

Peregrine

AssetCenter



AssetCenterの高度な使い方

(c) Copyright 2005 Peregrine Systems, Inc.

All rights reserved.

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

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

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

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

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

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

本書の内容は、ライセンス契約に基づくプログラムのバージョン4.4に適用されます。

AssetCenter

Peregrine Systems, Inc.
3611 Valley Centre Drive San Diego, CA 92130
858.481.5000
Fax 858.481.1751
www.peregrine.com



目次

PEREGRINE

1. リストデータ	13
不特定のリストデータ	13
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）	19
リンク（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n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	31
並べ替えとインデックス	40
AQLのシンタックス	42
AQLの関数	54
クエリの例	60

4. 書式	65
書式の定義	65
既製の書式を実際に使用するデータベースにインストールする	65
書式の使用	67
書式を作成する	67
書式とオブジェクトを編集する	68
書式内のオブジェクトのプロパティ	71
グリッドを使う	73
書式のページ設定	73
定期書式の作成	75
モジュールに関連する書式の識別	75
画面上のボタンへの書式の関連付け	76
5. Crystal Reports	77
レポート作成プログラムの操作とインストール	77
レポートの詳細設定	81
Seagate Crystal Reportsのレポートを変更する	83
Crystal Reports統計	83
詳細レポートを作成する	84
レポートを印刷する	86
モジュールに関連するCrystalレポートの識別	87
画面上のボタンへのレポートの関連付け	87
6. コントロールパネル	89
コントロールパネルの作成	89
インジケータの作成	90
コントロールパネルの例	91
7. アクション	95
アクションの定義	95
アクションの作成	97
アクションの例	106
変数を使う	109
アクションをテストする	110
アクションを実行する	110
画面上のボタンへのアクションの関連付け	111
8. メッセージ	113
メッセージシステムの仕組み	113
9. ワークフロー	117
定義	118

概要	119
ワークフローの活用法	121
ワークフローのグラフィカルエディタの使用法	121
ワークフローを使った依頼承認手続き例	124
ワークフローのコンテキスト	141
ワークフローの役割	143
ワークフローのアクティビティ	145
タスク	150
イベント	153
ワークフローのトランジション	164
ワークフローのアラームと制限時間	164
ワークフローの実行グループ	166
ワークフローのトラッキング	167
終了済みワークフローインスタンスの削除	167
技術情報：データモデル	172
10. データの出力およびSQLビューの作成	175
定義	175
AssetCenterデータベースからデータを出力する	176
AssetCenterデータベースでSQLビューを管理する	177
推奨事項	178
出力スクリプトを定義する	178
出力スクリプトを実行する	184
11. スクリプト	187
スクリプトの定義	187
スクリプトの適用	189
関数の概要	189
Basic関数を分類する	193
スクリプト作成の基本	193
スクリプトライブラリ	197
ヒントと注意事項	198
スクリプト例1	203
スクリプト例2	206
12. カレンダー	209
カレンダーの概要	209
カレンダーに影響される機能	210
カレンダーの作成方法	210
カレンダーの作成手順	211
13. タイムゾーン	217
タイムゾーンを管理する理由	217

タイムゾーン機能を実装する	218
タイムゾーンを作成する	218
タイムゾーンを管理する	219
AssetCenter Serverでタイムゾーンを管理する	225
操作に与える影響	226
14. 特殊フィールド	231
特殊フィールドの定義	231
特殊フィールドの用途	232
特殊フィールドを作成する	232
特殊フィールドの使用方法	237
15. ウィザード	239
表記法	239
定義	240
構造モデル	242
ウィザードページのモデル	244
ウィザードに関する一般情報	244
一般的な構造とシンタックス	244
ノードのプロパティ	245
ウィザードの連鎖化	250
Basic関数	251
ROOTノードの定義	252
ROOTノードのシンタックス	252
ROOTノードのプロパティ	252
ROOTノードのサブノード	258
PAGEノードの定義	258
PAGEノードのシンタックス	258
PAGEノードのプロパティ	259
PAGEノードのサブノード	260
TRANSITIONノードの定義	261
TRANSITIONノードのシンタックス	261
TRANSITIONノードのプロパティ	261
TRANSITIONノードの特徴	262
FINISHノードの定義	263
STARTノードの定義	264
TIMERノードの定義	265
倍長整数および文字列ノードの定義	266
コントロールノードの定義	267
コントロールノードの一般的なシンタックス	267
コントロールのタイプと関連プロパティ	267
グラフィカルエディタの使用方法	296
ウィザードの作成例	300
クエリウィザード (QBE) の作成の例	306

ウィザードに関する一般的な質問	311
16. ニュース	317
ニュースの概要	317
ニュースの概要	318
ニュースの重要度	318
配信するメッセージ	318
ニュースの配信先リスト	319
ニュースを表示する	319
インデックス	321

図の一覧表

PEREGRINE

1.1. リストデータの詳細画面	14
3.1. クエリエディタの活用法	26
4.1. 書式のヘッダー	74
7.1. [実行可能ファイル]タイプのアクション	106
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	107
8.1. メッセージシステムの仕組み	114
9.1. ワークフローチャートの概略	118
9.2. AssetCenterのワークフロー - 概要	120
9.3. ワークフローチャート - 依頼の承認	124
9.4. ワークフローチャート - 依頼の承認	138
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	155
9.6. 同期ワークフローチャートの例	162
9.7. 非同期ワークフローチャートの例	163
9.8. 終了イベント付きのワークフローチャート	163
9.9. ワークフローチャートを定義できるメインテーブル	173
9.10. 実行中のワークフローに関連するメインテーブル	173
11.1. スクリプトビルダ	191
12.1. カレンダの詳細画面の[予定表]タブページ	211
12.2. カレンダの詳細画面の[プレビュー]タブページ	215
15.1. ウィザードの構造	243
15.2. 実行およびデバッグ ボタン	297
15.3. ウィザードの編成	301

表の一覧表

PEREGRINE

1.1. システムリストデータの値	15
3.1. 表記法	42
3.2. 論理演算子	46
3.3. 比較演算子	47
3.4. 集合型のAQL関数	54
3.5. 文字列型のAQL関数	55
3.6. 日付型のAQL関数	56
3.7. 日付型のAQL関数	58
3.8. 数値型のAQL関数	59
3.9. テスト型のAQL関数	60
9.1. ワークフローを制限する場合	157
9.2. イベントの様々な処理方法	159
9.3. イベントの様々な処理方法	161
11.1. 使用可能なデータ型	192
14.1. 特殊フィールドのタイプ	233
15.1. 表記法	239
15.2. ROOTノードの論理プロパティ	253
15.3. ROOTノードの物理プロパティ	255
15.4. ROOTノードのサブノード	258
15.5. PAGEノードの論理プロパティ	259
15.6. PAGEノードの物理プロパティ	260
15.7. PAGEノードのサブノード	260
15.8. TRANSITIONノードの論理プロパティ	262
15.9. FINISHノードの論理プロパティ	263
15.10. FINISHノードの物理プロパティ	264

15.11. STARTノードの論理プロパティ	265
15.12. TIMERノードの論理プロパティ	265
15.13. 倍長整数および文字列ノードの論理プロパティ	266
15.14. すべてのコントロールに共通の論理プロパティ	268
15.15. すべてのコントロールに共通の物理プロパティ	270
15.16. CHECKBOXコントロールのプロパティ	272
15.17. COMBOBOXコントロールの物理プロパティ	273
15.18. OPTIONBUTTONSコントロールの物理プロパティ	274
15.19. LISTBOXコントロールの物理プロパティ	275
15.20. LISTBOXコントロールのメソッド	277
15.21. LISTBOX コントロールの必須論理プロパティ	278
15.22. LABELコントロールの物理プロパティ	279
15.23. PROGRESSBARコントロールの物理プロパティ	280
15.24. COMMANDBUTTONコントロールの物理プロパティ	280
15.25. DBLISTBOXコントロールの物理プロパティ	281
15.26. DBQUERYBOXコントロールの物理プロパティ	284
15.27. 標準モードのDBEDITコントロールの物理プロパティ	287
15.28. DBPATHコントロールの必須論理プロパティ	288
15.29. LINKEDITコントロールの論理プロパティ	289
15.30. TEXTBOXコントロールの物理プロパティ	291
15.31. CHARTコントロールの論理プロパティ	291
15.32. CHARTコントロールの物理プロパティ	292
15.33. FILEEDITLコントロールのプロパティ	294
15.34. TICKEDITコントロールのプロパティ	294
15.35. NUMBOXコントロールのプロパティ	295
15.36. COMBOEDITコントロールのプロパティ	295
15.37. コントロールのプロパティ	296
16.1. マーキーツールバーのボタン	320

