必要な計算はデータベースサーバで実行され、結果がクライアントマシンに返されます。このため、クライアントマシンの処理速度への影響はまったくなく、ネットワークトラフィックが軽減されます。これに対し、データベースに送信されるSQLクエリはより複雑です。

特殊フィールドの作成方法

ここでは、特殊フィールドの作成方法について詳しく説明します。

用途を分析する

次の2つの基準に従って、フィールドのタイプを選択します。

- フィールドのタイプのプロパティ。表示、並べ替え、フィルタまたはデフォルト値として使用できるかどうか。
- 使用するタイプに応じてかかる「コスト」。使用状況と計算式の複雑さを比較します。3種類の特殊フィールドをコストの高い方から見ると次のようになります。
  - 特殊フィールド
  - AQL
  - Basic

注意:

可能な限り、最も「経済的」なタイプのフィールドを使うようにしてください。

例：

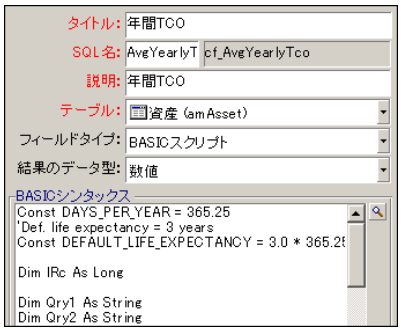
- 情報のみの場合は、Basicタイプの特殊フィールドで十分です。
- フィールド値に基づいてレコードを並べ替える場合は、AQLまたは特殊文字列を使う必要があります。

- フィールドの値に応じてフィルタを使ってレコードを検索する必要がある場合は、AQLを使う必要があります。

用途が明確になったら、次の手順に進みます。

作成画面を開く

［**管理／特殊フィールド**］メニューを選択します。特殊フィールドの作成画面が表示されます。



特殊フィールドの情報を入力する

まず、この画面の上部に入力して、特殊フィールドに固有の情報を指定します。

- ［**タイトル**］（SQL名：Label）フィールドには、リストの列ヘッダーとして使う、特殊フィールドのラベルを入力します。
- ［**SQL名**］（SQL名：SQLName）フィールドには、特殊フィールドのSQL名を入力します。この名前には、「cf_」という接頭語が付き、例えば、Basicスクリプト、クエリ、またはフィルタでこのフィールドを参照する場合に使います。

  注意:

  一度作成したフィールドのSQL名は変更できません。SQL名を変更すると、そのフィールドへの参照は無効になります。

- ［**説明**］（SQL名：Description）フィールドには、フィールドを表示するリスト（フィルタ、リストの設定画面など）で使う、フィールドの短い説明を入力します。

フィールドを使うコンテキストを定義する

［**テーブル**］（SQL名：TableName）および［**フィールドタイプ**］（SQL名：seType）フィールドでは、特殊フィールドを使うコンテキストを定義できます。

- ［**テーブル**］（SQL名：TableName）フィールドで、特殊フィールドを関連付けるテーブルを指定できます。フィールドはこのフィールドで指定したテーブルでのみ使用できます。

- ［**フィールドタイプ**］（SQL名：seType）フィールドで、特殊フィールドのタイプを指定できます。このタイプによって、フィールドのプロパティ（表示、並べ替え、またはフィルタが使用可能どうか）が変わります。
- ［**結果のデータ型**］フィールドで、特殊フィールドの結果タイプを指定できます。このタイプによって、表示形式が変わります。結果タイプが日付型の特殊フィールドは、データベース内の他の「日付」型のフィールドと同じように表示されます。

フィールドの計算式を入力する

フィールドの計算式を入力します。画面下部のテキストフィールドに直接入力するか（表示されるフィールドは、選択する属性タイプによって変わります）、をクリックし、［F4］キーを押して、対応するエディタ画面にアクセスします。

注意:

使用する言語は、フィールドのタイプによって異なります。

計算式の作成に使用できる言語の詳細については、次のドキュメントを参照してください。

- このガイドの「**スクリプト**」の章。使用する関数は**RetVal()**です。
- AQL言語の場合は、このガイドの「**AQLクエリ**」の章。
- 特殊文字列の場合は、『**管理**』ガイドの「**データベースの標準記述ファイル**」の章、「**テーブルについて**」のセクション、「**テーブルのレコード名の表記文字列**」。

特殊フィールドのユーザ権限を定義する

［**管理／ユーザ権限**］メニューを選択します。ユーザ権限を作成する画面が表示されます。

注意:

特殊フィールドには、読取専用でのみアクセスできます。

1. ［**説明**］（SQL名：Description）フィールドにユーザ権限の短い説明を入力し、必要に応じて、［**コメント**］（SQL名：Comment）フィールドにコメントを入力します。
2. 特殊フィールドに関連するテーブルのツリー構造を展開します。（特殊フィールド）アイコンが付いたブランチに、そのテーブルの特殊フィールドの完全なリストが含まれます。
3. ユーザ権限を編集するフィールドを選択します。［**フィールド、リンク、任意管理項目**］枠の［**読み取り**］チェックボックスを使って、そのフィールドの読み取り権限を定義できます。このチェックボックスをオンにすると、このユーザ権限を持つプロファイルのユーザだけが、特殊フィールドを表示で

きます。このチェックボックスをオフにすると、すべてのユーザにこのフィールドに対する（読み取り専用）権限が与えられます。

特殊フィールドの使用方法

特殊フィールドの使用方法はタイプによって異なります。タイプが使用目的に合っていることを確認する必要があります。フィールドを表示するリスト（フィルタの作成、リストの設定など）では、ユーザに分かりやすいよう、使用可能なフィールドのみが表示されます。

リストの設定で特殊フィールドを使う

ショートカットメニューの［**リストの設定**］コマンドを使って、テーブルのすべてのレコード用の特殊フィールドの値を表示できます。

警告:

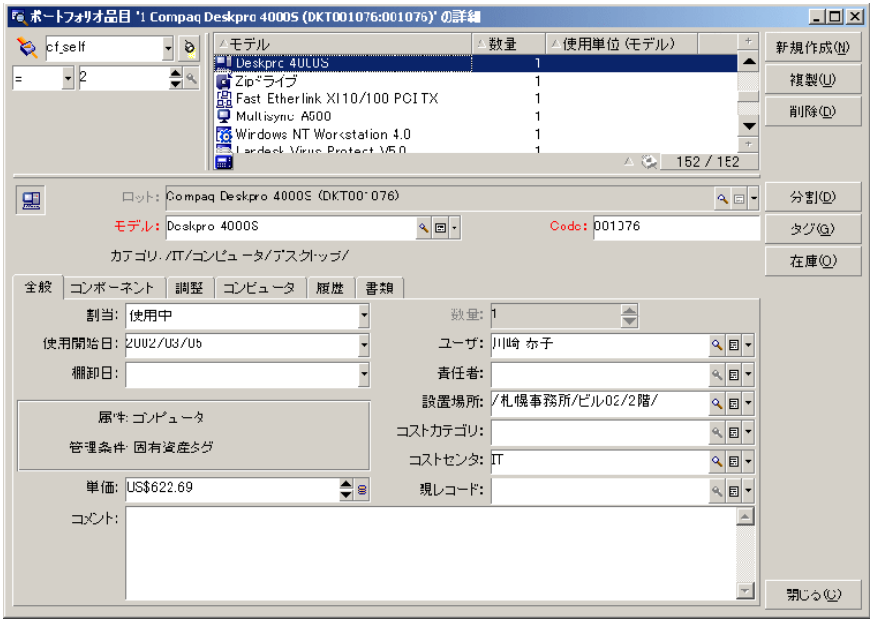
DBMSがMicrosoft SQLServerである場合、この操作は機能しません（特殊フィールドは表示されません）。

テーブルのレコードをフィルタを使って検索する

テーブルのレコードをフィルタを使って検索するAsset Managerでは、"AQL"タイプの特殊フィールドの値に応じて、テーブルのレコードを検索できます。これを行うには、ショートカットメニューから［**シンプルフィルタ**］コマンドを選択し、「**特殊フィールド**」ブランチに移動します。"AQL"タイプのフィールドだけが表示されます。

特殊フィールドを参照する

「cf_」という接頭語を持つSQL名を使って、特殊フィールドを参照できます。下の画面は、特殊フィールドのSQL名をフィルタに使って、レコードを検索した例です。



さらに、特殊フィールドは、Asset Managerの次のモジュールまたは機能でも使用できます。

- Asset Manager Web
- Asset Manager API
- レポート

これらのモジュールや機能については、対応するドキュメントを参照してください。

15 ウィザード

Asset Managerでは、独自のウィザードを作成したり、既存のウィザードを用途に合わせて変更したりすることができます。ウィザードは、テキストフィールドとして保存されます（［**ウィザードのスクリプト**］（SQL名：WizardScript）フィールド、［**ウィザード**］タイプアクションの詳細画面の［**ウィザード**］タブページ）。ウィザードを作成するには、このフィールドに直接コードを入力するか、グラフィカルエディタを使います。これには、ウィザードの構造とこの構造を記述するスクリプト言語をよく理解しておく必要があります。

警告:

スクリプトまたはウィザードタイプのアクションを変更する場合、Asset Manager Webを導入していた場合はWebサービスを再度タグ付けするようにしてください。

- **『カスタマイズ』ガイドの「データベースのカスタマイズ」の章、「開発のベストプラクティス／Webサービスのタグ付け」のセクション。**

表記法

ウィザードの構造の説明では、次の表記を使います。

表 15.1. 表記法

[ ]	大括弧は、データベース内のフィールドの値（コンテキストを指定するウィザードの場合）または「専用フィールド」のCurrentSelectionおよびCurrentTableのいずれかを参照する時に使います。大括弧はまた、オプションパラメータを示す際にも使用されます。
< >	山形括弧は、プロパティの短い説明を示します。実際にプロパティを入力するときは、山形括弧を使わずに、括弧内にあるテキストに該当する情報だけを入力してください。
|	縦線（パイプ）文字は、プロパティの値を区切るのに使います。また、複数の列からなるリストのタイトルと値を区切るときにも使います。
{ }	中括弧は、ノードの定義または複数行にわたるプロパティのスクリプトを囲むのに使います。また、ウィザードのプロパティの値を参照する時にも使います。
'	Basicコードの例のアポストロフィは、ウィザードの実行時にAsset Managerが無視するコメント行を示します。
; または //	ウィザード内のセミコロンまたは二重スラッシュは、Asset Managerが無視するコメント行を示します。

定義

ウィザードの構造の説明で使われている用語の定義を以下に示します。

Twip

Twipは、ウィザードで使われるサイズの単位で、デフォルトでは長さを表します。画面の解像度とは関係ありません。次の単位に相当します。

- 1440twipは1インチに相当します。
- 567twipは1cmに相当します。
- 96dpi解像度（Windowsの標準解像度）では、15twipは1ピクセルに相当します。

コントロール

コントロールは、データを編集する時に使うグラフィック要素です。一般的なコントロールには、チェックボックス、テキストエディタ、ボタン、ドロップダウンリストなどがあります。

ノード

ノードは、ウィザードのツリー構造の階層レベルに対応します。例えば、ノード「N」のサブノードは、ツリー構造内でノード「N」にリンクしている1レベル下のノードです。

注意:

ノード名には英数字のみを使用できます。ノード名の最大長は、22文字（半角の場合）です。

オブジェクト

オブジェクトは、次の要素を示す総称的な用語です。

- ウィザード全体
- ウィザードのページ
- ページ内のコントロール（チェックボックス、テキストエディタ、ボタン、ドロップダウンリストなど）
- 変数
- その他

オブジェクトと従属オブジェクト

オブジェクト「A」にオブジェクト「B」が含まれる場合

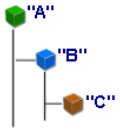
- オブジェクト「A」は、オブジェクト「B」の親オブジェクトになります。
- オブジェクト「B」は、オブジェクト「A」の「従属オブジェクト」になります。

警告:

この概念は構成上の関係だけで、データの継承とは関係ありません。

オブジェクトのフルネーム

オブジェクトのフルネームは、全ての親オブジェクト名とこのオブジェクトの名前で構成されます。各オブジェクトは、ピリオド（.）で区切ります。構造の例は以下のとおりです。



この例では、オブジェクト「C」のフルネームは、次のようになります。

<オブジェクトAの名前>.<オブジェクトBの名前>.<オブジェクトCの名前>

変数

変数は、ウィザードの実行中に変更可能なデータを格納する名前付きの保存領域です。各変数には、ウィザード内で個別に識別される名前があります。ウィザードで使用される変数は、すべてグローバル変数です。つまり、ウィザードのどのノードからも、これらのフルネームを使って参照できます。

Asset Managerのウィザードには2種類の変数があります。

- LONG（倍長整数）またはSTRING（文字列）型のノードで定義するウィザード変数。ノードのタイプによって変数のタイプが決まります。LONGノードで定義する変数は倍長整数、STRINGノードで定義する変数は文字列です。これらの変数は、定義上グローバルです。つまり、ウィザードのどのノードからも、これらのフルネームを使って参照できます。必要に応じて、これらの変数をAsset Managerで自動的に再計算することができます。
- ウィザード内のBasicスクリプトで使われるBasic変数。デフォルトでは、これらの変数はローカルですが、COMMONおよびGLOBALプロパティを使ってグローバルにすることができます。これらの変数は、Asset Managerによって自動的に再計算されることはありません。

遷移

遷移とは、ウィザードのあるページから別のページに移ることを示します。特定のページでいくつかの遷移を定義できます。それぞれの遷移には、遷移を有効にしてトリガするのに必要なユーザ定義の条件があります。

- 例えば、ユーザが［**次へ**］をクリックすると、最初に有効な遷移（条件が満たされた遷移）が実行されます。有効な遷移がない場合は、［**次へ**］ボタンは無効になります。
- ウィザードに必須プロパティがある場合、この必須プロパティが入力されていないと、［**次へ**］ボタンは使用不可能になります。
- ウィザードの全段階を完了せずに［**終了**］をクリックすると、指定されていない部分はデフォルト値になります。

構造モデル

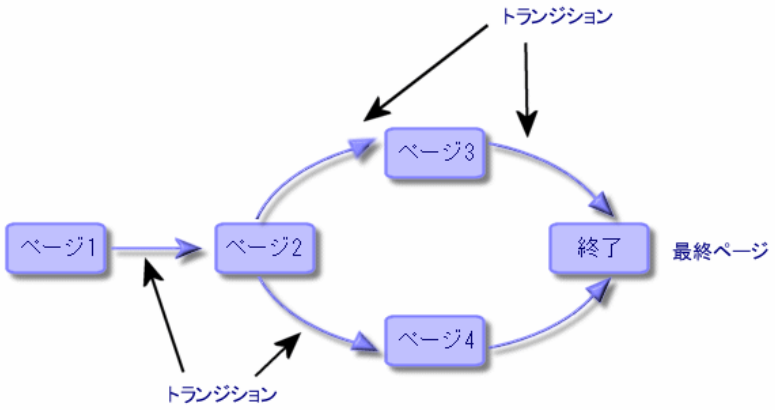
Asset Managerウィザードは、ウィザードを実行するタイプのウィザードです。ウィザードは、構造スクリプトで定義します。