1 | リストデータ

リストデータは、優先度、場所、国、メーカーなど、AssetCenterの特定のフィールド（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。

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。

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。リストから適切な値を選択するだけで、フィールドに値が入ります。

AssetCenterでは、次の2種類のリストデータを管理します。

- 不特定のリストデータ
- システムリストデータ

これらのリストデータは開くか閉じることが可能です。

不特定のリストデータ

AssetCenterの管理者は、[管理 / リストデータ] メニューから不特定のリストデータにアクセスできます。

次の2種類の不特定リストデータがあります。

- ユーザ自身が作成したリストデータ。ユーザは、このリストデータを任意管理項目にリンクできますが、フィールドにはリンクできません。フィールドへのリンクはプログラムによって自動的に割り当てられます。

- データベースのフィールドにリンクしているリスト。このリストは任意管理項目にもリンクできます。このリストデータを削除した場合やリストデータの名前を変更した場合に、このリストにリンクするはずのフィールドを含むレコードが作成されると、AssetCenterは自動的に元の名前でリストデータを再生します（値は割り当てません）。

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。




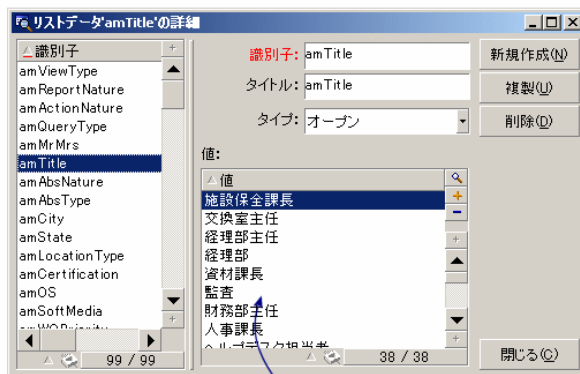
管理者は、リストの右側に表示される 、 および  ボタンを使って値の削除、変更、および追加を行うことができます。

図 1.1. リストデータの詳細画面



フィールドに入力する際に
選択肢として表示される値のリスト

オープンリストデータ

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] (SQL名: seType) フィールドが [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、AssetCenter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できます）。追加する際、作成を確認するメッセージが表示されます。

クローズドリストデータ

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

クローズドリストデータの場合、AssetCenter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。

注意:

リストデータの詳細画面の [作成] または [変更] ボタンをクリックすると、管理者がリストデータに加えた変更がデータベースに保存されます。この変更をクライアントのコンピュータで有効にするには、クライアントのコンピュータでデータベースに接続し直す必要があります。

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、AssetCenterが定義します。管理者もユーザもこれをカスタマイズすることはできません。

これらのリストデータは、[管理/リストデータ] メニューでは編集できません。

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、資産の詳細画面の [割当]（SQL名：seAssignment）フィールド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]

データベースに保存される値	表示される値
1	[在庫中]
2	[除却済]
3	[納品待ち]

システムリストデータを参照するには、次の方法があります。

- システムリストデータで入力したフィールドのヘルプを表示する。
- AssetCenter Database Administratorを使う。
- データベース構造が記述されている「Database.txt」ファイルを使う。このファイルはマニュアルと共にインストールされる。
- 「Database.xml」ファイルを使う。このファイルは、インストール先の「config」フォルダにある。

2 履歴

AssetCenterでは、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、AssetCenter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。以下の手順で指定します。

- 1 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。
- 2 フィールドの設定画面の [全般] タブページを選択します。
- 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:

フィールドの設定画面で [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AssetCenter Database Administrator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、AssetCenter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。

- [変更日] (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名 (ラベル)
- [元の値] (SQL名 PreviousVal) : 変更されたフィールドの元の値 (コメント型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 (コメント型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 [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AssetCenter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) : コメント型のフィールドの修正前の値。コメント型のフィールド (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。

AssetCenter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

- [変更日] : 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

- [変更日] : 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

リンク（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

- [変更日] : 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] : リンク先レコードの参照番号
- [新しい値] : 変更したフィールドの新しい値

リンク（別のテーブルへのnリンク）を削除する

記録される内容

- [変更日] : 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参照番号

- [元の値]：リンク先レコードの参照番号
- [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

AssetCenter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

AssetCenter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にある[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

- [変更日]：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

- [変更日]: 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

- [変更日]: 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AssetCenter Serverの更新時に実行されます。

履歴項目の作成、削除、または変更

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、AssetCenter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AssetCenter 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、AssetCenter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、AssetCenter Exportで実行できます。

はじめに

本節では、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、AssetCenterがAssetCenter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。