- ウィザードスクリプト（ウィザード）はノードで構成されます。

- ウィザードの各ノードには、名前、1つまたは複数のサブノード、およびプロパティブロックがあります。ノードのタイプは次のとおりです。
  - ROOTノード（起点ノード）：このノードは固有で、他のすべてのサブノードを含みます。
  - STARTノード：このノードは固有で、ウィザードの起動時に実行されるスクリプトが含まれます。
  - PAGEノード：ウィザードのページ情報を記述します。
  - TRANSTIONノード：2つの「PAGE」タイプのノード間の遷移を記述します。
  - FINISHノード：このノードは固有で、ウィザードの最後に実行されるスクリプトが含まれます。
  - PARAMSノード：このノードは固有で、別のウィザードに渡すパラメータが含まれます。複数のウィザードを連続して（パラメータを受け渡す、受け渡さないに関係なく）実行できます。このようなウィザードのことを、連鎖しているといいます。
  - LONGノードまたはSTRINGノード：対応する変数のデータ型を定義します
- プロパティの値は定数またはBasicスクリプトを用いて明確に指定します。Basicスクリプトの場合、値はスクリプト評価の結果です。

ウィザードは、遷移でリンクされたページで構成されています。。1つのページから別のページへの分岐は、ユーザが入力する選択情報によって決まります。ウィザードの構造は次の図のとおりです。

図 15.1. ウィザードの構造



ウィザードページのモデル

ウィザードページは、次のように構成されています。

ウィザードに関する一般情報

ウィザードのコード（ウィザードを使うアクションの詳細画面の［**ウィザード**］タブページの、［amAction］テーブルの［**ウィザードのスクリプト**］（SQL名：WizardScript）フィールド）は、中括弧（{}）で区切られたブロックからなる構造化テキストです。このテキストで、ウィザードの構造を定義します。構造はツリー構造です。

ウィザードのツリーの各ノード（「ROOT」、「PAGE」など）には、サブノードとプロパティのセットを無制限にリンクできます。

一般的な構造とシンタックス

ノードの構造とシンタックスは次の通りです。

```
; これはスクリプト外のコメントです
{ <ノードタイプ> <ノード名>
<プロパティ名> = <プロパティの値>
```

```
' これはスクリプト内のコメント行です。
...
{ <プロパティ名> =
...
}
{ <ノードタイプ> <ノード名>
<プロパティ名> = <プロパティの値>
...
{ <プロパティ名> =
...
}
}
}
```

ノードは、次の規則に従って指定します。

- ノードの名前は付けなくてもかまいません。ノードに名前を付けない場合は、ノードに自動的に名前と番号が割り当てられます。
- ノード名にスペースを使うことはできません。
- ノードの名前が「=」の場合、これはノードではなく複数行プロパティとなります。複数行プロパティの詳細については、この章の「プロパティのシンタックス」を参照してください。
- スクリプト外のセミコロン（;）で始まる行と、スクリプト内のアポストロフィ（'）で始まる行はコメントとして解釈され、無視されます。

注意:

注意：中括弧（{）とノードタイプ間にはスペースを入れる必要があります。スペースがない場合、ウィザードは実行されません。

ノードのプロパティ

プロパティの値は、定数またはスクリプトを使って定義できます。定数は、数値、ブール値、またはテキストのいずれかです。

注意:

オブジェクトに関連するプロパティは、任意にすることも必須にすることもできます。プロパティは、「論理的」（オブジェクトの定義を補足）または「物理的」（オブジェクトの視覚面に影響）に定義できます。

宣言的モデル

プロパティは、循環参照（A={B}、B={A}）を定義する宣言モードに従って定義します。

```
<プロパティ名>=<スクリプト>
```

この定義には従属関係のリストが関連付けられます。

```
A={B}+{C}
```

この例では、プロパティ「A」はプロパティ「B」および「C」に依存します。つまり「A」に従属するリストは、「B」と「C」になります。

この結果、プロパティは次の場合に変化します。

- この従属関係のリスト内のプロパティの1つが変化した場合。
- プロパティまたは従属プロパティをユーザが変更した場合。

定数をプロパティの値として定義する

次のシンタックスでは、プロパティの定数値を定義します。

- テキスト型のプロパティ
  - <プロパティ名> = "<テキスト>"
- ブール型のプロパティ
  - <プロパティ名> = TRUE
  - <プロパティ名> = FALSE
  - <プロパティ名> (<プロパティ名> = TRUEに相当)
- 数値型のプロパティ
  - <プロパティ名> = 42
- <プロパティ名> = {<Basic変数またはプロパティのフルネーム>}

注意:

ブール値「TRUE」は、「0」以外の数値と同じです。「FALSE」は数値「0」と同じです。

プロパティを参照する

オブジェクトのプロパティを参照する（プロパティまたはオブジェクトの内容、特に値を参照する）には、次のシンタックスを使います。

```
{<プロパティのフルネーム>}
```

ページ「Page1」のプロパティ「Prop」を参照する場合は、次のように入力します。

```
{Page1.Prop}
```

このシンタックスでは、大文字と小文字は区別されません。

スクリプトをプロパティの値として定義する

スクリプトの概念

スクリプトは、グローバル変数RetVal内の値を返す1行または複数行のBasicプログラムです。1行スクリプトの場合、変数は暗黙的です。複数行スクリプトの場合は指定する必要があります。

すべてのBasicスクリプトについて、戻り値のデータ型に注意してください。このデータ型は、スクリプトを介して計算されるプロパティのタイプに依存します。

1行スクリプトのシンタックス

```
<プロパティ名>=<スクリプト>
```

例：

```
Variable="The name is: " & {Name}
```

上の1行スクリプトは、次の複数行スクリプトと同じです。

```
{ Variable =
RetVal="The name is: " & {Name}
}
```

複数行プロパティのシンタックス

```
{ <プロパティ名>=
<スクリプト>
}
```

例：

```
{ LABEL =
IF {Page1.Title}="Choose an employee" THEN
RetVal="Employee"
ELSE
RetVal="Department"
END IF
}
```

プロパティで適用できるメソッド

メソッドでは、プロパティまたはノードにリンクしている値を取得したり、このプロパティで関数を実行したりすることができます。この点で、メソッドは高度な関数と考えることができます。

メソッドのシンタックスは次の通りです。

```
{ノード.ノード.ノード[.プロパティ][.メソッド([arg1[, arg2[...]]])]}
```

以下の値を入力します。

- ノード：ノード名
- プロパティ：プロパティ名
- メソッド：メソッド名
- arg1、arg2、...：定数またはBasic式（Basic式で中括弧{ }を使用してはなりません）。

注意:

この例では、大括弧（[ ]）文字を使ってオプション項目を区切っています。

例えば、ページ「PAGE1」内の「LISTBOX」コントロールから行数を取得するには、このタイプのコントロールに関連するCOUNTメソッドを使います。コマンドは次の通りです。

```
{PAGE1.LISTBOX.VALUES.COUNT()}
```

テーブルタイプのプロパティ

テーブルタイプのプロパティは、次の形式に従って値が定義されるプロパティです。

```
<列|列|列|...>=<行の識別子>, <列|列|列|...>=<行の識別子>, ...
```

これらのプロパティの値は、次のように表形式で表示できます。

		列1	列2	列3
行番号：1	**行の識別子（例：18）**	セル（1,1）	セル（2,1）	セル（3,1）
行番号：2	**行の識別子（例：29）**	セル（1,2）	セル（2,2）	セル（3,2）
行番号：3	**行の識別子（例：78）**	セル（1,3）	セル（2,3）	セル（3,3）

		列1	列2	列3
...	...	...	...	...

> **注意:**
>
> 識別子は「テキスト」タイプです。

例

［部署と従業員］テーブルのクエリ結果を値として含むLISTBOXノードのVALUESというプロパティがあるとします。このクエリはこのテーブルの各レコード用に、［**部署名/姓**］（SQL名：Name）および［**名前**］（SQL名：FirstName）フィールドの値を返します。このプロパティには次の値が入っているとします。

```
VALUES="Colombo|Gerard=32,Lubeck|Alexander=64,Daquin|William=24"
```

この値を、次の表形式で表示できます。

		名前	名前
1	**32**	Colombo	Gerard
2	**64**	Lubeck	Alexander
3	**24**	Daquin	William

グローバル変数CurrentTableとCurrentSelectionを使う

これらの変数の内容は、次のシンタックスを使って取得できます。

```
[CurrentTable]
[CurrentSelection]
```

これら2つの変数の特徴は次の表の通りです。

変数名	変数の説明	コメント
CurrentTable	ウィザードの起動時にアクティブなテーブルのSQL名を含みます。アクティブなテーブルがない場合は空の文字列を含みます。 文字列型の変数	この変数は、Asset Managerにより自動的に入力されます。ユーザが値を手動で入力することはできません。

変数名	変数の説明	コメント
CurrentSelection	ウィザードの起動時に選択されたレコードの内部識別子の、カンマ区切りのリストを含みます。 文字列型の変数	この変数は、Asset Managerにより自動的に入力されます。選択項目がない場合またはテーブルが指定されていない場合、変数は空の文字列を含みます。ユーザが値を手動で入力することはできません。

ウィザードの連鎖化

実行後のウィザードは、別のウィザードの実行をトリガしたり、このウィザードにパラメータ（変数）を渡したりすることができます。このようなウィザードを連鎖ウィザードと呼びます。

実行

ウィザードAからウィザードBをトリガするには、FINISHノードにCHAINプロパティが必要です。このプロパティには、実行する［**ウィザード**］タイプアクションのSQL名の値（この場合は「B」）を設定する必要があります。

ヒント:

デフォルトでは、プライベートウィザードアクションの連鎖は、アクションの所有者（作成者）以外不可能です。このため、ユーザは、使用可能なウィザードから「非表示」のウィザードに連鎖（実行）できません。

プライベートアクションとは、［共有しない］（bPrivate）チェックボックスがオンになっているアクションです。この場合、所有者（作成者）以外のユーザに非表示になります。

他のユーザが使用可能なウィザードから「非表示」のウィザードに連鎖（実行）できるようにする場合、［管理／データベースオプション］メニューリンクを使用して、［Wizard／ChainNotOwnedPrivWizards］オプションを［はい］に設定する必要があります。

作成パラメータ

パラメータは、ウィザードAのPARAMSノードを使って、ウィザードBに渡されます。これらのプロパティは、ウィザードBのPARAMSノードのパラメータに追加されます。ウィザードAのPARAMSノードとウィザードBのPARAMSノードの両方で同じパラメータが定義されている場合は、ウィザードAのパラメータが優先されます。

Basic関数

Asset Managerの一般的な関数（AmCounter関数を除く）に加え、ウィザードでは次の関数を使用できます。

- AmComputeString()
- AmDecrementLogLevel()
- AmExecTransition()

  警告:

  このAPIはWebクライアントでは使用できません。

- AmLog()
- AmMsgBox()
- AmPagePath()
- AmProgress()
- AmRefreshProperties
- AmSetProperty
- AmUpdateDetail
- AmValueOf
- AmWizChain

関数が返す値

ウィザードスクリプトからBasic関数を呼び出す場合は、関数が返す値を常に変数に割り当てる必要があります。変数に割り当てないと、エラーが返されます。次の例はコンパイルされません。

```
AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

正しいスクリプトは次の通りです。

```
Dim lValue as Long
lValue=AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

Basicスクリプトでの文字列連結

ウィザードスクリプトは、Basicスクリプトを呼び出すことができます。

Basicスクリプトでテキスト文字列を連結するには、**+**演算子ではなく**&**演算子を使用します。

+演算子を使用すると、加法演算子として解釈されるため、ウィザード実行時にエラーが発生します。

ROOTノードの定義

ROOTノードでは、ウィザード全体を定義します。このノードは、ウィザード全体に適用できる一般的なプロパティブロックと、ウィザードに含まれるオブジェクトを表す一連のサブノードで構成されます。

ROOTノードのシンタックス

TRANSITIONノードのシンタックスは次の通りです。

```
' ROOTノードの一般的なプロパティブロック
NAME=...
IMAGE=...
...
' ROOTノードのサブノードの定義
{ FINISH
...
}
{ PAGE
...
}
{ TRANSITION
...
}
```

ROOTノードのプロパティ

ROOTノードで定義できるすべての論理／物理プロパティの一覧は、次の表の通りです。

表 15.2. ROOTノードの論理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
NAME="<ウィザード名>"	ウィザードの名前を定義します。 文字列型のプロパティ	NAME = "MoveWizard"	このプロパティに値を定義する必要があります。ウィザード名は22文字以内（半角の場合）で指定します。このプロパティは、ウィザードを連鎖化するために使われます。このウィザードに関連するデータは、この名前で保存されます。そのため、各ウィザードに異なる名前を付けることお奨めします。
TITLE=<"ウィンドウのタイトル>"	ウィザードのウィンドウのタイトルを定義します。 文字列型のプロパティ	TITLE="Move Wizard"	このプロパティにはできるだけ値を定義するようにしてください。
GLOBAL=<スクリプト>	ウィザードを連鎖化するか（=TRUE）しないか（=FALSE）を指定します。ウィザードを連鎖化する場合は、次のウィザードの実行時に前に入力された値が（「ini」ファイルに）保存されます。 このスクリプトは宣言スクリプトのため、実行されません。起動時にスクリプトを実行するには、START ノードを使用します。 ブール型のプロパティ	{GLOBAL=Dim Filter As String}	
COMMON	全ウィザード内に自動的に入るBasic関数が含まれます。 このプロパティは、データベースの作成時にデータベースに挿入される「gbbase.wiz」ファイル（読取専用）に相当します。		

プロパティ名=値	プロパティの説明	例	コメント
SERIALIZE=<TRUE|FALSE>	ウィザードを連鎖化するか（=TRUE）しないか（=FALSE）を指定します。ウィザードを連鎖化する場合は、次のウィザードの実行時に前に入力された値が（「.ini」ファイルに）保存されます。 NAMEプロパティは、「.ini」のどのセクションに値が格納されるかを決定します。 ブール型のプロパティ 「.ini」ファイルの場所： ▶ **『Asset Manager - インストールとアップグレード』**ガイドの**「.iniおよび.cfgファイル」**の章。	SERIALIZE=TRUE	デフォルトでは、このプロパティは「FALSE」に設定されます。
MODAL=<TRUE|FALSE>	ウィザードがモーダルか（=TRUE）否か（=FALSE）を定義します。		

表 15.3. ROOTノードの物理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
IMAGE="<ビットマップファイルのパス> "IMAGE16="<ビットマップファイルのパス>"	ウィザードで表示するビットマップタイプのグラフィックファイル（.bmp）を定義します。 文字列型のプロパティ	IMAGE="Wizard.bmp"	値を定義しない場合、画像は表示されません。画像ファイルのパスは、Asset Managerの「Config」フォルダを基準にします。 Asset Managerは、まず、データベース内で画像を検索します。 「IMAGE16」に値を定義すると、画面の色深度が16の場合、「IMAGE」でなくこのプロパティが使われます。
WIDTH=<幅>	ウィザードのウィンドウのデフォルト幅（<幅>）を定義します。値はtwipで指定します。 LONG（倍長整数）型のプロパティ	WIDTH=6000	
HEIGHT=<高さ>	ウィザードのウィンドウのデフォルトの高さ（<高さ>）を定義します。値はtwipで指定します。 LONG（倍長整数）型のプロパティ	HEIGHT=5000	
MINWIDTH=<MinWidth>	ウィザードのウィンドウの最小幅を定義します。 値は**twip**で指定します。		

プロパティ名=値	プロパティの説明	例	コメント
MINHEIGHT=<MinHeight>	ウィザードのウィンドウの最小の高さを定義します。 値は**twip**で指定します。		
CTRLHEIGHT=<CtrlHeight>	高さが固定しているコントロール（TEXTBOXコントロールなど）の高さを定義します。 間隔の値は**twip**で指定します。		
LABELSPACING=<labelSpacing>	コントロールのタイトルがコントロールの上にある場合、タイトルとコントロール間の間隔を定義します。 間隔の値は**twip**で指定します。		
CTRLSPACING=<CtrlSpacing>	2つのコントロール間の上下の間隔を定義します。 間隔の値は**twip**で指定します。		
IMGBORDER=<Width>	ウィザードの画像とコントロール間の左右の間隔を定義します。 値は**twip**で指定します。		
NAVIGATION=<TRUE|FALSE>	ウィザードウィンドウ内の［次へ］や［キャンセル］ボタンを含むナビゲーションバーを表示するか（=TRUE）しないかを（=FALSE）指定します。		

プロパティ名=値	プロパティの説明	例	コメント
CONFIRMCANCEL=<TRUE|FALSE>	キャンセルの確認メッセージを表示するか（=TRUE）しないか（=FALSE）を定義します。		
DEFAULTONNEXT=<TRUE|FALSE>	デフォルト（=TRUE）で［次へ］ボタンを選択します。 DEFAULTONNEXT=FALSEの場合、デフォルトで選択されるボタンは［終了］です。		

ROOTノードのサブノード

ROOTノードに定義できるサブノードのタイプを下の表に示します。各ノードタイプはオブジェクトを表します。

表 15.4. ROOTノードのサブノード

ノードタイプ	説明
PAGE	ウィザードのページ名。
FINISH	ウィザードの最終ページからの遷移名（終了）。この遷移タイプのノードには「FROM」および「TO」プロパティはありません。
START	例えば、ウィザードの起動時に（「DO」プロパティ）を使って実行するスクリプトおよびウィザードの開始ページ（「TO」プロパティ）名が含まれます。
PARAMS	FINISHサブノードのプロパティCHAINが入力されていると、ウィザードのパラメータを別のウィザードへ転送できます。
TIMER	ウィザードのページへタイマを関連付けることができます。

PAGEノードの定義

PAGEノードでは、ウィザードのページ名を定義します。このノードは、このノードに適用できるプロパティブロックと、すべてのサブノード（ページで定義されたオブジェクトを定義するサブノードのセット）で構成されます。

PAGEノードのシンタックス

PAGEノードのシンタックスは次の通りです。

```
' ページの宣言
{ Page <ページ名>
' PAGEノードのプロパティブロック
IMAGE=...
TITLE=...
' PAGEノードのサブノードの定義
{ TRANSITION
...
}
{ <コントロールのタイプ> <コントロール名>
...
}
...
}
```

PAGEノードのプロパティ

PAGEノードで定義できる全プロパティの一覧は、次の表の通りです。

表 15.5. PAGEノードの論理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
TITLE="<ページのタイトル>"	ページのタイトルを定義します。このタイトルは、ページ上部に太字で表示されます。 文字列型のプロパティ	TITLE="移動"	このプロパティに値を定義しない場合は、ROOTノードの「TITLE」プロパティの値が継承されます。ラベルと異なり、この文字列はHTMLをサポートしません。
ONENTER=<スクリプト>	［次へ］または［前へ］ボタンをクリックしてページにアクセスする際、実行されるBasicスクリプトを定義します。 ブール型のプロパティ	{ONENTER = AmMsgBox ("Hello") }	

表 15.6. PAGEノードの物理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
IMAGE="<ビットマップファイルのパス>" IMAGE16="<ビットマップファイルのパス>"	ウィザードのページで表示するビットマップタイプのグラフィックファイル（.bmp）を定義します。 ブール型のプロパティ	IMAGE =" C:\Images\Page1.bmp"	このプロパティに値を定義しない場合は、ROOTノードのIMAGEプロパティの値が継承されます。 このプロパティに空の値を定義すると、画像は表示されません。 「IMAGE16」に値を定義すると、画面の色深度が16の場合、「IMAGE」でなくこのプロパティが使われます。

PAGEノードのサブノード

PAGEノードには、2つのタイプのサブノードを定義できます。

表 15.7. PAGEノードのサブノード

ノードタイプ／オブジェクト	説明
<コントロールのタイプ> <コントロール名>	現在のページに表示するコントロールを定義します。
TRANSITION <遷移名>	ウィザードの現在のページと別のページ間の遷移名を定義します。
TIMER	ウィザードのページへタイマを関連付けることができます。

TRANSITIONノードの定義

TRANSITIONノードでは、ウィザード内の2つのページ間の移動を定義します。このノードは、プロパティのブロックだけで構成されます。

注意:

遷移はPAGEノード内から定義（この場合、「FROM」プロパティは必要ありません）またはROOTノード内から定義できます。ウィザードを閉じる最終遷移は、FINISHノード（ROOTノードの）で定義し、「FROM」および「TO」プロパティは必要ありません。

TRANSITIONノードのシンタックス

TRANSITIONノードのシンタックスは次の通りです。

```
' 遷移の宣言
{ TRANSITION0 <遷移名>
' TRANSITIONノードのプロパティブロック
FROM=...
TO=...
CONDITION=...
}
```

TRANSITIONノードのプロパティ

次の表に、TRANSITIONノードで定義できる全プロパティの一覧を表示します。

表 15.8. TRANSITIONノードの論理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
FROM="<遷移のリンク元のページの名前>"	遷移のリンク元のページ名を定義します。 文字列型のプロパティ	FROM="Page2"	このプロパティは、遷移がROOTノード内で定義されている場合は必須、PAGEノード、FINISHノードまたはSTARTノードで定義されている場合は適用されません。
TO="<ターゲットページの名前>"	遷移のターゲットページ名を定義します。 文字列型のプロパティ	TO="Page3"	このプロパティは、ROOTノードまたはPAGEノード内で定義されている場合は必須、FINISHノードで定義されている場合は適用されません。
CONDITION=<スクリプト>	遷移を起動するのに必要な条件を定義します。 ブール値を返すスクリプトタイプのプロパティ	CONDITION={Comment}="user"	このプロパティは、STARTノードでは使用できません。
DO=<スクリプト>	遷移時に実行されるスクリプトを定義します。 ブール型のプロパティ	{DO= Filter=""}	

TRANSITIONノードの特徴

TRANSITIONノードには、サブノードはありません。

ROOTノードで遷移を定義する理由

PAGEノードの外で遷移を定義すると、すべてのウィザードで再使用可能なページを作成できるため、スクリプトの作成を簡略化できます。

FINISHノードの定義

FINISHノードでは、ウィザードの最終ページにリンクする最終遷移を定義します。これは、「FROM」および「TO」プロパティを持たない特別なタイプのTRANSITIONノードです。その他の点では、FINISHノードのシンタックスとプロパティは、TRANSITIONノードと同じです。

FINISHノードに固有のCHAINプロパティを使うと、他のウィザードの実行をトリガできます。

表 15.9. FINISHノードの論理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
CHAIN=<実行するウィザードのSQL名>	現在のウィザードの終了時に実行するウィザードのSQL名を定義します。 このプロパティが入力されていないとウィザードは実行されません。 文字列型のプロパティ	CHAIN="Move"	
CONDITION=<スクリプト>	［終了］ボタンを有効にする条件を定義します。 ブール型のプロパティ		
DO=<スクリプト>	ウィザードの最後に実行するスクリプトを定義します。 ブール値を返すスクリプトタイプのプロパティ		

注意:

PARAMSノードを使うと、次のウィザードにパラメータを渡すことができます。

表 15.10. FINISHノードの物理プロパティ

プロパティ名=値	プロパティの説明	例

SUMMARY=<TRUE|FALSE>	ウィザード実行時に概要のページを表示するか（=TRUE）しないか（=FALSE）を定義します。 amLog関数とamProgress関数によりこのページに入力できるようになります。	
SHOWPROGRESSBAR=<TRUE|FALSE>	概要ページに進行状況バーを表示するか（=TRUE）しないか（=FALSE）を定義します。	
SHOWLOGLIST=<TRUE|FALSE>	概要ページに進行状況ログを表示するか（=TRUE）しないか（=FALSE）を定義します。	
LABEL="タイトル"	概要ページにタイトルを表示します。	
ISHTML=<TRUE|FALSE>	ラベルのテキストの属性を定義します。 ブール型のプロパティ	
TITLE="タイトル"	概要ページのタイトルを定義します。 概要ページのデフォルトのタイトルは、ROOTノードのタイトルです。	

STARTノードの定義

STARTノードでは、ウィザードの開始方法を定義します。これは、「FROM」または「CONDITION」プロパティを持たない特殊なタイプのTRANSITIONノードです。その他の点では、STARTノードのシンタックスとプロパティは、TRANSITIONノードと同じです。

表 15.11. STARTノードの論理プロパティ

プロパティ名=値	プロパティの説明	例
DO=<スクリプト>	起動時に実行されるスクリプトを定義します。 ブール値を返すスクリプトタイプのプロパティ	
TO="<開始ページ名>"	表示する最初のページの名前を定義します。 文字列型のプロパティ	

> **注意:**
> このノードが存在しない場合、ウィザードは最初のページから起動します。

TIMERノードの定義

TIMERノードは、定期的にタスクを実行できるようにします。

表 15.12. TIMERノードの論理プロパティ

プロパティ名=値	プロパティの説明	例
AUTO= <TRUE|FALSE>	ページの表示時にタイマが自動的に起動されるかどうかを指定します。 ブール型のプロパティ	
ENABLED= <TRUE|FALSE>	タイマをアクティブにするか（=TRUE）しないか（=FALSE）を定義します。 このプロパティは、タイマを停止または再起動するためにも使用されます。 ブール型のプロパティ	
INTERVAL=間隔	タイマを実行する時間の間隔を定義します。 間隔はミリ秒単位で指定します。	
TIMER="Script"	タイマの間隔（"INTERVAL"）ごとに実行される内容を定義します。 スクリプトタイプのプロパティ	
VALUE=tickcount	時間の間隔が経過した回数。間隔に依存するプロパティは全て、定期的な間隔で再評価されます。 tickcountの値は自動的に増加します。	

倍長整数および文字列ノードの定義

LONG（倍長整数）およびSTRING（文字列）ノードでは、変数を定義します。変数はウィザードのすべてのノードで参照できます。ノード名によって、変数名が決まります。

これらのノードには1つのプロパティしかなく、そのデータ型はノードに依存します。データ型は倍長整数ノードの場合は、倍長整数、文字列ノードの場合は、文字列です。この「VALUE」プロパティを使って、変数の値を定義できます。

表 15.13. 倍長整数および文字列ノードの論理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
VALUE=<値>	ノード名を名前に持つ変数の値を定義します。 倍長整数の場合は倍長整数型、文字列ノードの場合は文字列型	VALUE=12	

注意:

倍長整数および文字列ノードは、ウィザードのどのノードでも定義できます。サブノードはありません。

コントロールノードの定義

ページのコントロールを使ってユーザは情報を指定できます。1ページ内に好きな数だけコントロールを定義できます。ページ内でのコントロールの編成は、Asset Managerが自動的に管理します。定義した各コントロールの位置を指定する必要はありません。

コントロールタイプのノードは、定義したコントロールに適用可能なプロパティブロックだけで構成されます。

コントロールノードの一般的なシンタックス

コントロールタイプのノードの一般的なシンタックスは次の通りです。

```
' コントロールの宣言
{ <コントロールのタイプ> <コントロール名>
' コントロールのプロパティ
...
}
```

コントロールのタイプと関連プロパティ

すべてのコントロールには、共通のプロパティがあります。一方、特定のコントロールに固有のプロパティもあります。

共通のプロパティ

すべてのコントロールに適用できるオプションのプロパティは、次の表の通りです。

表 15.14. すべてのコントロールに共通の論理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
MANDATORY=<TRUE|FALSE>	遷移を確認するために、ユーザにコントロールへの入力を強制します。 以下の場合このプロパティは無視されます。 ■ コントロールが表示されていない。 ■ コントロールは読取専用である。 ■ コントロールが無効になっている。	MANDATORY=TRUE	このプロパティはCHECKBOXおよびLABELコントロールには使用できません。

プロパティ名=値	プロパティの説明	例	コメント
VALUE="<値, 値,...>"	作成時にコントロールのデフォルト値を定義します。ウィザードでコントロールが表示されるとき、この値はユーザが選択した値で置き換えられます。コントロールに応じて、VALUEプロパティには複数の値を含めることができます。 このプロパティのタイプはコントロールタイプによって変化します（ブール、テキスト、など）。	■ 「チェックボックス」コントロールの場合、<Value>は"TRUE"または"FALSE"のいずれかをとることができます。 ■ VALUE="1,3,4"がである「リストボックス」コントロールの場合、1、3、4は、コントロールで選択された行に対応する値です。	VALUEプロパティに複数の値を含めることができる場合、コントロールのMULTISELプロパティをアクティブにする必要があります（TRUEにする）。このようにしないと、VALUEプロパティの最初の位置で定義された値のみが使用されます。
PERMANENT=<TRUE|FALSE>	ウィザードのあるページから別のページへ移る時、コントロールは削除されます。 ウィザードのあるページから別のページへ移る時に、コントロールを保存し隠すか（=TRUE）否か（=FALSE）を指定します。 ブール型のプロパティ		
SERIALIZE=<TRUE|FALSE>	このプロパティでは、コントロールに関してウィザードをシリアル化するかしないかを指定できます。ROOTノードのシリアル化が有効になっている場合（=TRUE）、コントロールについてはシリアル化を無効にすることができます。		このプロパティは、デフォルトでROOTノードのSERIALIZEプロパティの値を含みます。
HELP="ヘルプ"	このプロパティにより、HTMLフォーマットのヘルプ用テキストをウィザードのコントロール内に含むことができます。 ［Shifit+F1］キーを押してヘルプを表示します。		