AQLを使うには、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

AssetCenter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。

データベースの種類に無関係

AssetCenter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_IDModelId) 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 IModelId, 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_IDModelId) */ M1.IModelId, M1.FullName FROM amModel M1
```

Microsoft SQL ServerまたはSybase SQL Serverでは、次のSQLコードが生成されません。

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

IBM DB2では、次のコードが生成されます。

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

AssetCenterデータベースへのアクセスを簡略化

AQLは、リンクと結合の管理を簡略化します。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 Language）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AssetCenterデータベースに書き込まないでください。

AssetCenterのクエリ

クエリを使うと、特定のテーブルまたはリンクしているテーブル内の情報の選択基準を組み合わせたことができます。

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。
- AssetCenter APIを使う場合。
- AssetCenterをDDEサーバとして使う場合。

AQL（Advanced Query Language）は、AssetCenterに組み込まれたクエリ言語であり、AssetCenter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。

AssetCenter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ーフェースを使って作成
- AQLで直接クエリを記述

警告:

本章後半では、AQLの使い方を分かりやすく説明するために、すべてのAQLシンタックスを使った記述例を紹介します。特にSELECT、WHERE、およびFROM句が説明されています。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を使うことはできません。

クエリエディタ

AssetCenter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、クエリの設計と結果をSQL言語で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。

動作

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

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クエリの作成、変更、削除を実行できます。作成者や他のユーザは、作成されたクエリを状況に応じて使います。

クエリエディタにアクセスする

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] メニューを使う。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。
 - [ツール / クエリ] メニューを選択すると表示されるクエリの詳細画面で直接実行
 - クエリのメインテーブルに表示される専用メニューから、クエリフィルタを選択して実行
- アクセス制限、クエリフィルタ、リストの設定など、AssetCenterのクエリを呼び出す機能を使う。
- AssetCenter Exportなどの外部プログラムを使う。

次のクエリエディタでは、内容に応じてクエリを簡略化できます。

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）は暗黙的に指定します。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 [リストの設定] ポップアップ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。
- テーブルは暗黙的に指定します。

クエリエディタを使ってクエリを作成する

クエリエディタを使ってクエリを作成するには、[ツール/クエリ] メニューを選択します。[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。

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 [作成] をクリックすると、クエリの詳細を指定するタブページを使うことができます。

手順2：[フィルタ (WHERE 句)] タブページでフィルタ条件を定義する

AssetCenter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。

フィルタ条件を定義するには、次の手順に従います。


- 1 [フィールド1] に開始テーブルのフィールド、定数、または式を指定します (比較演算子を使う場合は [フィールド2] フィールドにも同様に指定します)。
- 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 [変更] をクリックし、クエリの作成を確定します。

ANDやORで連結した複数のフィルタ条件を定義するには、次の手順に従います。

- 1 前述のように、最初のフィルタ条件を作成します。
- 2 その他のフィルタ条件を定義し、[AND] ボタンまたは [OR] ボタンで条件を連結します。

3 [変更]をクリックし、クエリの作成を確定します。

 注意:


選択した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削除するか、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ウィンドウに直接AQLでクエリを入力することもできます。

手順3：クエリの実行をプレビューする

クエリをテストし、SQL言語に変換したクエリを表示する手順は以下の通りです。

- 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

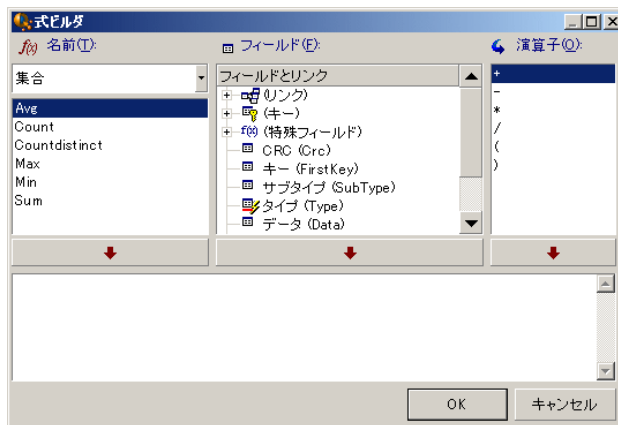
式  を使うとクエリで計算を実行できます。例えばCount関数を使うと、クエリによって検索されたレコード数をカウントできます。

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。
- AssetCenterの式ビルダを使う。

式ビルダを使うには、クエリの詳細画面の [フィルタ (WHERE句)] タブページで、編集領域の隣にある  ボタンをクリックします。


このボタンは、[フィールド1] または [フィールド2] フィールドに [式] パラメータ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。

- [名前] 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 (集計、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。

[名前]、[フィールド]、[演算子] を式に挿入するには、


- 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わり、定義した式が表示されます。

定数


定数 k は、選択基準に割り当てる固定の値です。例えば、メーカーが「3Com」の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、次の手順に従います。

- 1  アイコンをクリックします。

- 2 データベース内の既存の値を表示するウィンドウが表示され、そこから検索条件として指定するフィールドを選択できます。

 **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、この節をお読みください。

この節では、主に次の点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AssetCenterデータベースの特異性

「[AQLのシンタックス \[倭麟2\]](#)」と「[AQLの関数 \[倭麟4\]](#)」の節に、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、[AssetCenter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、マニュアル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

AssetCenter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、AssetCenter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 (SQL名: IModelId)
- モデルの完全名 (SQL名: FullName)
- メーカー (SQL名: amBrand) に関連するテーブルの名前 (SQL名: Name)

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

Oracle SQLまたはMicrosoft SQL Serverで記述した同じクエリは、以下の通りです。

```
SELECT M1.IModelId, M1.FullName, B2.Name FROM amModel M1, amBrand B2 WHERE M1.IBrandId=B2.IBrandId
```

モデル (SQL名: amModel) のテーブルとメーカー (SQL名: amBrand) のテーブル間の2つの結合は、AQLでは自動的に処理されます。AssetCenter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、結合数は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。次の手順に従って、接続の詳細画面で、「amdb.ini」ファイルを変更します。

```
UseSql92Join=1
```

このファイルの場所: ▶ AssetCenter - インストールガイドの.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

AssetCenterデータモジュールには、次の特異性があります。

- 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、「0」（NULLではない）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。

有効性

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。つまりこのレコードは、テーブルAのレコードの内、テーブルBの「0」主キーのレコードにリンクしているものを指します。

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、部署と従業員の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

ここでは、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、Sybase SQL Server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, amEmpIDept e, amLocation l WHERE a.UserId *= e.EMPIDeptID AND e.lLocald *= l.lLocald
```

このコードは複数の外部結合を次々に使うので、Sybase SQL Server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。AssetCenter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, amEmpDept e, amLocation l WHERE a.IUserId = e.IEmpDeptId AND e.ILocald = l.ILocald
```

上記のクエリでは、「ユーザ」（SQL名：User）と「場所」（SQL名：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にアクセスする（リンクがない場合は主キーが「0」のレコードにアクセスする）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を処理す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

AssetCenter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> = NULL

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「!Locald=0」。

Self

Selfは、適用先テーブル名の表記（特殊）文字列に相当する式です。

Selfを使うとクエリが簡単になり、AssetCenter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名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 self FROM amEmplDept
```

これは、次のクエリに相当します。

```
SELECT (((((Name + ',') + FirstName) + '(') + Phone) + ')') FROM amEmplDept
```

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 IPortfolioItemId FROM amPortfolio WHERE User = CurrentUser
```

リンクとして使う

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」形式では、この関数は現在のユーザに対応するレコードにアクセスします。
- 「CurrentUser.Field」形式では、この関数は現在のユーザのフィールド値を返します。

例：データベースに接続しているユーザがアクションをトリガした場合、別のメッセージタイプのアクションが状況に応じて発生するように設定できます。メッセージタイプのアクションは、接続中のユーザに自動的に警告メッセージを送信するとします。これを指定するには、アクションの詳細画面で次のように入力します。

The screenshot shows a configuration window for a message. It has tabs for '説明', 'メッセージ', '任意管理項目', '履歴', and '書類'. The 'メッセージ' tab is selected. Fields include: '参照オブジェクト:' (empty), '優先度:' (普通), '配信済通知' (checkbox), '受信者:' ([CurrentUser.Email]), 'Cc:', 'Bcc:', '件名:' (確認メッセージ), and 'メッセージ:' (アクションがトリガされました.).

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] (SQL名 : 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] (SQL名 : 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通りです。

- 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:**

AssetCenter Database Administrator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、「 [AssetCenterのインストール先フォルダ] /doc/infos」にあります。

階層構造のテーブル

すべての階層構造のテーブルには、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。

パスはスペースなしのスラッシュ (/) 記号で区切られます。パス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 amEmpDept 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 amEmpDept 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 amEmpDept 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。

例えば次のクエリ

```
WHERE location = 0
```

は次のクエリと同等に処理されます。

```
WHERE ILocald = 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、「ILocald」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名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) + ')') FROM 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。特定の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] テーブル (SQL名 : 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名の前に「cf_」を付けて記述してください。

並べ替えとインデックス

AQLで並べ替え（ORDER BY句）を使うクエリには、以下の2通りがあります。

- クエリで指定したインデックスをAssetCenter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AssetCenterで使わないモード。この場合は、DBMSがデータの保存方法を決定します。

例

次のクエリについて説明します。

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

- インデックスを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]（メーカー）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「メーカー」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。
- インデックスを使用するアクセス：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。

インデックスを使う方法

インデックスを使う方法は、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

AssetCenterのリストごとにデータのアクセス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。手順は次の通りです。

- 1 設定するリストに移動します。
- 2 右クリックします。
- 3 ポップアップメニューから [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。別のアクセスタイプを選択する場合は、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句でインデックスを使うことを指定できます。

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、良好に機能しないことがあります。

並べ替え順

並べ替え順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

インデックスを使う場合

- 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）。

インデックスを使わない場合

- 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

並べ替え

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

Microsoft SQL ServerまたはSybase SQL Server

並べ替え順は、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のシンタックス

AQLを使うには、SQL言語に関する知識が必要です。ただし、このマニュアルには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。
<>	山形括弧は、項目の短い説明を示します。実際に項目を入力するときは、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、複数の項目候補を区切るのに使います。
...	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 [修飾8]]
[WHERE句 [修飾9]]
[GROUP BY句 [修飾0]]
[HAVING句 [修飾1]]
[ORDER BY句 [修飾2]]
```

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 [修飾8]]
[WHERE句 [修飾9]]
[GROUP BY句 [修飾0]]
[HAVING句 [修飾1]]
)
```