表 **15.15.** すべてのコントロールに共通の物理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
VISIBLE=<TRUE|FALSE>	コントロールを表示するか（=TRUE）、しないか（=FALSE）を定義します。 ブール型のプロパティ	Label1.Visible=TRUE	
ENABLED = <TRUE|FALSE>	コントロールをアクティブにするか（=TRUE）、しないか（=FALSE）を定義します。ブール型のプロパティ ブール型のプロパティ	Choice1.Enabled=FALSE	
READONLY = <TRUE|FALSE>	コントロールの値を読み取り専用にして（=TRUE）、ユーザが変更できないようにするか、変更できるようにするか（=FALSE）を定義します。	READONLY=TRUE	
LABEL = "<ラベルのテキスト>"	コントロールの上または左に表示される任意のテキストを定義します。 文字列型のプロパティ	Choice1.Label="ユーザの選択"	このラベルはHTMLをサポートします。
LABELLEFT	このプロパティは、コントロールのラベルをコントロールの左側に置きます。 このプロパティを使用する場合は、XOFFSETプロパティに入力する必要があります。 ブール型のプロパティ		
XOFFSET	LABELLEFTプロパティを使用して、コントロールのラベルをコントロール左側に置く場合、ラベル用に確保される場所を定義します。 **twip**型のプロパティ		

プロパティ名=値	プロパティの説明	例	コメント
ISHTML	ラベルのテキストの属性を定義します。 デフォルトのテキストの属性はHTMLです。 ブール型のプロパティ		このラベルはHTMLをサポートします。
INDENT	このプロパティは、コントロールとラベルを右側に移動させます。 **twip**型のプロパティ		
YOFFSET	コントロールとラベルの前の「オフセット」を定義します。 **twip**型のプロパティ		
YOFFSET2	コントロールとラベルの後の「オフセット」を定義します。 **twip**型のプロパティ		

CHECKBOXコントロール

CHECKBOXコントロールでは、チェックボックスを定義します。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、CHECKBOXコントロールでは、次のプロパティが認識されます。

表 15.16. CHECKBOXコントロールのプロパティ

プロパティ名=値	プロパティの説明	例
CAPTION="<テキスト>"	チェックボックスのテキストを定義します このテキストはHTMLではありません。1行のみを含まなければなりません。 文字列型のプロパティ	TEXT="名前で識別"

COMBOBOXコントロール

COMBOBOXコントロールでは、定義済みの値リスト（リストデータ）の1つの選択肢を定義します。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、COMBOBOXコントロールでは、次のプロパティが認識されます。

表 15.17. COMBOBOXコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
VALUES="<名前=値, 名前=値, 名前=値,...>"	Comboコントロールの値の組合せ（「名前」=「値」）を定義します。「名前」にはコントロールに表示するテキストを定義します。「値」には、ユーザがこの「名前」を選択した場合に割り当てる値を定義します。 文字列型のプロパティ	VALUES="Table of assets=asset, User=user"	値を省略すると、「値」が自動的に割り当てられます。 **例** VALUES=A,B,C と定義すると、VALUES=A=1,B=2,C=3 と値が割り当てられます。

OPTIONBUTTONSコントロール

OPTIONBUTTONSコントロールでは、オプションボタン（ラジオボタン）のグループを定義します。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、OPTIONBUTTONSコントロールでは、次のプロパティが認識されます。

表 15.18. OPTIONBUTTONSコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例
VALUES="<タイトル=値, タイトル=値, タイトル=値,....>"	CHOICEコントロールの値の組み合わせ（「タイトル」=「値」）を定義します。「名前」にはオプションボタンのテキストを定義します。「値」には、ユーザがこのオプションボタンを選択した場合にコントロールに割り当てる値を定義します。 文字列型のプロパティ	VALUES="Table of assets=asset, User=user"
BORDER=<TRUE|FALSE>	オプションボタンのグループに境界を引くか（=TRUE）、引かないか（=FALSE）を指定します。 ボタンのグループが囲まれていると、テキストが枠の端に統合されます。このテキストはHTMLおよび多言語であってはなりません。 ブール型のプロパティ	BORDER= TRUE

LISTBOXコントロール

LISTBOXコントロールでは、選択可能なオブジェクトのリストを定義します。LISTBOXコントロールは、複数の列にすることができます。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、LISTBOXコントロールでは、次のプロパティが認識されます。

表 15.19. LISTBOXコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
LISTHEIGHT = <パーセント>	LISTBOXコントロールのサイズを、ウィザード全体に使われている他のLISTBOXコントロールを基準して定義します。 LONG（倍長整数）型のプロパティ	LISTHEIGHT=50	このプロパティの値がそれぞれ「10」と「20」の2つのLISTBOXコントロールがある場合、2番目のコントロールの高さは最初のコントロールの2倍になります。

プロパティ名=値	プロパティの説明	例	コメント
MULTISEL = <TRUE|FALSE>	コントロールで複数選択をサポートする（=TRUE）かしない（=FALSE）かを指定します。 ブール型のプロパティ	MULTISEL=TRUE	
COLTITLE = "<列|列|列...>"	リスト内の列のタイトルを定義します。「列」には列のテキストを定義します。 文字列型のプロパティ	COLTITLE = "姓|名前"	
COLWIDTH = "<幅|幅|幅...>"	コントロール全体のサイズに比例した列のサイズを定義します。 文字列型のプロパティ	COLWIDTH = "50|50"	
VALUES = "<テキスト|テキスト|...= 値, テキスト|テキスト|...= 値,...>"	LISTBOXコントロールの値の組み合わせ（テキスト|テキスト|...=値）を定義します。「テキスト|テキスト|..」には、LISTBOXコントロールの行の各列に表示するテキストを定義します。「値」には、ユーザがこの行を選択した場合にこのコントロールに割り当てられる値を定義します。 文字列型のプロパティ	VALUES="Table of assets=asset, , User=user,"	値を省略すると、「値」が自動的に割り当てられます。 VALUES=A,B,Cと定義すると、VALUES=A=1,B=2,C=3と値が割り当てられます。 **例** このプロパティには、直接入力したり、次のようにAmdbGetList関数を使って入力することもできます。 VALUES = AmDbGetList ("SELECT Name, FirstName FROM amEmplDept WHERE Name Like 'A%'", "|", ",", "=") 「VALUES」プロパティを「VALUE」プロパティ混同しないでください。
EDITABLE="<0|1>"	列のテキストが編集可能か（=TRUE）、不可能か（=FALSE）を定義します。 文字列型のプロパティ	EDITABLE="0|1"	

プロパティ名=値	プロパティの説明	例	コメント
TABLE="<テーブル名>"	列のタイトルがCOLNAMEプロパティで定義されている場合、列のタイトルの適用コンテキストを定義します。	TABLE="amEmplDept"	
COLNAME="<名前|名前|名前>"	リストの列のプロパティとタイトルを、フィールドのSQL名と共に定義します。TABLEプロパティには値を入力する必要があります。 列のタイトル用に「名前」をSQLで置き換えます。 列のタイトルにCOLTITLEプロパティで値が入力されている場合、このプロパティはCOLNAMEプロパティよりも優先されますが、SQLフィールドタイプ（テキスト, 日付など）を保持します。	COLNAME="Name|FirstName|Title"	警告: COLNAMEプロパティには、Webクライアントで適切に表示されるように、LISTBOXコントロールの内容に対して入力する必要があります。
MULTISEL = <TRUE|FALSE>	リスト用に複数選択の使用を定義します。 ブール型のプロパティ	MULTISEL=1	

表 15.20. LISTBOXコントロールのメソッド

プロパティ名=値	プロパティの説明	例	コメント
VALUES (i)	（i）行目の内容を返します。	a|b|c	
VALUES.COUNT()	VALUESプロパティ内の行数を計算します。	retval = {listbox1.values.count()}	
VALUES.CELL(h,v)	座標（水平座標,垂直座標）で指定されたセルの内容を返します。	VALUES.CELL(2,4)	
VALUES.COLUMN(i)	列（i）の内容（値）を返します。 i=0の場合または空の場合、この命令は列の識別子（ID）を返します。	VALUES.COLUMN(1)	

プロパティ名=値	プロパティの説明	例	コメント
VALUES.LINE(i)	行（i）の内容（値）を返します。 i=0の場合または空の場合、この命令は行の識別子（ID）を返します。	VALUES.LINE(1)	
VALUES.SORT(iCol, bAsc)	列（i）を昇順または降順（bAsc=1または bAsc=0）で並べ替えます。	{ LISTBOX lb VALUES = "第1,第2,第3" } { COMMANDBUTTON btn { CLICK = RetVal = {lb.Values.Sort(1)} } }	
VALUES (i,0)	行（i）のID値を返します。		

表 15.21. LISTBOXコントロールの必須論理プロパティ

プロパティ名=値	プロパティの説明	例
TABLE=<テーブル名>	列のタイトルの抽出に使うテーブル名。 文字列型のプロパティ	TABLE=**amAsset**
COLNAME=<タイトル|タイトル|...>	TABLEプロパティを使って定義したテーブルのフィールドのSQL名を使って、列のタイトルを定義します。このプロパティを使って、使用する編集コントロールも定義できます。コントロールは、Asset Managerでフィールドの入力に使うものと同じです。 列のタイトルを定義する場合、Asset ManagerはまずCOLTITLEプロパティ（ある場合）の値を取ります。 文字列型のプロパティ	COLNAME="Name|FirstName"

LABELコントロール

LABELコントロールでは、ラベルを定義します。このコントロールには次のプロパティがあります。

表 15.22. LABELコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
CAPTION=<テキスト>	ラベル内に表示されるテキストが含まれます。	CAPTION="場所の選択"	

PROGRESSBARコントロール

PROGRESSBARコントロールでは、進行状況バーを定義します。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、PROGRESSBARコントロールでは、次のプロパティが認識されます。

表 15.23. PROGRESSBARコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例
MAXVALUE=<最大値>	進行状況バーの100％に対応する最大値を定義します。 「VALUE」プロパティは、コントロールの現在の値を示します。 LONG（倍長整数）型のプロパティ	MAXVALUE=200

COMMANDBUTTONコントロール

COMMANDBUTTONコントロールでは、アクションボタンを定義します。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、COMMANDBUTTONコントロールでは、次のプロパティが認識されます。

表 15.24. COMMANDBUTTONコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例
WIDTH=<幅>	ボタンの幅をtwipで定義します。 LONG（倍長整数）型のプロパティ	WIDTH=250

プロパティ名=値	プロパティの説明	例
HEIGHT=<高さ>	ボタンの高さをtwipで定義します。 LONG（倍長整数）型のプロパティ	HEIGHT=125
CAPTION=<テキスト>	ボタン内に表示するテキスト（非HTML）を定義します。 文字列型のプロパティ	CAPTION="開始"
CLICK=<Basicスクリプト>	ユーザがこのボタンをクリックしたときに実行されるBasicスクリプトを定義します。	

DBLISTBOXコントロール

DBLISTBOXでは、データベースから選択できるレコードのリストを定義します。このコントロールは、複数の列にすることができます。コントロールに表示されるリストは、Asset Managerデータベースで部分AQLクエリ（WHERE句のみ使用）を実行した結果です。

注意:

「VALUE」プロパティは、選択した行の識別子（Id）のリストを返します。リストのセルの値にアクセスすることはできません。セル値にアクセスするには、別のクエリを実行するか、LISTBOXタイプのコントロールを使う必要があります。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、DBLISTBOXコントロールでは、次のプロパティが認識されます。

表 15.25. DBLISTBOXコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
TABLE="<テーブルのSQL名>"	クエリを行うテーブルを定義します。 文字列型のプロパティ	TABLE=amAsset	必須プロパティ
COLNAME="<フィールドまたはリンクのSQL名|フィールドまたはリンクのSQL名|...>"	データベースから抽出するデータ項目を定義します（SQL名で識別）。 文字列型のプロパティ	COLNAME = "Name|FirstName"	

プロパティ名=値	プロパティの説明	例	コメント
COLWIDTH = "<幅|幅|幅|...>"	データベースの列の幅を、DBLISTBOXコントロール全体に占めるパーセントで定義します。 文字列型のプロパティ	COLWIDTH="40|60"	
LISTHEIGHT = <パーセント>	DBLISTBOXコントロールのサイズを、ウィザード全体の他のDBLISTBOXコントロールを基準にして定義します。 LONG（倍長整数）型のプロパティ	LISTHEIGHT=50	このプロパティの値がそれぞれ「10」と「20」の2つのDBLISTBOXコントロールがある場合、2番目のコントロールの高さは最初のコントロールの2倍になります。
TREE=<TRUE|FALSE>	データをツリーで表示（=TRUE）するか、表示しない（=FALSE）かを指定します。 ブール型のプロパティ	TREE=TRUE	デフォルトでは、このプロパティは「FALSE」に設定されます。
MULTISEL = <TRUE|FALSE>	コントロールで複数選択をサポートする（=TRUE）かしない（=FALSE）かを指定します。 ブール型のプロパティ	MULTISEL=TRUE	
DBLCLICK = <TRUE|FALSE>	このプロパティをTRUEに設定する場合、Asset Manager行上をダブルクリックすると、現在のページの**「次へ」**ボタンのクリックがシミュレートされます。	DBLCLICK=FALSE	
FILTER= "<条件>"	クエリで処理するレコードをフィルタで検索するAQLの「WHERE」条件を定義します。 文字列型のプロパティ	FILTER = "UserlEmplDeptId=Colombo, Gerard' "	