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 FROM 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グループ化できます。

```
SELECT <選択リスト>
[FROM句 [修飾8]]
[WHERE句 [修飾9]]
[GROUP BY句 [修飾0]]
[HAVING句 [修飾1]]
[ UNION | UNION ALL | INTERSECTS | MINUS]
SELECT <選択リスト>
[FROM句 [修飾8]]
[WHERE句 [修飾9]]
[WHERE句 [修飾9]]
[GROUP BY句 [修飾0]]
[HAVING句 [修飾1]]...
[ORDER BY句 [修飾2]]
```

クエリの要素

フィールドとリンク

クエリには、AssetCenter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。

[Lien. ...[Lien.]]<フィールド>

ポートフォリオ品目（SQL名：AmPortfolio）テーブルの例

Model User.Name User.Location.Name
--

- 参照フィールドとして指定する場合は、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、AssetCenter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]（SQL名：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

次のシンタックスは、クエリで使用できる有効な定数です。

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

一重引用符で囲みます。

例

```
'コンピュータ'
```

```
'モニタ'
```

日付または時刻型の定数

日付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時間制ではありません）。
- 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#
```

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

これらの要素に演算子や括弧を組み合わせると、複雑な式を作成できます。
比較の式のシンタックスは次の通りです。

```
<式> <比較演算子> <式>
```

論理式のシンタックスは次の通りです。

```
<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つの式を連結できます。

表 3.2. 論理演算子

演算子	説明
AND	論理積演算
OR	論理和演算

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] (SQL名: seAssignment) フィールドが [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	「=」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。
NOT LIKE	次のワイルドカード文字を使用できます。 パーセント (%) : すべて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 Anywhere、SQL Server、Sybaseはサポートしますが、Oracle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、文字列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。この演算子で利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	フィールド値がNULLかどうかを調べます。
IS NOT NULL	AssetCenter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+時刻型フィールドがNULLと見なされます。

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AssetCenter Exportクエリでは、「府中支社」で使用しているメーカー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 IModelId, Model.Brand FROM amPortfolio WHERE Model.Brand = ANY (SELECT Model.IBrandID 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。例は以下の通りです。

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

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] (SQL名: amAsset) テーブルの[資産タグ] (SQL名: AssetTag) 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またはSybase 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 <集合を使わない式> [, <集合を使わない式>]...

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、データベース内のメーカーの総数を算出します。このクエリの場合AssetCenterは、1つのメーカーに関連付けられている資産ごとに、1つのメーカー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、メーカーのリストと各メーカーの資産数が算出されます。

```
SELECT Model.Brand.Name, count(IAsId) 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

次の例に示すクエリでは、WHERE句とHAVING句が同等に機能します。

次のクエリは、名前が文字「B」で始まるメーカーのリストと、メーカーごとの資産数を返します。

```
SELECT Model.Brand.Name, count(IAsId) 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(IAsId) 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メーカーをすべて検索できます。

```
SELECT Model.Brand.Name, count(IAsId) 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AssetCenter API AmDbExecAqlに含まれています。

AssetCenter APIの詳細については、マニュアル『プログラマーズリファレンス』の「関数の説明」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(hrAsset, 2)-IDaysBefore*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

```
IErr = AmDbExecAql("insert into amDateAlarm (bSecondLevel, dtTrig1, IActionId, IMonitObjId, MonitoredField, MonitoredTable, sDaysBefore1) values ( 0, " & AmGetFieldLongValue(AmGetFieldLongValue(hrAsset, 2)-IDaysBefore*86400 & ", " & IASTid & ", 'dWarrEnd', 'amAsset', " & IDaysBefore & ")")
```

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 [修飾8]] [WHERE句 [修飾9]]

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", [IPOrdId])
IErr = AmSetFieldLongValue(hr, "seStatus", "$(IDS_POSTATUS_ORDERED)")
IErr = AmUpdateRecord(hr)
```

UPDATE句を使用したアクションのコード

```
IErr = AmDbExecAql("update amPOrder set seStatus = 21 where IPOrdId = " & [IPOrdId])
```

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AssetCenter特有です。

詳細については、マニュアル『はじめに』の「AssetCenter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 [修飾8]] [WHERE句 [修飾9]]

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 [修飾8]] [WHERE句 [修飾9]]

AQLの関数

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

機能	説明
Avg(<列>)	数値型の列項目の平均を返します。列にレコードがない場合は「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。

機能	説明
Max(<列>)	「数値」、「文字列」、または「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」（「文字列」型の列）、または「空の日付」（「日付」型の列）を返します。
Min(<列>)	「数値」、「文字列」、または「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」（「文字列」型の列）、または「空の日付」（「日付」型の列）を返します。
Sum(<列>)	数値型の列値の合計を返します。列にレコードがない場合は、「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

機能	説明
Ascii(<文字列>)	<文字列>の先頭の文字のASCIIの値を返します。
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。
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>をテキストの日付タイプ（ISO国際形式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>)	次のように、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。 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。

機能	説明
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>をテキストデータタイプに変換します。
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<n2>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+時刻タイプのオブジェクトを次の国際規格に合うテキストデータタイプに変換します。 YYYY-MM-DD HH24:MI:SS
TimeToText(<時刻>)	時刻タイプのオブジェクトを次の国際規格に合うテキストデータタイプに変換します。 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

機能	説明
AddDays(<日付>, <数値>)	「日付」または「日付+時刻」型のフィールドに特定の日数を加えます。
AddHours(<日付>, <数値>)	「日付」または「日付+時刻」型のフィールドに特定の時間数を加えます。

機能	説明
AddMinutes(<日付>, <数値>)	「日付」または「日付+時刻」型のフィールドに特定の分数を加えます。
AddSeconds(<日付>, <数値>)	「日付」または「日付+時刻」型のフィールドに特定の秒数を加えます。
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(<時間>)	「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(<時間>)	「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。
NullTimeStamp()	NULL日付および時刻を作成します。 注意: プログラム上、日付および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「日付および時刻」タイプデータに変換します。

機能	説明
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、データベースのマニュアル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、Sybaseまたは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	AssetCenterクエリ言語
先週変更したすべてのレコード	AddDays(dtLastModif,7)>=Getdate()
過去1時間に通知されたすべての作業指示	HoursDiff(Getdate(), dtNotif) <= 1 または AddHours(dtNotif, 1) >= Getdate()
過去30分間に通知され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、オープンした日付とクローズ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(dtActualFixed)) = DayOfYear(DbToLocalDate(dtActualFixed))
```

次のクエリは、今日の日付でオープンした作業指示を検索します。

```
SELECT Self FROM amWorkorder WHERE DayOfYear(DbToLocalDate(dtActualFixed)) = 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>,)	「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 $0 \leq r < q$)。
NullNumeric()	NULL数値を作成します。
<p>注意:</p> <p>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。</p>	
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」を小数点以下第「n」位で四捨五入します。
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。 <ul style="list-style-type: none"> 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NULL (=0) です。
TextToNumber(<テキスト>)	<テキスト>を数値データタイプに変換します。
Trunc(<a>, <n>)	「a」を小数点以下第「n」位で切り捨てます。

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

機能	説明
IsNull(<a>,)	「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

本節では、各例につきクエリ作成の1側面が説明されています。以下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。これらのクエリを実際にテストする場合は、AssetCenter Exportを使います。これらのクエリをデータベースの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'
```

AssetCenter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：メーカーが「Compaq」であるすべてのポートフォリオ品目

```
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」に直接リンクしてい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
```

階層構造のテーブル

[FullName] フィールドを使う

例：名前が「大阪支社」である場所の、すべての関連場所

```
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

階層構造のテーブルのクエリでは、[FullName] フィールドと [sLvl] フィールドが頻繁に使用されます。

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階層レベルは「0」です。

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/大阪支社/%') AND (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 amEmpIDEpt WHERE (Title = '会計部長') AND (Location.Name = '府中支社')
```

数値、日付、テキスト型フィールドの比較

例：2003年1月1日から2003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 >= #2003-01-01 00:00:00#) AND (dtActualFixStart <= #2003-12-31 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。空の文字列を一重引用符で囲みます。

```
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.IAstId FROM amAsset A1 WHERE A1.IModelId = amModel.IModelId))
```

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.IModelId FROM amModel A1 WHERE A1.IParentId = amModel.IModelId))) 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.IModelId) FROM amModel a WHERE a.IParentId = IModelId)) 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 [IModelId] を選択し、 [IParentId] 識別子が [IModelId] 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 IModelId FROM amModel WHERE (FullName LIKE (p.FullName + '%/') AND (Nature.Name = 'コンピュータ'))))
```

注意:

AssetCenter Export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：「Peregrine」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

クエリは次の通りです。

```
SELECT Self FROM amEmpIDept WHERE (Trainings_1.Title = 'Peregrine') 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 amEmpIDept WHERE (Trainings.Title = 'Peregrine') 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 amEmpIDept WHERE (Trainings.Title = 'Peregrine') 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 portfolio
```

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, amModel FullName
```



4 | 書式

本章では、AssetCenterの書式を作成する方法について説明します。
書式のリストを表示するには、[ツール / レポート機能 / 書式] メニューを選択します。

書式の定義

書式は、データを印刷するための書類様式です。
書式は、Seagate Crystal Reportsとは異なり、AssetCenter内で直接作成できます。

既製の書式を実際に使用するデータベースにインストールする

AssetCenterには既製の書式が付属しています。書式はデモ用データベースにインストールされていますが、実際に使用するデータベースを作成または更新する際に、書式をこの本番用データベースに挿入する必要があります。

データベース作成時の書式のインポート

データベース作成時に書式をインポートするには、

- 1 AssetCenter Database Administrator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 AssetCenter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 [アクション/データベースの作成]メニューを選択します。
- 6 『管理』マニュアルの、「AssetCenterデータベースの作成、変更、および削除」の章の指示に従ってフィールドに値を入力し、データベースを作成します。

ページ [インポートするデータ] で、[書式] を選択します。

既存のデータベース内に書式をインポートする

書式を既存のデータベースにインポートするには、

- 1 AssetCenter Database Administrator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 AssetCenter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 [アクション/データベースの作成]メニューを選択します。
- 6 ウィザードのページに次のように入力します(ウィザードのページを移動するには [次へ] または [戻る] 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	書式をインポートしたい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。
[Use advanced creation options] (詳細な作成オプションを使用)	このオプションは選択しないでください。

[作成パラメータ] ページ:

フィールド	値
パスワード	<p>管理者のパスワードを入力します。</p> <p>注意:</p> <p>AssetCenterデータベース管理者は、[名前] (Name)フィールドが「Admin」に設定されている[部署と従業員] (amEmplDept)テーブル内のレコードです。</p> <p>データベース接続のログインは[ユーザ名] (UserLogin)フィールドに保存します。管理者名は「Admin」です。</p> <p>パスワードは[パスワード]フィールド (LoginPassword)に保存します。</p>

[インポートするデータ] ページ:

フィールド	値
使用できるデータ	[書式] オプションを選択します。
[エラー発生時にインポートを停止]	このオプションは、問題が発生したときにインポートを停止する場合に選択します。
[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([終了] ボタン)。

書式の使用

書式を使用するためには、関連テーブルを表示してから [ファイル/印刷] コマンドで書式を選択する必要があります。

書式を作成する

書式を作成するには、[ツール/レポート機能/書式] メニューを選択して、書式のリストを表示します。

基本的な情報

- 1 書式の名前を入力します。
- 2 書式のタイプ（リストまたは詳細）を選択します。
 どちらのタイプの書式にも、テキストと事前に定義した画像を入力できます。
 2つのタイプは、次の点が異なります。
 - リスト：フィルタを適用するなどして現在表示しているレコードのリストを印刷します。印刷されるデータは、現在表示している列項目のデータです。
 - 詳細：選択している1個のレコードの詳細画面のフィールド（例：資産の詳細画面）の値と、そのレコードにリンクしているレコードのリスト（例：その資産の付属品）を印刷できます。
- 3 書式のメインテーブルを選択します。
- 4 確定



警告:





[テーブル]（SQL名：TableName）フィールドでテーブルを指定すると、作成した書式がそのテーブルのリストだけに適用されます。



書式とオブジェクトを編集する

書式を編集するには、まずオブジェクトを定義してから、そのオブジェクトをページ内に配置します。

ページに新しいオブジェクトを挿入するには、次の手順に従います。

- 1 [書式] タブページを選択します。
- 2 [書式] タブページの左側に表示されるオブジェクトのアイコンをクリックします。

アイコン	機能
	書式内のオブジェクトを選択します（変更する場合など）。
	印刷するレコードに依存しない固定テキストと変数（現在の日付など）を追加します。
	画像を追加します。
	フィールド値と固定テキストの文字列を含む計算式を挿入します。

アイコン	機能
	書式内のオブジェクトを選択します（変更する場合など）。
	<p>レコードのリストを挿入します。</p> <p>このツールを使うと、ページ内にリストを配置できます。</p> <p>詳細タイプの書式では、印刷するレコードとフィールドを含んでいるリンクテーブルを定義できます。</p>

- 3 ページ内にマウスポインタを置きます。
- 4 ページ内でクリックします。
- 5 マウスでドラッグしてフレームを描きます。このフレームはオブジェクトを配置するためのスペースです。
- 6 オブジェクトを配置するフレームをダブルクリックします。オブジェクトのプロパティを指定できるウィンドウが表示されます。
- 7 オブジェクトのプロパティを定義します。
- 8 [変更]をクリックします。

書式には次のオブジェクトを挿入できます。

- 固定テキスト
- 計算式
- リスト
- 画像

固定テキスト

固定テキストは、印刷するレコードに依存しないテキストです。様々な文字や次に示す変数と組み合わせることができます。

\$D 印刷日

\$U：書式を印刷するAssetCenterユーザの名前

\$C：ページ番号

\$N：印刷するページ総数

 **警告:**

テキストは引用符で囲まないでください。

例

印刷日:\$D 担当者:\$U

計算式

計算式は、詳細タイプの書式でのみ使用できます。

計算式で次の情報を組み合わせることができます。

- AssetCenter データベースのフィールド値
- 引用符で囲んだ固定テキスト

例