プロパティ名=値	プロパティの説明	例	コメント
MAXSEL = <TRUE|FALSE>	99よりも多くの選択できるか（=TRUE）できないか（=FALSE）を定義します。 デフォルトで（=TRUE）選択は99要素に制限されています。		
SORT(iCol, bAsc)	列（iCol）を昇順または降順（bAsc=1またはbAsc=0）で並べ替えます。	SORT(2, 0)	
[Value.]ISSELECTION()	選択が0以外の値を含む場合、つまりユーザがリストから何も選択しない場合、このメソッドは非NULL値を返します。	MANDATORY = not {dblistbox1.IsSelection()} （ユーザがリストから何も選択しないと、遷移は不可能になります。）	
TABLELABEL(iNameType)	このメソッドはコントロールが定義されるテーブルのタイトルを返します。 タイトルタイプ（iNameType）は以下の通りです。 ■ 1 - システム名 ■ 2 - SQL名 ■ 3 - デフォルトのタイトル ■ 4 - 説明（ヘルプ用テキスト） TABLEプロパティには値が入力されなければなりません。	RetVal = {dblistbox1.table.label(2)}	

DBQUERYBOXコントロール

DBQUERYBOXコントロールでは、選択可能なレコードのリストを定義します。このコントロールは、複数の列にすることができます。コントロールに表示されるリストは、Asset Managerデータベースの完全AQLクエリ（WHERE句のみ使用）を実行した結果です。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、DBQUERYBOXコントロールでは、次のプロパティが認識されます。

表 15.26. DBQUERYBOXコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
QUERY="<完全AQLクエリ>"	DBLISTコントロールに表示する情報を返すAQLクエリを定義します。 文字列型のプロパティ	QUERY="SELECT Name, FirstName FROM amEmplDept WHERE Location='大阪ビル'"	
COLTITLE="<列|列|...>"	リストの列のタイトルを定義します。 文字列型のプロパティ	COLTITLE = "Name|FirstName"	
COLWIDTH="<幅|幅|...>"	リストの列のサイズを、コントロール全体のサイズに占めるパーセントで定義します。 文字列型のプロパティ	COLWIDTH = "50|50"	
LISTHEIGHT=<パーセント>	DBQUERYBOXコントロールのサイズを、ウィザード全体の他のDBQUERYBOXコントロールを基準にして定義します。 LONG（倍長整数）型のプロパティ	LISTHEIGHT=50	このプロパティの値がそれぞれ「10」と「20」の2つのDBQUERYBOXコントロールがある場合、2番目のコントロールの高さは最初のコントロールの2倍になります。
TREE=<TRUE|FALSE>	データをツリーで表示（=TRUE）するか、表示しない（=FALSE）かを指定します。 ブール型のプロパティ	TREE=TRUE	デフォルトでは、このプロパティは「FALSE」に設定されます。

プロパティ名=値	プロパティの説明	例	コメント
MAXSEL = <TRUE | FALSE>	99よりも多くの選択できるか（=TRUE）できないか（=FALSE）を定義します。 デフォルトで（=TRUE）選択は99要素に制限されています。		
MULTISEL=<TRUE | FALSE>	コントロールで複数選択をサポートする（=TRUE）かしない（=FALSE）かを指定します。 ブール型のプロパティ	MULTISEL=TRUE	
DBLCLICK=<TRUE | FALSE>	このプロパティをTRUEに設定する場合、Asset Manager行上をダブルクリックすると、現在のページの「次へ」ボタンのクリックがシミュレートされます。 ブール型のプロパティ	DBLCLICK=FALSE	
[Value.]ISSELECTION()	選択が0以外の値を含む場合、つまりユーザがリストから何も選択しない場合、このメソッドは非NULL値を返します。		

コントロールのクエリで返されるレコードの最大数を変更するには

デフォルトで、戻されるレコードの最大数は**200**に設定されています。

これにより、返される可能性があるレコード数が多い場合に、Webクライアントのパフォーマンスが悪いインパクトを受けるのを防ぎます。

レコードの最大数に達すると、ユーザに対するメッセージが表示されます。

デフォルトの値を変更するには：

1 各Asset Manager Web Serviceサーバ上にある「aamapi52.ini」ファイルを編集します。
2 **[Option]**セクションで、次の行を変更、または追加します。

```
/ACWeb/WizQueryBoxMaxLoad=1|<レコードの最大数>
```

DBEDITコントロール

DBEDITコントロールでは、Asset Managerデータベースのフィールドに入力するのに使ったのと同じコントロールが作成されます。コントロールは、各フィールドのデータ型（日付、金額など）によって異なります。

注意:

このコントロールの拡大鏡ボタンを使って、データベースから有効な値を選択できますが、別の値を入力することもできます。

このコントロールでは、「VALUE」プロパティは「可変」です（コントロールによって異なります）。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、DBEDITコントロールでは、次のプロパティが認識されます。

表 15.27. 標準モードのDBEDITコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例
TABLE="<テーブルのSQL名>"	開始テーブルのSQL名 文字列型のプロパティ	TABLE="amAsset"
FIELD=<フィールドのSQL名>	コントロールに使用されるフィールドのAQL名。 文字列型のプロパティ	FIELD="seAcquMethod"

プロパティ名=値	プロパティの説明	例
TABLE.LABEL([iNameType])	このメソッドはテーブルのタイトルを返します。 タイトルタイプ（iNameType）は以下の通りです。 ■ 1 - システム名 ■ 2 - SQL名 ■ 3 - デフォルトのタイトル ■ 4 - 説明（ヘルプ用テキスト） TABLEプロパティとNAMEプロパティには値が入力されなければなりません。	▶ DBQUERYBOXコントロールの物理プロパティ [献 307]
FIELD.LABEL([iNameType])	このメソッドはフィールドのタイトルを返します。 タイトルタイプ（iNameType）は以下の通りです。 ■ 1 - システム名 ■ 2 - SQL名 ■ 3 - デフォルトのタイトル ■ 4 - 説明（ヘルプ用テキスト） TABLEプロパティには値が入力されなければなりません。	

DBTABLEコントロール

DBTABLEコントロールは、Asset Managerデータベースのテーブルに入力するためのコントロールを作成します。

プロパティ

このコントロールに固有のプロパティはありません。

DBPATHコントロール

DBPATHは、Asset Managerデータベースのパスを入力するためのコントロールを作成します。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、DBPATHコントロールでは、次のプロパティが認識されます。

表 15.28. DBPATHコントロールの必須論理プロパティ

プロパティ名=値	プロパティの説明	例
TABLE="<テーブルのSQL名>"	パスを選択する際の選択元のテーブル名。 文字列型のプロパティ	TABLE=**amAsset**

LINKEDITコントロール

LINKEDITコントロールは、Asset Managerデータベースにリンクを入力するためのコントロールを作成します。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、LINKEDITコントロールでは、次のプロパティが認識されます。

表 15.29. LINKEDITコントロールの論理プロパティ

プロパティ名=値	プロパティの説明	例／コメント
TABLE=<テーブルのSQL名>	リンクを選択するテーブル名。 文字列型のプロパティ	TABLE="amAsset"
FILTER=<AQLクエリのWHERE句>	AQLフィルタを定義します。 文字列型のプロパティ	このプロパティはオプションです。
LINK=<リンクのSQL名>"	TABLEプロパティ内で定義されるテーブルリンクのSQL名。 オプションプロパティ	LINK="POrLIne"
ZOOM=<TRUE|FALSE>	拡大鏡ツールを表示するか（=TRUE）しないか（=FALSE）を定義します。 ウィザードがモーダルでない（ROOTノードでプロパティMODAL=FALSE）場合のみこのプロパティが適用されます。	
SRCCHOICE=<TRUE|FALSE>	回 アイコンを表示するか（=TRUE）しないか（=FALSE）を定義します。 ウィザードがモーダルでない（ROOTノードでプロパティMODAL=FALSE）場合のみこのプロパティが適用されます。	

プロパティ名=値	プロパティの説明	例／コメント
TABLE.LABEL([iNameType])	このメソッドは、リンクのソーステーブルのタイトルを返します。 タイトルタイプ（iNameType）は以下の通りです。 ■ 1 - システム名 ■ 2 - SQL名 ■ 3 - デフォルトのタイトル ■ 4 - 説明（ヘルプ用テキスト） TABLEプロパティには値が入力されなければなりません。	▶ DBQUERYBOXコントロールの物理プロパティ [献 307]
LINK.LABEL([iNameType])	このメソッドはリンクのタイトルを返します。 タイトルタイプ（iNameType）は以下の通りです。 ■ 1 - システム名 ■ 2 - SQL名 ■ 3 - デフォルトのタイトル ■ 4 - 説明（ヘルプ用テキスト） TABLEプロパティとLINKプロパティには値が入力されていなければなりません。	

TEXTBOXコントロール

TEXTBOXコントロールは、テキストを入力するためのコントロールを作成します。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、TEXTBOXコントロールでは、次のプロパティが認識されます。

表 15.30. TEXTBOXコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例
MULTILINE=<数値>	TEXTBOXコントロールが1行の場合、このプロパティは「0」に、複数行の場合は、コントロールの表示高さのパーセント値になります。	MULTILINE=50
PASSWORD=<TRUE|FALSE>	タイプされたテキストをマスクするか（=TRUE）しないか（=FALSE）を定義します。	

CHARTコントロール

CHARTコントロールではグラフを表示できます。グラフを複数の系列で構成することも可能です。

プロパティ

すべてのコントロールに共通のオプションのプロパティに加え、CHARTコントロールでは、次のプロパティが認識されます。

表 15.31. CHARTコントロールの論理プロパティ

プロパティ名=値	プロパティの説明	例
SERIES	縦線（パイプ文字）で区切られた系列のリストを定義します。 このリストは空であってはなりません。このリストなしではグラフは表示されません。	SERIES="購入価格|販売価格"
VALUES	グラフの系列の数値を定義します。 2次元の図表	VALUES="1|2,1|4"
FORMAT	データ型を定義します。 ■ 倍長整数（long） ■ 倍精度浮動小数点型（double） ■ 数値（number） ■ パーセント（percent）	
SERIE	クリックした系列の番号 CHARTプロパティは相互作用モード（=TRUE）でなければなりません。	
INDEX	クリックした列の番号 CHARTプロパティは相互作用モード（=TRUE）でなければなりません。	

プロパティ名=値	プロパティの説明	例
CLICK	図表上でクリックする際にこのプロパティのスクリプトを呼び出します。	

表 15.32. CHARTコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例
MODE	図表のタイプを定義します。 ■ MODE=0：縦の棒グラフ ■ MODE=1：横の棒グラフ ■ MODE=2：セグメント ■ MODE=3：円グラフ	
LABELS	列のタイトルを定義します。	1月|2月
3D	グラフを3Dにするか（=TRUE）しないか（=FALSE）を定義します。	
COLORS	各系列用の色のリストを定義します。リストの要素は縦線（パイプ文字）で区切られます。 RGB値は十進法です。	255|16777215|16711680 赤、白、青色を表示します。
INTERACTIVE	グラフが相互作用するか（=TRUE）しないか（=FALSE）、つまりグラフがマウスのポインタの変化に応じて起動するかどうかを定義します。	
POPUP	ショートカットメニューを表示するか（=TRUE）しないか（=FALSE）を定義します。	
BACKGROUND	グラフに背景を表示するか（=TRUE）しないか（=FALSE）を定義します。	
BACKIMAGE	グラフの背景になる画像のパスを定義します。 背景を表示するには、BACKIMAGEプロパティが有効（=TRUE）でなければなりません。	
STACKED	積み上げ棒グラフにするか（=TRUE）しないか（=FALSE）を定義します。	
CHARTHEIGHT	CHARTコントロールの相対的なサイズを、ウィザード全体の他のコントロールを基準にして定義します。	
CAPTION	タイトルを表示します。	
ELEVATION	3D円グラフの仰角（傾き）度を定義します。	
ROTATION	円グラフの回転角度を定義します。 値は「度」単位で指定します。	

プロパティ名=値	プロパティの説明	例
DISPLAYLABELS	列のタイトル（LABELS）を表示するか（=TRUE）しないか（=FALSE）を定義します。	
DISPLAYSLEGEND	系列の凡例を表示するか（=TRUE）しないか（=FALSE）を定義します。	

FILEEDITコントロール

このコントロールはダイアログボックスを表示し、ファイルやフォルダの保存または読み込みを可能にします。

警告:

このコントロールはWebクライアントでは動作しません。

Webクライアントでは、このコントロールはEDIT**コントロールとして表示され、値を入力することはできません。**

ファイルブラウザを起動するためのアイコンはありません。

表 15.33. FILEEDITLコントロールのプロパティ

プロパティ名=値	プロパティの説明	例
OPENMODE	ダイアログボックスのタイプを定義します。 ■ OPENMODE=1：ファイルを開きます。 ■ OPENMODE=2：ファイルを保存します。 ■ OPENMODE=4：フォルダを開きます。 ■ OPENMODE=8：フォルダを保存します。	
FILTERS	ダイアログボックス内に列挙されるファイルの表示基準を定義します。	(*.txt)|*.txt|(*.scn)|*.scn|
DEFEXT	デフォルトのファイルの拡張子を定義します。	(*.scn)|*.scn|

TICKEDITコントロール

このコントロールではスケジューラを挿入できます。

表 15.34. TICKEDITコントロールのプロパティ

プロパティ名=値	プロパティの説明	例
VALUE	ユーザがタイマで定義したパラメータを文字列の形式で表示します。	

プロパティ名=値	プロパティの説明	例
LISTHEIGHT	TICKEDITコントロールのサイズを、ウィザード全体の他のコントロールを基準にして定義します。 LONG（倍長整数）型のプロパティ	

CALENDARコントロール

このコントロールではカレンダを挿入できます。

TIMESPANEDITコントロール

このコントロールでは、期間の編集枠に入力できます。

NUMBOXコントロール

このコントロールでは数値タイプのコントロールを挿入できます。

表 15.35. NUMBOXコントロールのプロパティ

プロパティ名=値	プロパティの説明	例
MINVALUE	数の最小値を定義します。デフォルト値は無限値です。	
MAXVALUE	数の最大値を定義します。デフォルト値は無限値です。	
FORMAT	数のフォーマットを定義します。 ■ オペレーティングシステムのコントロールパネル内で定義したフォーマットのLONG ■ RAWLONG ■ オペレーティングシステムのコントロールパネル内で定義したフォーマットのDOUBLE	

COMBOEDITコントロール

このコントロールではドロップダウンリスト型のコントロールを挿入できます。

表 15.36. COMBOEDITコントロールのプロパティ

プロパティ名=値	プロパティの説明	例
VALUES	コントロールの値の組合せを定義します。「名前」にはコントロールに表示するテキストを定義します。「値」には、ユーザがこの「名前」を選択した場合コントロールに割り当てる値を定義します。 文字列型のプロパティ	

DATETIMEEDITコントロール

DATETIMEEDITコントロール

表 15.37. コントロールのプロパティ

プロパティ名=値	プロパティの説明	例
FORMAT	コントロールのフォーマットを定義します。 ■ 日付（date） ■ 時刻（time） ■ 日付+時刻 コントロールのフォーマットは、ユーザが定義するシステムパラメータにより変化します。	2002/02/07 13:37:19 2002/02/07 13:37:19

グラフィカルエディタの使用方法

Asset Managerでは、内蔵のグラフィカルエディタを使ってウィザードを作成できます。このエディタは、ウィザードの作成プロセスを簡略化および高速化することを目的としています。スクリプト言語の代わりに使用するものではありません。ウィザードを設計するには、スクリプト言語を理解できることが必要です。

- インタフェースの概要
- 新しいノードを作成する
- ノードのプロパティを編集する
- ウィザードをコンパイル、実行、およびデバッグする

注意:

グラフィカルエディタを使用するには、作成または変更するアクションのタイプ（SQL名：seActionType）が「ウィザード」のアクションでなければなりません。

インタフェースの概要

グラフィカルエディタにアクセスするには、［**ツール／アクション／編集**］メニューを選択します。グラフィカルエディタは、アクションの詳細画面の［**ウィザード**］タブページに表示されます。このエディタは次の3つの部分から構成されています。

- 最も一般的な機能を含むツールバー
- ウィザードの構造をツリー構造で示す［**階層**］セクション
- 階層で選択されたノードのプロパティの一覧を表示するセクション

ツールバー

ツールバーでは、編集コマンドを直接起動できます。アイコン上にしばらくマウスポインタを置くと、ツール名が表示されます。

編集コマンド

4つの編集コマンドを使用できます。

- は、エディタのテキストモードとグラフィックモードを切り替えます。
- は、同じ親ノード内の1つ上位のノードに移動します。
- は、同じ親ノード内の1つ下位のノードに移動します。
- は、選択したノードを削除します。

実行およびデバッグコマンド

これらのコマンドを使って、スクリプトをコンパイル、デバッグ、および実行できます

図 15.2. 実行およびデバッグ ボタン

検索ツール

ツールバーには、ウィザードのツリー構造で文字列を検索できる検索ツールがあります（［Ctrl+F］キーで直接このコントロールを起動できます）。

この枠をクリックし、検索するテキストを入力します。該当するテキストが見つかった場合は、その場所が強調表示されます（［F3］および［Shift+F3］キーを使って、それぞれ次のまたは前の一致文字列に移動できます）。

注意:

テキストモードの場合は、テキスト全部を検索します。グラフィックモードの場合は、プロパティ名だけで検索します。

ウィザードのツリー構造

グラフィカルエディタの左側には、ウィザードがツリー構造で表示されます。

ツリー内のノードを選択すると、画面の右側に、このノードに関連するプロパティが一覧表示されます。

選択したノードに対応するプロパティのリスト

画面の右側では、ノードのプロパティの値を入力できます。

各プロパティには固定値またはスクリプトがあります。プロパティの種類は次のように色分けされています。

- プロパティでデフォルト値が使われている場合は、その名前と値がグレーで表示されます。このプロパティに他の値を指定できます。その値は黒で表示されます。
- プロパティでユーザ定義値またはスクリプトが使われている場合は、その名前と値が黒で表示されます。
- 必須プロパティの場合は、その名前と値が赤で表示されます。
- 変更された値は、青で表示されます。

新しいノードを作成する

ここでは、ノードで実行できる操作について説明します。ツールバーでは、ノードを上下に移動したり、削除したりすることができます。まず、新しいノードの作成について説明します。

注意:

ショートカットメニューを使って、ノードを上下に移動したり、削除したりすることもできます。この場合は、選択したノードをマウスの右ボタンでクリックします。

ノードを作成するには、まず、親ノードを選択します。例えば、新しいPAGEノードを作成するには、最初にROOTノードをクリックします。親ノードを選択したら、マウスの右ボタンでクリックしてポップアップメニューを表示します。［新規作成］メニューを選択すると、作成できるノードがグループごとに表示されます。

ウィザードのツリーにノードが挿入されます。

ノードのプロパティを編集する

ノードを作成したら、ノードのプロパティに値を割り当てることができます。この作業は、エディタの右側で行います。

プロパティの値は、次の2つの方法で定義できます。

- 固定値を入力する。

- スクリプトを定義する。

注意:

スクリプトは常に、固定値より優先されます。プロパティにスクリプトと固定値の両方を割り当てると、固定値は無視され、スクリプトが解釈されます。

プロパティに固定値を割り当てる

対象のプロパティの［値］列を直接クリックします。プロパティで使用可能なデータ型（テキスト、ブール、倍精度数値など）に従って、リストから値を選択するか、テキスト編集枠に入力します。

プロパティにスクリプトを割り当てる

スクリプトを割り当てるプロパティを選択します。スクリプトそのものは、プロパティのリストの下にある［**スクリプト**］フィールドに入力します。

注意:

ショートカットメニュー（プロパティを右クリック）の［デフォルト値に戻す］を選択すると、Asset Managerは固定値やスクリプトを取り消し、プロパティをそのデフォルト値で再設定します。この操作は、値やスクリプトがユーザによって定義されているプロパティに対してのみ実行できます（このようなプロパティは黒色で表示されます）。

ウィザードをコンパイル、実行、およびデバッグする

エディタのツールバーの ボタンをクリックして、ウィザードを起動できます。実行中に発生したエラーは、エラー履歴ウィンドウ（内蔵のウィザードデバッガからアクセス可能）に表示されます。［Shift+F9］キーを押すと、実行を中断し（ウィザードが固定ウィンドウの場合）、デバッガを起動できます。

このようにして、ウィザードのエラーを簡単に訂正できます。

注意:

ウィザードがコンテキストに依存する場合は、実行ボタンを使用できません。

ウィザードの作成例

ウィザードのプログラミングを論理的に説明するために、「移動」ウィザードを作成します。「データベース」タイプのアクションと関連させることで、ユーザ

と関連資産をある設置場所から他の設置場所へ移動するプロセスを簡略化します。このウィザードの作成方法の手順を次に説明します。この手順に従って実際にウィザードを作成し、問題が発生した場合に参考にしてください。

1 ウィザードの作成例
2 手順1 - 用途を分析する
3 手順2- ウィザードの構成方法を定義する

ウィザードの作成例

このウィザードの目的は、資産をある設置場所から別の設置場所へ移動することです。このため、次の操作を行う必要があります。

1 移動する資産を指定する。
2 これらの資産の新しい設置場所を選択する。

移動する資産の指定方法

移動する資産を指定するには、3つの方法があります。

- 資産のユーザから移動する資産を特定する。ユーザを選択した後に、移動する資産を選択する必要があります。
- ［資産］テーブルのレコードを選択して、移動する資産を直接選択する。
- 設置場所から移動する資産を特定する。まず、設置場所を選択し、次にこの設置場所から移動する資産を選択します。

注意:

ユーザが、移動する資産の選択方法をこれらの中から選択できるよう、ユーザ選択ページを作成する必要があります。

新しい設置場所を選択する

資産の新しい設置場所を選択するには、設置場所のテーブルからレコードを選択するだけです。

手順1 - 用途を分析する

手順1で定義したニーズに従って、次のように、ウィザードの編成を定義する必要があります。

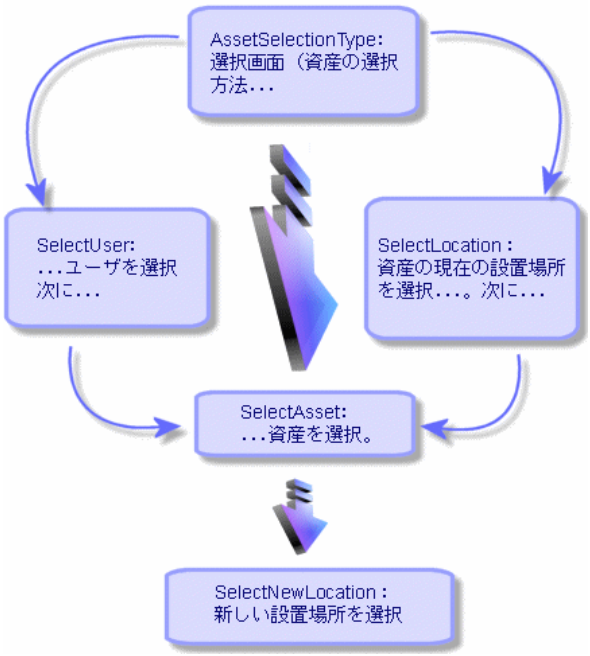
1 ページ数
2 ページ同士のリンク方法
3 各ページの内容

注意:

手順1で、選択ページを作成する必要があることがわかりました。このページがウィザードの最初のページになります。このページを「AssetSelectionType」と呼ぶことにします。

次の図を参考にして、ウィザードの定義方法を定義します。

図 15.3. ウィザードの編成



このフローチャートを使って、ページごとの遷移を定義できます。

ページ	遷移のリンク先ページ
AssetSelectionType	SelectAsset、SelectUser、SelectLocation
SelectAsset	SelectNewLocation
SelectUser	SelectAsset
SelectLocation	SelectAsset
SelectNewLocation	なし

次に、ページの内容を定義します。つまり、ユーザが選択を実行できるコントロールを定義します。

ページ	このページの目的	使用するコントロール
AssetSelectionType	ユーザが3つの選択肢から選択できるようにします。	CHOICEコントロール
SelectAsset	ユーザが［資産］テーブルのレコードリストから資産を選択できるようにします。	ADBLISTコントロール
SelectUser	ユーザが、［部署と従業員］テーブルから資産を移動するユーザを選択できるようにします。	ADBLISTコントロール
SelectLocation	ユーザが、［場所］テーブルから現在の設置場所を選択できるようにします。	ADBLISTコントロール
SelectNewLocation	ユーザが、［場所］テーブルから新しい設置場所を選択できるようにします。	ADBLISTコントロール

手順2- ウィザードの構成方法を定義する

この手順では、ウィザードのスクリプトを作成します。スクリプトには、ウィザードの各ノードの構造を記述します。下に、「移動」ウィザードのコメント付きソースノードを示します。このコードは、ウィザードの作成の一例にすぎません。同じタスクを実行するウィザードの作成方法は、他にもたくさんあります。

```
;==========================================================
=======
NAME = "移動"
TITLE = "移動する従業員"
VERSION = "699"
;==========================================================
==========================
;移動するユーザを指定。デフォルトでは、このテーブルがコンテキストの場合
、amEmplDeptから選択
;==========================================================
==========================
{ PAGE pgUser
TITLE = "移動する従業員の選択"
{ DBLISTBOX Users
COLNAME = "Name|FirstName"
COLWIDTH = "50|50"
DBLCLICK = 1
LABEL = "移動するユーザ"
MULTISEL = 1
TABLE = "amEmplDept"
{ VALUE =
if [CurrentTable] = "amEmplDept" then
```

```
RetVal = [CurrentSelection]
else
RetVal = ""
end if
}
VISIBLE = 1
}
{ TRANSITION trPersonToNewLoc
TO = "pgNewLoc"
}
}

;==========================================================
==========================
;新しい設置場所を指定
;==========================================================
==========================
{ PAGE pgNewLoc
TITLE = "新しい設置場所の選択"
{ STRING UserName
VALUE = AmDbGetString("SELECT FirstName + ' ' + Name FROM amEmp
lDept WHERE lEmplDeptId IN (" & {pgUser.Users} & ")" )
}
{ LABEL LABEL1
CAPTION = "ユーザ: " & {UserName}
}
{ DBLISTBOX NewLocId
COLNAME = "名前"
COLWIDTH = "100"
DBLCLICK = 1
TABLE = "amLocation"
VALUE = "-1"
}
{ TRANSITION trNewLocToAssets
TO = "pgRecap"
}
}

;==========================================================
==========================
;選択内容の確認
;==========================================================
==========================
{ PAGE pgRecap
```

```
TITLE = "Recapitulation"
{ LISTBOX Users
COLTITLE = "名前"
COLWIDTH = "100"
LABEL = "移動するユーザ"
MANDATORY = 0
MULTISEL = 1
READONLY = 1
VALUE = ""
VALUES = AmDbGetList("SELECT FullName FROM amEmplDept WHERE
FullName LIKE LikeParam(amEmplDept_2:FullName)+'%' AND amEmplDe
pt_2:lEmplDeptId IN(" & {pgUser.Users} & ")","|",",", "=")
}
}

;==========================================================
=========================
;最後
;==========================================================
=========================
{ FINISH FINISH
{ DO =
On Error Goto ErrHandler
Dim lErr as long

dim hRecord as Long

dim iEmplCount as Integer
iEmplCount  = {pgRecap.Users.VALUES.Count()}
dim iMax as Long
iMax = iEmplCount

dim lLocaId as long
lLocaId = {pgNewLoc.NewLocId}

lErr = amStartTransaction()

dim i as Integer
For i = 1 To iEmplCount
lErr = AmProgress((100 * i ) / iMax)
lErr = AmLog("Moving the employee" + {pgRecap.Users.VALUES(i,1)})
hRecord = AmGetRecordFromMainId("amEmplDept", {pgRecap.Users.VALU
ES(i,0)} )
If hRecord <> 0  then
```

```
lErr = AmSetFieldLongValue( hRecord, "lLocaId", lLocaId)
lErr = AmUpdateRecord(hRecord)
lErr = AmReleaseHandle(hRecord)
End If
Next i

lErr = amCommit()

RetVal = 0
Exit Function


ErrHandler:
On Error Goto 0
AmLog(AmLastError() & " - " & AmLastErrorMsg())
AmLog("The transaction has been cancelled")
RetVal = 1
Exit function
}
SUMMARY = 1
}
```

ウィザードに関する一般的な質問

ここでは、ウィザードの作成時に起こり得る問題の対処法を紹介します。

質問

次のコードが機能しません。

```
{lbxMyListBox.Values.Count}
```

答

メソッドのシンタックスに開き括弧と閉じ括弧を入力する必要があります。正しいコードは次のようになります。

```
{lbxMyListBox.Values.Count()}
```

質問

次のコードが機能しません。

```
{lbxMyListBox.Line(lRow)}
```

答

LINEメソッドは、LISTBOXコントロールのVALUESプロパティに関連付けられます。正しいコードは次のようになります。

```
{lbxMyListBox.Values.Line(lRow)}
```

質問

次のコードが機能しません。

```
{lbxMyListBox.Values.Line({lbxTmp})}
```

答

メソッドでは参照するプロパティを使用できません。次のように記述します。

```
Dim lRow As Long
lRow = {lbxTmp}
{lbxMyListBox.Values.Line(lRow)}
```

質問

プロパティに固定値を割り当てる次のコードが機能しません。

```
{Property} = 123
```

答

プロパティに値を割り当てるには、次のように、AmSetProperty()関数を使う必要があります。

```
Dim irc as Integer
irc= AmSetProperty("Property", 123)
```