```
"資産:" CodInt " AssetTag/ " Brand" Model.Brand.Name
```

計算式を使って計算することはできません。

リスト

- リストタイプの書式：1つのリストを挿入できます。このリストは、[ファイル/印刷]メニューを選択した時にアクティブウィンドウのリストと置き換わります。
- 詳細タイプの書式：リストの数に制限はありません。リストには、現在のレコードに関連するすべてのレコード（資産のすべての付属品など）が表示されます。

 注意:

リストタイプの書式を作成する場合は、印刷するフィールドを選択できません。指定したテーブルのリストに表示される列項目のフィールド値が印刷されます。詳細タイプの書式を作成する場合は、印刷するフィールドを1つずつ選択できます。

画像

画像（ロゴなど）を挿入できます。

書式内のオブジェクトのプロパティ

位置とサイズ

オブジェクトの位置を変更するには、オブジェクトをドラッグします。オブジェクトのサイズを変更するには、オブジェクトの端をドラッグします。

 **注意:**

複数のオブジェクトを同時に選択すると、オブジェクトの位置とサイズを一度に変更できます。複数のオブジェクトを一度に選択するには、オブジェクト全体の周りをマウスでなぞって選択フレームを描きます。[Ctrl]キーを押したまま、オブジェクトを1つずつ連続してクリックして選択する方法もあります。

プロパティ

オブジェクトをダブルクリックするか、または[内部用書式/プロパティ]メニューを選択すると、選択したオブジェクトのプロパティのリストが表示されます。

表示されるプロパティは、書式に挿入されたオブジェクト(テキスト、画像、リスト、または計算式)オブジェクトによって決まります。

プロパティリストには2つの列があります。1つ目の列にはプロパティ名が表示され、2つ目の列ではプロパティの値を変更できます。

プロパティを変更するには、2つ目の列をクリックします。

単純なプロパティ(テキスト、計算式、リスト、背景色、文字の色、行揃え、オブジェクトの配置)の場合は直接変更できます。複雑なプロパティ(テキストや計算式のフォント、枠、リストの内容、画像)の場合は、別のウィンドウが表示されます。

テキストプロパティ

テキストを直接入力します。

計算式プロパティ

計算式の特許文字列ウィンドウには、計算式を簡単に作成できるように、選択している書式のタイプと互換性のあるフィールドだけをツリー構造で表示するドロップダウンリストがあります。ツリー内のノードをクリックすると、現在の

フィールドがリスト内で選択したフィールドと置き換わります。複数のフィールドを連続して指定するには、各フィールドを2重引用符で囲みます。

背景色プロパティ、文字の色プロパティ

16色から選択できます。

行揃えプロパティ


フレーム内のテキストの行揃えを指定します。ドロップダウンリストから、左揃え、中央揃え、または右揃えを選択できます。

ページ内の配置プロパティ


ページ内のテキストの横の位置を指定します。ドロップダウンリストから次の位置を選択できます。

- 左揃え
- 右揃え
- 中央揃え
- 両端揃え：この場合は、書式作成画面で定義した位置を保持します。

フォントプロパティ

オブジェクトのフォントとサイズを選択するには、プロパティ画面の編集セルの右端にある  ボタンをクリックします。

枠プロパティ

オブジェクトの周囲に境界線を追加するには、プロパティ画面の編集セルの右端にある  ボタンをクリックします。罫線の選択画面が表示されます。

3D効果を選択すると、境界線に浮き出し効果を付けることができます。

3D効果を使わない場合は、上下左右の境界線、境界線の色および太さを選択できます。

画像プロパティ

画像を挿入するには、次の手順に従います。

- 1 プロパティ画面の編集セル内のボタンをクリックします。
- 2 [画像の選択]画面でグラフィックファイルを選択します。

リスト内のリンクプロパティ

書式内に表示するリストを選択するには、ドロップダウンリストから希望するリスト（従業員が使う資産のリストなど）を選択します。

リストの内容プロパティ

注意:

リストの内容は、詳細タイプの書式を編集する場合にのみ設定できます。

リストの内容を定義するには、プロパティ画面の編集セル内のボタンをクリックします。設定ウィンドウが開き、リスト内のすべての列項目が表示されます。

各列項目ごとに、次の内容を定義できます。

- 列のタイトル
- 列の内容を定義する計算式
- 列のサイズ（その列がリスト内に占めるパーセント）
- 列のタイトルと内容の行揃え
- 列のタイトルまたは内容の個々の行揃え
- 縦と横の境界線

リストから列を削除するには、[Delete]キーを使います。

リストに列を挿入するには、リストの最後の列項目の次のセルに新しい値を入力します。

各セルは、プロパティのリストと同じ原理で編集できます。

グリッドを使う

グリッドは、画面の背景に表示される縦と横の目盛線です。オブジェクトの配置に使用します。

書式の詳細画面の [内部用書式 / グリッド] ポップアップメニューを使って、次の操作を行うことができます。

- グリッドの表示と非表示の切り替え
- グリッドの目盛幅の定義

横と縦の線の交点のみが表示されます。線の間隔によって、ページにオブジェクトを配置するときの精度が決まります。

書式のページ設定

書式の詳細画面の [書式 / ページ設定] ポップアップメニューを使って、次のフォーマットを定義できます。

- ページサイズ

- 縦または横のレイアウト
- 書類のマージン（余白）
- ヘッダーとフッター

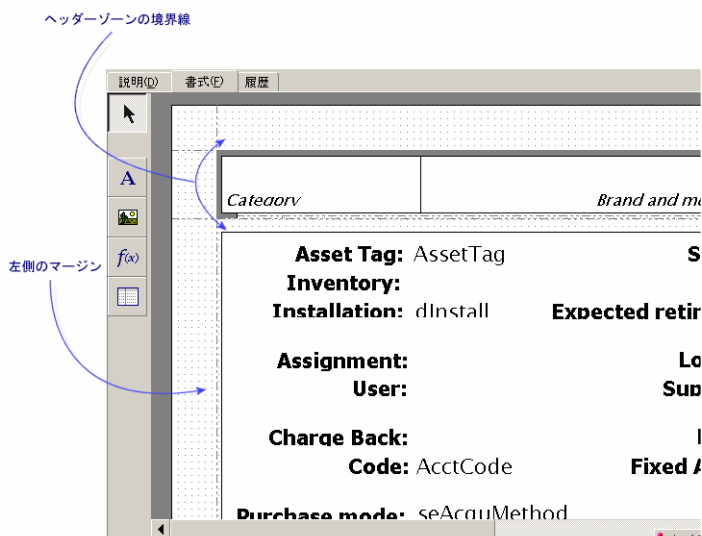
 注意:

書類の余白部分にあたるフッターおよびヘッダーのサイズは、編集ゾーンで直接変更できます。マージンマーカーや点線で表示されるヘッダーまたはフッターゾーンの境界線をドラッグしてください。

ページのヘッダーとフッターにテキストを挿入するには、次の手順に従います。

- 1 [書式/ページ設定]メニューを選択します。
- 2 [ヘッダー]および[フッター]チェックボックスをオンにします。
- 3 [OK]をクリックします。
- 4 書式のヘッダーゾーンまたはフッターゾーンに移動します（ヘッダーゾーンとフッターゾーンは点線の境界線で示されます）。
- 5 オブジェクトをこれらのゾーンで直接挿入するか、または本文のページゾーンからオブジェクトを移動します。
- 6 [変更]をクリックして変更を確定します。

図 4.1. 書式のヘッダー



 注意:

ヘッダーゾーンとフッターゾーンから本文のページゾーンにオブジェクトを移動することはできません。

定期書式の作成

書式を使って、レポートのページレイアウトを設定します。

定期的に必要なレポートを作成するには、次の方法をお奨めします。

- 1 必要な情報を表示できるビューを作成します。
- 2 このビューを書式にリンクします。

ビューには、次の情報を記憶させることができます。

- 並べ替えの基準
- 適用するフィルタとフィルタの値
- 表示する列項目のリスト

レポートを印刷するには、

- 1 事前に作成したビューを表示します（ [ツール / ビュー] メニューを使います）。
- 2 表示したビューから印刷を行います（ [ファイル / 印刷] メニューを使います）。このときに、必ずレポートの適切なタイプと適切な書式を選択します。

モジュールに関連する書式の識別

あるモジュールに関連する書式を識別するには、

- 1 AssetCenterを起動します。
- 2 書式のリストを表示します（ [ツール / レポート機能 / 書式] ）。
- 3 リスト内で右クリックします。
- 4 表示されたショートカットメニューで [ユーティリティ / リストの設定] を選択します。
- 5 リストの列見出しにリンク [ドメイン]（Domain）を追加します。
- 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 モジュールの書式はドメインの名前で識別できます。

例：/調達/書式/

画面上のボタンへの書式の関連付け

画面上のボタンに書式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。

- マニュアル『ユーザインタフェース』の「クライアントワークステーションのカスタマイズ」の章の「ボタンのカスタマイズ」の節。
- マニュアル『管理』の「データベースのカスタマイズ」の章の「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 / アクションボタンの作成」の節。
- マニュアル『管理』の「データベースのカスタマイズ」の章の「新規オブジェクトの作成 / アクションボタンの作成」の節。



5 Crystal Reports

本章では、AssetCenterでレポートを編集する方法について説明します。
レポートのリストを表示するには、[ツール/レポート機能/レポート]メニュー
を選択します。

レポート作成プログラムの操作とインストール

概要

AssetCenterでは、レポート作成プログラムであるCrystal Reportsを使います。このソフトウェアで作成したレポートのファイル拡張子は「.rpt」です。

 注意:

使用可能なレポートのリストとコメントは、AssetCenter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/pt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。

限定版のCrystal Reports

AssetCenterのインストール時に適切なオプションを選択すれば、限定版のCrystal Reportsと一緒にインストールされます。

AssetCenterデータベースの現在のデータを既存のレポートを使ってプレビューおよび印刷するには、この限定版のCrystal Reportsで十分です。

Crystal Reportsの作成または変更

AssetCenterで直接新しいレポートを作成することはできません。

新規レポートを作成するには、Crystal Reportsの完全バージョンをインストールする必要があります。制約点は以下の表の通りです。

サポートされるCrystal Reportsバージョン	バージョン9.xと10 日本語版：バージョン7.0
サポートされるCrystal Reportsの言語	<ul style="list-style-type: none"> ■ フランス語 ■ 英語 ■ ドイツ語 ■ スペイン語 ■ イタリア語 ■ 日本語 ■ ポーランド語 <p>Crystal ReportsとAssetCenterではそれぞれ別の言語を使用できます。</p>

Crystal ReportsにサポートされるODBCドライバ ODBC AssetCenterドライバ

Crystal Reports完全版またはランタイムのインストール

AssetCenterのインストールプログラムでは、Crystal Reportsランタイムバージョンをインストールできます。既存のレポートを印刷するにはこのバージョンで十分です。AssetCenterのインストール時に適切なオプションをオンにします。

既にランタイムバージョンがインストールされている場合、またはCrystal Reportsの完全版をインストールする場合は、ランタイムバージョンをインストールする必要はありません。

既製のCrystal Reportsレポートを実際に使用するデータベースにインストールする

AssetCenterには既製のレポートが付属しています。レポートはデモ用データベースにインストールされていますが、実際に使用するデータベースに挿入する必要があります。

注意:

既製レポートはAssetCenterの以下のバージョンには付属していません。

- 日本語版
- ポーランド語版

レコードを1つずつ挿入するには、

- 1 AssetCenterを起動します。
- 2 実際に使用するデータベースを開きます。
- 3 [ツール/レポート機能/レポート]メニューを選択します。
- 4 新しいレポートを作成します。
- 5 レポートの詳細画面で [ファイル] タブページを選択します。
- 6 [インポート] ボタンをクリックします。
- 7 AssetCenter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の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 AssetCenter Database Administrator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 AssetCenter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「AssetCenter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内にレポートをインポートする

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 AssetCenter Database Administrator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 AssetCenter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(ウィザードのページを移動するには[次へ]または[戻る]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	レポートをインポートしたい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。