注意:

戻りコードを使う必要がない場合でも、必ず戻りコードを取得するようにしてください（この場合は「irc」）。

質問

データベースに資産を作成するウィザードを実行すると、次のエラーメッセージが表示されます。

12001 - 書き込み権限がありません。

ユーザが管理者として接続している場合も、このメッセージが表示されます。

答

このメッセージは、ウィザードのFINISH.DOノード外で書き込みアクセスを試みると表示されます。ウィザードは次の操作を実行します。

1. Asset Manager管理者も書き込みできない一連のページから情報を収集する。
2. ユーザの権限に応じて書込みできるFINISH.DOノードに含まれるスクリプトを実行する。

質問

ウィザードの実行中に表示されるエラーメッセージが不完全なことがあります。

答

［Shift+F9］キーを押して、デバッグを表示します。多くの場合、履歴ウィンドウのエラーメッセージの方が明確です。

質問

ウィザードのページでDBLISTBOXコントロールを使うと、パフォーマンスが低下します。これは普通のことなのでしょうか。

答

この問題は、フィルタと一緒にDBLISTBOXコントロールを使うと発生します。この場合、選択を変更するたびに、データベースにクエリが送られ、選択がフィルタに一致するかどうかが確認されます。選択がユーザによって設定された場合、この追加クエリは実行されません。

質問

LISTBOXコントロールで特定の列を編集可能または編集不可にするにはどうすればよいですか。

答

このコントロールのEDITABLEプロパティを使います。このプロパティに割り当てられる値は「0」と列の区切り文字として使われるパイプ文字（|）で区切られる文字で構成されています。「0」は列を「編集不可」として、「1」は列を「編集可能」として定義します。値を省略すると、対応する列は編集できず、列2と4が編集可能になります。

```
EDITABLE = "|1||1"
```

質問

ウィザードで詳細ウィンドウを開くようにするにはどうすればよいですか。

答

ウィザード内でDDEコール（関数から）を使う必要があります。ウィザードは固定ウィンドウにしないでください。ウィザードから資産テーブルを開く方法は、以下の通りです。

```
Dim irc as Long
irc = AmActionDDE("am", "Asset Manager", "OpenTable(amAsset)")
```

質問

LISTBOXコントロールの「COLNAME」プロパティと「COLTITLE」プロパティの違いを教えてください。

答

LISTBOXコントロールの列のタイトルは、自動的に、または手動で定義できます。

- 「TABLE」プロパティに関連付けられた「COLNAME」プロパティでは、データベースのフィールドラベルを使って、LISTBOXコントロールの列タイトルを自動的に定義できます。
- 「COLTITLE」プロパティに値が入力されている場合は、その値が列のタイトルとして使用されます。このプロパティが定義されていない場合は、列タイトルは「COLNAME」プロパティで定義されたタイトルになります。

例

```
...
TABLE = "amEmplDept"
COLNAME = "Name||FirstName"
COLTITLE = "|A|B"
...
```

この例の場合、LISTBOXコントロールには、Name、A、Bというラベルが表示されます。

「COLNAME」プロパティでは、列値が編集可能な場合に、使用するコントロールのタイプも定義します。

16 ニュース

この章では、Asset Managerのユーザにニュースを配信し、管理する方法について説明します。

ニュースのリストを表示するには、［**ツール／ニュース**］メニューを選択します。

ニュースを表示するには、［**ウィンドウ／ニュースマーキーの表示**］メニューまたはツールバーの アイコンを使います。

ニュースの概要

ニュースとは特定の期間に指定した受信者のグループに流す新情報のことを指します。

これらの従業員は従業員グループに所属します。

原則として、ニュースは短い期間に関する情報です。

例：「2009/02/10、11：00から12：00の間にサーバXXXがダウンします。」

ニュースの概要

新しいニュースを作成する

ニュースのテーブルでのニュース作成の権限を持つAssetCenterのユーザは、［**ツール／ニュース**］メニューを使ってニュースを作成できます。

ニュースの詳細には、主に次の情報が表示されます。

- メッセージ
- メッセージの配信先リスト
- メッセージの有効期間

ニュースを表示する

ニュースの配信先リストに含まれているユーザは、ニュースマーキーを表示できます。

ニュースの重要度

ニュースメッセージの重要度を定義するには、ニュースの詳細画面にある［**重要度**］（SQL名：seSeverity）フィールドで指定します。

各重要度には［**色**］（SQL名：IColor）フィールドで選択された色が関連付けられます。ニュースマーキーには、この色でニュースが表示されます。

配信するメッセージ

ニュースの詳細画面の［**メッセージ**］タブページには、配布するメッセージが表示されます。

メッセージの長さは最大255文字です。すべての文字を使用できます。

ニュースの配信先リスト

ニュースの詳細画面の［**配信先**］タブページには、そのニュースを読むことができる従業員グループの一覧が表示されます。

［すべての従業員グループ］（SQL名：bAllGroups）チェックボックス

このチェックボックスをオンにすると、すべての従業員グループのメンバーはAssetCenterでメッセージを表示できます。

チェックボックスがオフの場合は、［**配信先**］タブページで選択された従業員グループのメンバーのみがメッセージを表示できます。

配信先リストの従業員グループを追加、削除、表示、または変更するには、それぞれ、、および ボタンを使います。

［サブグループを含む］（SQL名：bChildGrps）チェックボックス

ニュースのメッセージは、全従業員グループまたは選択したグループだけに配信できます。このチェックボックスをオンにすると（デフォルトではオン）、選択されたグループの全サブグループもニュースを表示できます。これは従業員グループのテーブルが階層構造になっているからです。

ニュースを表示する

ニュースを表示するには：

1 ニュースマーキーを起動します。
2 マーキーの右側にあるボタンを使って、複数のニュースをスクロールおよび参照します。

以下を定義できます。

- ニュースの重要度に応じたマーキーの配色
- ニュースマーキーツールバーの自動更新モード

ニュースマーキーを起動する

Asset Managerのすべてのユーザがニュースマーキーを表示できます。現在ログインしているユーザに関連するメッセージが表示されます。ニュースマーキーは、次の2つの方法で起動および終了できます。

- ［**ウィンドウ／ニュースマーキーの表示**］メニューを使う。
- ツールバーの アイコンを使う。

表 16.1. マーキーツールバーのボタン

	オプションで定義されている時間を待たずに（［**編集／オプション**］メニュー）、即座に現在のニュースを読むことができます。

◀	前のメッセージを表示します。
▶	次のメッセージを表示します。
❚❚	表示されているメッセージを中断または再開します。

索引

*= (AQL), 50
< (AQL), 50
<> (AQL), 50
=! (AQL), 50
= (AQL), 50
=* (AQL), 50
=< (AQL), 50
= ALL (AQL), 50
= ANY (AQL), 50
= SOME (AQL), 50
> (AQL), 50
>= (AQL), 50
アクション, 127-146
(参考 ウィザード)
タイプ, 130
テスト, 142
ボタン - 関連付け, 146
メッセージ, 148
印刷 - コンテキスト, 133
機能ドメイン, 129
作成, 130
実行, 143
実行可能タイプ
実行, 144
定義, 128
変数, 142
例, 139
DDE, 139
スクリプト, 140
メッセージ, 140
実行可能ファイル, 139
［DDE］タブ, 135
［メッセージ］タブ, 137
アクションのタイプ, 130
アクションのテスト, 142
アクション（アクションのタイプ）, 134, 133
アクセス権の定義
Basic関数, 222
FINISH（ウィザードのノード）, 290
LONG（ウィザードのノード）, 293
ROOT（ウィザードのノード）, 280
START（ウィザードのノード）, 291
STRING（ウィザードのノード）, 293
TIMER（ウィザードのノード）, 292
TRANSITION（ウィザードのノード）, 288
twip, 268
オブジェクト, 269
オブジェクトのフルネーム, 269
コントロール, 268

コントロール（ウィザードのノード）, 293
ニュース, 331
ノード, 269
ページ（ウィザードのノード）, 286
ワークフローのアクティビティの担当者, 153
終了ワークフローイベント, 197
従属オブジェクト, 269
出力クエリ, 210
出力スクリプト, 209
親オブジェクト, 269
遷移, 270
特殊フィールド, 259
変数, 270
アドレス - メッセージタイプのアクション, 137
アプリケーションサーバのURLアドレスの編集...（ウィザード）, 85
インデックス
(参考 クエリ)
ウィザード, 267-330
{ - シンタックス, 273
Basic
連結, 279
Basic関数, 279
FINISH（ノード）, 290
LONG（ノード）, 293
ROOT（ノード）
サブノード, 285
シンタックス, 280
プロパティ, 280
定義, 280
START（ノード）, 291
STRING（ノード）, 293
TIMER（ノード）, 292
TRANSITION（ノード）
シンタックス, 288
プロパティ, 288
定義, 288
特異性, 289
エディタ
アクセス, 317
インタフェース, 318
ツールバー, 318
テキスト - 検索, 318
使用, 317
コントロール, 294
CALENDAR, 316
CHART, 313
CHECKBOX, 297
COMBOBOX, 297
COMBOEDIT, 316
COMMANDBUTTON, 303
DATETIMEEDIT, 317
DBEDIT, 309
DBLISTBOX, 304
DBPATH, 310
DBQUERYBOX, 306
DBTABLE, 310
FILEEDIT, 315
LABEL, 302
LINKEDIT, 311
LISTBOX, 299
NUMBOX, 316
OPTIONBUTTONS, 298
PROGRESSBAR, 303
TEXTBOX, 312
TICKEDIT, 315
TIMESPANEDIT, 316
コントロール（ノード）, 293 , 293
スクリプト - 制限, 279
セル - 値, 304
デバッグ, 320
ノード
CurrentSelection, 277
CurrentTable, 277
スクリプト, 275
スクリプトまたは固定値 - 優先順位, 320
テーブル, 276
プロパティ, 319 , 274 , 273
プロパティ - 制限, 273
メソッド, 276
構造とシンタックス, 272
作成, 319
数値, 274
宣言的モデル, 274
定数, 274
名前 - 制限, 269
プロパティ, 294
ページ - モデル, 272

ページ（ノード）
サブノード，287
シンタックス，286
プロパティ，286
定義，286
開始 - ページ，292
概要，272
構造テンプレート，270
作成パラメータ，290
作成 - 例，320
質問，326
実行，320
実行 - 制限，320
表記 - 規則，267
連鎖，278
ウィザードの遷移，270
ウィザード（アクションのタイプ），132
エラー - スクリプト，234
エラーメッセージ - スクリプト，234
オブジェクト，269
オブジェクトのフルネーム，269
カレンダ，239-245
コントロール，244
プレビュー，239
一般情報，241
概要，239
作成，240 ，240
精度，242
他の機能への影響，240
変更 - 考慮，240
［予定表］タブ，241
クエリ，23-66
AQLとSQLの比較，25
AQLの利点，24
CurrentUser，38
DELETE，56
DUPLICATE，56
FROM，51
FullName（フィールド），40
GROUP BY，53
HAVING，53
INSERT，55
NULL，37
ORDER BY，54
Self，37
sLvl（フィールド），40
UPDATE，55
WHERE，52
インデックス，42
インデックス - 注意事項，44
エディタ，26
アクセス，27
概要，26
システムリストデータ，39 ，39
シンタックス，44
UNION，46
サブクエリ，45
シンプルクエリ，45
表記法，45
ソート，42
インデックスの強制使用，43
リストデータ，43
順序，43
注意事項，44
テーブル数，51
はじめに，23
フィールド，30
フィルタ，28
直接入力，29
プレビュー，29
リストデータ，33
階層構造をもつテーブル，40
外部キー，41
外部結合
演算子，37
数値，35
概要，25
管理者，28
簡単な表記法，41
関数 (参考 AQL関数)
共有，28
結合，34
作成，28
式，30
主キーが0のレコード，35
推奨事項，33
数値フィールド，37
定数，33
特殊フィールド，42
任意管理項目，42

必要な技術，23
表記文字列，41
要素，46
フィールドとリンク，46
演算子，49
式，49
選択リスト，51
定数，47
変数，48
例，62 , 62
2つの条件，63
alias，65
FullName，63
FullNameおよびsLvl，63
メインテーブルにリンクしたテーブルのフィールド，63
リンク，63 , 62
リンクなし，64
階層構造をもつテーブル，63
式，64
数値、日付、テキスト，64
任意管理項目，64
変数，66
未入力のフィールド，64
クエリエディタ (参考 クエリ)
クエリのシンタックス (参考 クエリ)
クエリ（メニュー），27
コメント - スクリプト，233
コントロール，268
コントロールパネル，101-104
インジケータ，101
ナビゲーション，104
レポート機能，101
概要，101
作成，101
統計とレポート，101
例，103
コントロール（ウィザードのノード），293 , 293
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），258 , 255
サブクエリ，45
システムリストデータ - スクリプト，231
システムリストデータ - 値，39
シングルサインオン (参考 Crystal Reports Server)
スクリプト，219-238
Basic関数
コールに応答する，221
タイプ - 制約，225
パラメータの型，224
プログラマブル関数，223
ヘルプ，220
関数の型，224
組み込み関数，222
分類，225
名前 - 制限，229
CurrentUser，232
エラーメッセージ，234
コメント，233
システムリストデータ，231
ノード（ウィザード），275
ヒント，230
ライブラリ，228
リンク - ID，220
期間，231
時刻，230
手順1，225
大文字と小文字の区別，227
注意事項，230
日付，230
表記 - 規則，220
編集，219
用途，221
例，237 , 234
スクリプトライブラリ，228
スクリプト（アクションのタイプ），132
ソート
(参考 クエリ)
ソフトウェア配布（アクションのタイプ），133
タイムゾーン
Asset Manager Automated Process Manager - 制限，255
インポート，258
クライアントマシンの時刻，257
可用性，247
夏時間，254 , 249
概要，254