[Use avanced creation options] (詳細な作成オプションを使用)	このオプションは選択しないでください。

[作成パラメータ]ページ:

フィールド	値
パスワード	<p>管理者のパスワードを入力します。</p> <p>注意:</p> <p>AssetCenterデータベース管理者は、[名前] (Name)フィールドが「Admin」に設定されている[部署と従業員] (amEmplDept)テーブル内のレコードです。</p> <p>データベース接続のログインは[ユーザ名] (UserLogin)フィールドに保存します。管理者名は「Admin」です。</p> <p>パスワードは[パスワード]フィールド (LoginPassword)に保存します。</p>

[インポートするデータ] ページ:

フィールド	値
使用できるデータ	[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停止]	このオプションは、問題が発生したときにインポートを停止する場合に選択します。
[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([終了] ボタン)。

レポートの詳細設定

レポートのリストを表示するには、[ツール/レポート機能/レポート]メニューを選択します。

レポートの管理

注意:

[ファイル] (SQL名:FileName)フィールド (つぎの文書で説明します) を直接編集することはできません。このフィールドには、[インポート] を使ってインポートしたレポートのファイルの完全名 (ファイルへのパスと拡張子付き) が入ります。

次に、レポート機能に関連するボタンについて説明します。

- [インポート] : このボタンは、レポートの詳細画面に表示されます。このボタンをクリックすると外部のレポートをインポートできます (初回はレポートの作成、2回目以降はレポートの変更として処理されます)。外部のレポートのファイル拡張子は「.rpt」です。外部のレポートをインポートすると、レポートの詳細画面の [ファイル] (SQL名:FileName) フィールドにそのファイルのファイル名が入ります。
 - [出力] : このボタンは、レポートの詳細画面に表示されます。このボタンをクリックすると、AssetCenterデータベース内のレポートから「.rpt」形式のファイルを作成できます。デフォルトでは、ダイアログボックスが開き、[ファイル] フィールドに入っているファイル名が表示されます。「.rpt」形式のファイルを作成すると、外部のレポート作成プログラムを使ってレポートを変更できます。
 - [プレビュー] : このボタンにも、[ファイル/印刷] メニューからアクセスできます。このボタンをクリックすると、レポートを印刷する前にプレビューすることができます。
 - [印刷] : このボタンにアクセスするには、[ファイル/印刷] メニューを選択します。このボタンをクリックするとレポートを印刷できます。
-

注意:

[プレビュー] ボタンまたは [印刷] ボタンをクリックすると、AssetCenterはデータベース内のレポートから一時ファイルを作成します。このファイルはCrystal Reportsの印刷エンジンで処理されます。処理されると、一時ファイルは直ちに消去されます。表示または印刷されるデータは、現在データベース内に存在するデータです。

Seagate Crystal Reportsのレポートを変更する

AssetCenterデータベース内のレポートを変更するには、Crystal Reportsプログラムが必要です。

レポートを変更するには、次の手順に従います。

- 1 AssetCenterで [ツール / レポート機能 / レポート] メニューを選択して、レポートの詳細画面を表示します。
- 2 [出力] ボタンをクリックし、「.rpt」形式のファイルとして出力します。
- 3 出力した「.rpt」レポートをCrystal Reportsで変更して保存します。
- 4 AssetCenterで [ツール / レポート機能 / レポート] メニューを選択して、もう一度レポートの詳細画面を表示します。
- 5 変更した「.rpt」ファイルをインポートして更新し、レコードを変更します。

Crystal Reports統計


自動的に更新されるCrystal Reportsレポートを表示するには、[ツール / レポート機能 / Crystal Reports統計] メニューを選択します。

[ツール / レポート機能 / レポート] メニューを使って表示するのと同じレポートを表示できます。

属性


表示するレポートの属性を指定します。このフィールドの右のフィールドで、特定のレポートを選択できます。選択できるレポートは、選択する属性によって異なります。

[自動更新] ボタン

このボタンのアイコンは、 です

- このボタンをクリックすると、レポートが更新されます。
- このボタンをマウスの右ボタンでクリックすると、レポートを自動的に更新する頻度を定義できます。

[ズーム] ボタン

このボタンのアイコンは、 です。
拡大 / 縮小率 (3段階) を変更します。

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。

利用例

- 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 [IAstId] で、[amAssetId] ではありません。

計算式 [CurrentUserId] (大文字と小文字を区別) を使って、レポートを印刷するユーザを特定します。印刷時に、この計算式がAssetCenter 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: IEmplDeptId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、AssetCenter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。

{amAsset.IAstId} = {@amAssetId}

このようにして設定したレポートをAssetCenter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リストを表示します。
- 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] (SQL名: TableName) フィールドにそのテーブルの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

詳細レポート



注意:

詳細レポートの内容と、Crystal Reportsで詳細レポートを作成する方法: ▶ [詳細レポートを作成する \[倭融4\]](#)。

AssetCenterで詳細レポートを印刷するには、

- 1 詳細レポートの印刷が必要なレコードのリストを表示します（ [ポートフォリオ / 資産とロット] など）。
- 2 印刷するレコードを複数選択します。
- 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] をクリックします。

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。

規定レポート



注意:

リストとグラフのレポートを「規定レポート」と呼びます。

「詳細レポート」と違い、テーブルはレポートのコンテキストにはなりません。

AssetCenterで規定レポートを印刷するには、

- 1 [ファイル / 印刷] メニューを選択します。
- 2 [タイプ] フィールドを [規定レポート (Crystal Reports)] に設定します。
- 3 レポートを選択します。
- 4 [印刷] をクリックします。

リストまたはグラフのレポートが印刷されます。

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 AssetCenterを起動します。
- 2 レポートのリストを表示します（ [ツール / レポート機能 / レポート] ）。
- 3 リスト内で右クリックします。
- 4 ポップアップメニューから [リストの設定] を選択します。
- 5 リストの列見出しにリンク [ドメイン] （ Domain ） を追加します。
- 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/

画面上のボタンへのレポートの関連付け

画面上のボタン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。

- マニュアル『ユーザインタフェース』の「クライアントワークステーションのカスタマイズ」の章の「ボタンのカスタマイズ」の節。
- マニュアル『管理』の「データベースのカスタマイズ」の章の「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 / アクションボタンの作成」の節。
- マニュアル『管理』の「データベースのカスタマイズ」の章の「新規オブジェクトの作成 / アクションボタンの作成」の節。

6 | コントロールパネル

AssetCenter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タイプ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 [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]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。また、コントロールパネルの[ドメイン]をその他の共通アクションプロパティと共に定義します。

 **注意:**

詳細については、このマニュアルの「[アクション \[修繕5\]](#)」の章を参照してください。

- 3 [インジケータ] タブで、コントロールパネルに追加したいインジケータを選択します。インジケータの追加または削除は、とで行うこともできます。
- 4 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。

インジケータの作成

レポートと統計という2種類のオブジェクトをAssetCenter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、統計の作成についてのみ詳細を説明します。Crystalレポートの作成については、このガイドの「[Crystal Reports \[倭献7\]](#)」の章を参照してください。

インジケータを作成するには、

- 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、[レポート] または [統計] を選択します。

統計ドロップダウンリストで ボタンをクリックして表示された詳細画面で [新規作成] ボタンを押すと、統計作成画面にアクセスできます。

統計を作成するには、

- 1 統計の [名前] を定義します。
- 2 統計で使用するデータとデータによるグループ化に使用する値を含むテーブルを指定します ([グループ化] フィールド)。
- 3 [表示] フレームのオプションを使用して統計の表示設定を定義します。

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
- 場所ごとの従業員の内訳
- 部署ごとの従業員の内訳
- 社内の勤務年数別従業員の内訳

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：インジケータを作成する

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、4つの統計用インジケータを連続して作成できます。

- 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。以前に定義されたインジケータのリストから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ドロップダウンリストで  ボタンをクリックしてから、[新規作成] をクリックして新しい統計を作成します。

男性 / 女性の内訳

この統計では、従業員に関係したすべての情報を含む [部署と従業員] テーブルのデータを使用します。また、統計では、会社内の男性と女性の内訳を表示する

必要があります。したがって、[グループ化]基準は[性別](MrMrs)フィールドに基づく必要があります。

サンプルデータから部署(同じテーブル内で定義)を除外するためにフィルタを使用します。

- 1 [フィルタ]フィールドの横にあるドロップダウンリストの🔍ボタンをクリックしてから、[新規作成]をクリックして新しいフィルタを作成します。
- 2 フィルタの[名前]を選択します。例えば、「Employees only」(従業員のみ)を選択します。
- 3 フィルタ詳細画面の[クエリ]タブに、次のクエリを入力します。