管理，249
作成，248
出力，258
接続，257
操作に与える影響，255
有用性，247
タイムゾーンのインポート，258
タイムゾーンの使用（オプション），256，248
タイムゾーン - 実装，248
タスクを保存（オプション），198
テーブル
(参考 履歴)
ワークフロー，207
テスト/スクリプト（ワークフローのアクティビティ），181
デフォルト値
フィールド (参考 スクリプト)
任意管理項目 (参考 スクリプト)
デフォルト値に戻す（メニュー），320
データベース
(参考 履歴)
作成，72
準備，72
接続，72
データベースサーバのタイムゾーンの検証（オプション），255
データ - 出力 (参考 出力)
ニュース
メッセージ，332
概要，332
作成，332
重要度，332
定義，331
読取り，332
配信先リスト，332
表示，333
ニュースの重要度，332
ノード（ウィザード）
構造とシンタックス，272
定義，269
フィールド
(参考 履歴)
クエリ，63，46
未入力のフィールド，64
フィールドの設定
(参考 スクリプト)
フィルタ - 特殊フィールド，265
プログラマブル関数，230
ページ（ウィザードのノード），286
メソッド - ウィザード，276
メッセージ，147-149
メッセージの参照，148
メッセージの送信，148
概要，147
内部メッセージシステム - アドレス，131
配信済通知，149
メッセージ（アクションのタイプ），131
メッセージ（タブ），137
メッセージ（メニュー），149
モジュール
レポート，99
ユーザのアクション（ワークフローのアクティビティ），180
ユーザ権限 - 特殊フィールド，264
リスト
リストデータ値，14
特殊フィールド，265
リストデータ，13-334
オープンカスタムリストデータ，14
カスタムリストデータ，14，14
クローズ済カスタムリストデータ，14
システムリストデータ，15，15
リストの出力（メニュー），211
リストの設定（メニュー），43，28
リンク
(参考 履歴)
クエリ，63，62，46
リンクなし，64
レポート，69-99
1つずつインポート，91
Asset Managerデータベースへの格納，91
Crystal Reports Serverへの格納，93
CurrentUserId，96
ID，99
SQL名（制約），96
URLアドレス，89
インストール，72，71
既存のデータベース，92
新規データベース，91
すべてのレポートのインポート

既存のデータベース，92
新規データベース，91
タイプ，70
ボタン - 関連付け，100
印刷，97
可用性，88
概要，69
格納，70
構成，72
作成，90
参照，70
詳細レポート
作成，95
使用 - 例，95
設定，96
制限，69
設定，71
統計，88
表示，86
変更，95
レポートの印刷，97
ワークフロー，151-205
SQL名（制限），182
アクション - トリガ，179
アクティビティ，178
テスト/スクリプト，181
テンプレート，183
トリガ，183
ユーザのアクション，180
開始，182
自動アクション，181
質問，179
担当者 - アクティビティのタイプ，177
［アラーム］タブ - 制限，199
［制限時間］タブ - 制限，199
アクティビティテンプレート - 制限，183
アラーム，198
イベント，187
アラーム，188
システム，187
データベース，188
ユーザ，188
起動，191
終了イベント，197
処理，192
定期的，191
［開始］アクティビティ，188
インスタンス
削除，203 ，203
自動削除，203
グラフィカルエディタ，155
コンテキスト，175
タスク，184
アラーム - 制限，184
スクリプト，184
テスト，184
ユーザタスク，185
委任，186
割当，185
完全リスト，185
管理者，187
作成，184
自動アクション，184
テーブル，207
トラッキング，201
トリガ - 制限，190
概要，153
期間，199
期限，198
実行グループ，200
制限時間，199 ，199
遷移，198
定義，152
同期ワークフロー，196
導入，155
複雑な入力条件，183
役割，177
例，158
アクティビティ，164
イベントの設定，170
トリガ，172
開始イベント，170
遷移，171
前提条件，159
ワークフロー - アクション - スクリプト，233
ワークフローインスタンス
削除，203
制限，176
定義，152
ワークフローエディタ，155

ワークフロースキーム (参考 ワークフロー)
ワークフロースキーム（メニュー），153
ワークフローのアクティビティ，178
　グラフィカルエディタ，156
　定義，152
ワークフローのアクティビティのテンプレート，183
ワークフローのアクティビティの担当者
　割当，178
　定義，153
ワークフローのアラーム，199 , 198
ワークフローのイベント，187
　エディタ，156
　定義，153
ワークフローのコンテキスト，175 , 175
ワークフローのタスク，184 , 153
ワークフローの期限，198 , 198
ワークフローの実行グループ，200 , 153
ワークフローの制限時間，199
ワークフローの遷移，198
　エディタ，157
　定義，153
ワークフローの役割，177
印刷（アクションのタイプ），133
演算子 - クエリ，49
夏時間，252
夏時間（フィールド），249
開始（ワークフローのアクティビティ）, 182
階層構造をもつテーブル - クエリ，63
外部キー - クエリ，41
外部結合
　演算子，37
　数値，35
期間 - スクリプト，231
機能ドメイン - アクション，129
計算（ボタン），143
結合，34
参照オブジェクト - ワークフロー，175
参照オブジェクト（フィールド），137
時刻 - スクリプト，230
自動アクション（ワークフローのアクティビティ），181
式 - クエリ，49
質問（ワークフローのアクティビティ）, 179
実行可能ファイル
　アクション
　　実行，144
実行可能ファイル（アクションのタイプ），130
実行（ボタン），143
終了ワークフローイベント，197 , 197
従属オブジェクト，269
出力，209-218
　SQLビュー，211
　　作成，216
　　削除，216
　アクセス権限，211
　クエリ，215
　スクリプトの出力形式，215
　タイムゾーン，258
　ビュー（タブ） - 表示，216
　フィルタ，215
　プレビュー，215
　プロセス，210
　　スクリプト，210
　　状況依存メニュー，211
　出力クエリ，213
　出力スクリプト
　　実行，217
　　実行 - DOS，218
　　実行 - プロセス，217
　　書込み，212
　出力スクリプトの作成方法，213
　出力するデータ，214
　出力する列，214
　書式設定（タブ） - 表示，215
　推奨事項，212
　並べ替え，214
出力クエリ
　設計，213
　定義，210
出力スクリプト
　実行，217
　書込み，212
　定義，209
親オブジェクト，269
進行中のワークフローがある場合にワークフローを再度初期化（オプション），191
数値 - クエリ，64
宣言的モデル - ウィザード，274

選択リスト - クエリ，51
値のリスト (参考 リストデータ)
定義
　アクション，128
　ワークフローインスタンス，152
　ワークフローのアクティビティ，152
　ワークフローのイベント，153
　ワークフローのタスク，153
　ワークフローの実行グループ，153
　ワークフローの遷移，153
定数
　クエリ，47
　ノード（ウィザード），274
統計，105
　グラフ，109
　スクリプト，107
　　例，115
　ナビゲーション，109
　パフォーマンスの最適化，108
　　ソリューション，108
　　ソリューションの実装，108
　　問題，108
　結果の格納，108
　作成，105
　統計を見やすくする，110
　　コントロールパネルから，110
　　ビジネスホームページから，110
　表示，111
　表示の設定，109
　例，111 ，111
特殊フィールド，259-266
　(参考 スクリプト)
　SQL名 - 制約，263
　クエリ，42
　タイプ，260
　　使用する言語，264
　　推奨事項，262
　　用途，261
　データベース - 値の格納，259
　フィルタ，265
　ユーザ権限，264
　リスト，265
　計算式，264
　作成，260
　参照，266
　使用，265
　使用のコンテキスト，263
　定義，259
　有効性，260
日付
　クエリ，64
　スクリプト，230
任意管理項目
　(参考 履歴)
　クエリ，64 ，42
配信済通知，149 ，137
配信先リスト - ニュース，332
表記文字列 - クエリ，41
変数
　アクション，142
　クエリ，48
　定義，270
変数（AQL），66
予定表，241
履歴，17-21
　1リンク，18
　nリンク，20 ，19 ，19
　データベース - インポート，18
　フィールド，18
　レコード，18
　概要，17
　削除，21
　任意管理項目，20
　　起動，21
　　削除，20
　　追加，20
　　変更，21
　　履歴項目の作成、削除、変更，21
　履歴項目を作成する，17
履歴項目 (参考 履歴)
例外（タブ），242

，82 ，77 ，75 ，75

A

Abs (AQL)，61
AddDays (AQL)，59
AddHours (AQL)，59
AddMinutes (AQL)，59

AddSeconds (AQL), 59
Advanced Query Language (参考 クエリ)
alias (AQL), 65
am.ini, 257
amexpl.exe, 211
AND (AQL), 49
AQL (参考 クエリ)
AQL関数, 56
テスト, 62
IsNull, 62
集合型, 57
Avg, 57
Count, 57
Countdistinct, 57
Max, 57
Min, 57
Sum, 57
数値, 61
Abs, 61
Ceil, 61
DataLength, 61
Floor, 61
Length, 61
Mod, 61
NullNumeric, 61
NumberToNumber, 61
Round, 61
Sign, 61
TextToNumber, 61
Trunc, 61
日付, 59
AddDays, 59
AddHours, 59
AddMinutes, 59
AddSeconds, 59
Day, 59
DayOfYear, 59
DaysDiff, 59
DbToLocalDate, 59
Getdate, 59
Hour, 59
HoursDiff, 59
LocalToDbDate, 59
Minute, 59
MinutesDiff, 59
Month, 59
NumberToTime, 60
Second, 60
SecondsDiff, 60
TextToTime, 60
WeekDay, 60
Year, 60
文字列, 57
Ascii, 57
Char, 57
CharIndex, 58
DateToText, 58
EmptyString, 58
Left, 58
Length, 57
LikeParam, 58
Lower, 58
Ltrim, 58
NullBlob, 58
NullDate, 59
NullMemo, 58
NullString, 58
NullTime, 59
NullTimeStamp, 60
NumberToText, 58
Right, 58
Rtrim, 58
Substring, 58
TimeStampToText, 58
TimeToText, 58
Upper, 58
Ascii（AQL）, 57
Asset Manager Automated Process Manager - タイムゾーン, 254
Avg (AQL), 57

B

Basic (参考 スクリプト)
(参考 ウィザード)
情報, 220
Basic関数
(参考 ウィザード)
(参考 スクリプト)
定義, 222

C

CALENDAR（ウィザードのコントロール），316
Ceil (AQL)，61
Char (AQL)，57
CharIndex，58
CHART（ウィザードのコントロール），313
CHECKBOX（ウィザードのコントロール），297
COMBOBOX（ウィザードのコントロール），297
COMBOEDIT（ウィザードのコントロール），316
COMMANDBUTTON（ウィザードのコントロール），303
Count (AQL)，57
Countdistinct (AQL)，57
Crystal Reports (参考 レポート)
　目的，69
Crystal Reports (Designer)
　インストール，73
　　手動インストール，74
　　無人モード，74
Crystal Reports Server
　Asset Managerユーザ，77
　インストール，73
　　構成，75
　　手動インストール，74
　　無人モード，74
　サーバの宣言，85
　シングルサインオン
　　設定，82
　レポート
　　レポートのインポート，75
Crystal Reports - レポートの設定，96
Crystal Reports統計，88
CurrentSelection（ウィザードのノード），277
CurrentTable（ウィザードのノード），277
CurrentUser (AQL)，38
CurrentUserId（レポート），96
CurrentUser（スクリプト），232

D

database.txt，232，34，15
DataLength (AQL)，61
DATETIMEEDIT（ウィザードのコントロール），317
DateToText，58
Day (AQL)，59
DayOfYear (AQL)，59
DaysDiff (AQL)，59
DBEDIT（ウィザードのコントロール），309
DBLISTBOX（ウィザードのコントロール），304
DBPATH（ウィザードのコントロール），310
DBQUERYBOX（ウィザードのコントロール），306
DBTABLE（ウィザードのコントロール），310
DbToLocalDate (AQL)，59
DDE（アクションのタイプ），131
DDE（タブ），135
DELETE (AQL)，56
DUPLICATE (AQL)，56

E

EmptyString，58

F

FILEEDIT（ウィザードのコントロール），315
FINISH（ウィザードのノード），290
Floor (AQL)，61
FROM (AQL)，51
FullName (AQL)，63
FullName（フィールド），40
　クエリ，63

G

g_lTimeZoneCheckInMns (am.ini)，258
gbbase.wiz，281
gbbase.xml，92
Getdate (AQL)，59
GROUP BY (AQL)，53

H
HAVING (AQL), 53
Hour (AQL), 59
HoursDiff (AQL), 59

I
INSERT (AQL), 55
IS NOT NULL (AQL), 50
IsNull (AQL), 62
IS NULL (AQL), 50

L
LABEL（ウィザードのコントロール），302
Left (AQL), 58
Length, 57
Length (AQL), 61
LIKE (AQL), 50
LikeParam, 58
LINKEDIT（ウィザードのコントロール），311
LISTBOX（ウィザードのコントロール），299
LocalToDbDate (AQL), 59
LONG（ウィザードのノード），293
Lower (AQL), 58
Ltrim (AQL), 58

M
MAPI (参考 メッセージ)
Max (AQL), 57
Microsoft SQL Server
　クエリ
　　ソートの順序, 44
Min (AQL), 57
Minute (AQL), 59
MinutesDiff (AQL), 59
Mod (AQL), 61
Month (AQL), 59

N
NOT LIKE (AQL), 50
NULL (AQL), 37
NullBlob, 58
NullDate, 59
NullMemo, 58
NullNumeric (AQL), 61
NullString, 58
NullTime, 59
NullTimeStamp, 60
NumberToNumber (AQL), 61
NumberToText, 58
NumberToTime (AQL), 60
NUMBOX（ウィザードのコントロール），316

O
Old, 233
OPTIONBUTTONS（ウィザードのコントロール），298
OR (AQL), 49
Oracle for WorkGroups
　クエリ
　　ソートの順序 , 44
ORDER BY (AQL), 54

P
PROGRESSBAR（ウィザードのコントロール），303

R
reports.txt, 91 , 91
Right (AQL), 58
ROOT（ウィザードのノード），280
Round (AQL), 61
Rtrim (AQL), 58

S
Second (AQL), 60
SecondsDiff (AQL), 60
SELECT (AQL)
　サブクエリ - 警告, 45
　制限, 51
Self (AQL), 37
Sign (AQL), 61
sLvl - クエリ, 40
sLvl（フィールド） - クエリ, 63
SMTP (参考 メッセージ)
SQL

クエリ，57
データベース - 変更，25
SQLとAQLの比較，25
SQLビュー (参考 出力)
START（ウィザードのノード），291
STRING（ウィザードのノード），293
Substring (AQL)，58
Sum (AQL)，57

T

TEXTBOX（ウィザードのコントロール），312
TextToNumber (AQL)，61
TextToTime (AQL)，60
TICKEDIT（ウィザードのコントロール），315
TIMER（ウィザードのノード），292
TIMESPANEDIT（ウィザードのコントロール），316
TimeStampToText，58
TimeToText，58
TRANSITION（ウィザードのノード），288
Trunc (AQL)，61
twip，268
tz.scr，249

U

u2lamlib.dll（ファイル），75
UNION (AQL)，46
UPDATE (AQL)，55
Upper (AQL)，58
useSQL92Join (amdb.ini)，35

V

VIM (参考 メッセージ)

W

WeekDay (AQL)，60
WHERE (AQL)，52

Y

Year (AQL)，60