```
bDepartment = 0
```

- 4 [追加]をクリックしてから、[選択]をクリックします。

統計用の表示設定を選択します。この例では、3Dセクタを使用します。

場所ごとの従業員の内訳

この統計では、従業員に関係したすべての情報を含む[部署と従業員]テーブルのデータを使用します。また、統計では、場所ごとの従業員の内訳を表示する必要があります。したがって、[グループ化]基準は「場所」(Location)リンクの値に基づく必要があります。そのため、[グループ化]フィールドには、Location.FullNameが含まれます。

以前に定義したフィルタ(「Employees only」(従業員のみ))もサンプルデータから部署を除外するために使用します。

統計用の表示設定を選択します。この例では3D縦線グラフを使用します。

部署ごとの従業員の内訳

この統計では、従業員に関係したすべての情報を含む[部署と従業員]テーブルのデータを使用します。また、統計では、部署ごとの従業員の内訳を表示する必要があります。したがって、[グループ化]基準は「部署」(Parent)リンクの値に基づく必要があります。そのため、[グループ化]フィールドには、Parent.Nameが含まれます。

以前に定義したフィルタ(「Employees only」(従業員のみ))もサンプルデータから部署を除外するために使用します。

統計用の表示設定を選択します。この例では3D縦線グラフを使用します。

社内の勤務年数別従業員の内訳

この統計では、従業員に関係したすべての情報を含む[部署と従業員]テーブルのデータを使用します。また、統計では、社内の勤務年数別従業員の内訳を表示する必要があります。この統計(会社に勤務している従業員の勤務年数)に対する[グループ化]の値は、データベース内で直接使用することができません。一方、[部署と従業員]テーブルの[雇用日](dHire)フィールドには、この値を計

算するのに必要なすべての情報が含まれます。特殊フィールドは、このために使用され、[グループ化]の値として使用されます。

特殊フィールドを作成するには、

- 1 [管理 / 特殊フィールド]メニューを選択して、新しいレコードを次に示す情報で作成します。

フィールド	値
タイトル	当社での勤務年数
テーブル	部署と従業員
フィールドタイプ	AQL
結果のデータ型	数値
BASICシンタックス	$\text{Round}((\text{DaysDiff}(\text{GetDate}(), \text{dHire}) / 365), 0)$ AQLクエリでは、DaysDiff()関数を使用して、次の間の期間を計算します。 <ul style="list-style-type: none"> ■ 計算した時刻: GetDate()関数 ■ 雇用日: 雇用日 (dHire)フィールドの値 DaysDiff()関数からは日単位の期間が返ります。結果が365(1年の平均日数)で除算され、Round()関数で小数点以下が切り捨てられます。

- 2 [作成]をクリックして作成を確定します。
- 3 特殊フィールド画面を閉じて、[部署と従業員]テーブルを開きます([ポートフォリオ / 部署と従業員]メニュー)。
- 4 以前に作成された特殊フィールドを次の手順でリストに追加します。
 - 1 リストを右クリックし、状況依存メニューの[ユーティリティ / リストの設定]を選択します。
 - 2 特殊フィールドを選択して、リストに追加します。
 - 3 変更箇所を確定します。
- 5 特殊フィールドに該当する列が部署と従業員の画面に表示されることを確認します。

これで統計の作成を終了できます。

以前に定義したフィルタ(「Employees only」(従業員のみ))もサンプルデータから部署を除外するために使用します。

統計用の表示設定を選択します。この例では曲線を使用します。

手順3：コントロールパネルを開いてデータ内を移動する

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、機能とお気に入りの枠で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、フィルタの値として [男性] を使用して [性別] フィールドのデータにフィルタが実行されます。

 **注意:**

レポートでは、ナビゲーションは使用できません。



7 | アクション

本章では、AssetCenter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。

アクションの定義

アクションは、AssetCenterから直接実行できるプログラムを呼び出す操作です。
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] : AssetCenter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]
- [導入]
- [アクション]
- コントロールパネル

メニューのリストからアクションを選択できるようにするには、まずこれらのアクションを定義する必要があります。

 **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

AssetCenter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当たります。

機能ドメインは、機能とお気に入りの枠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。アクション用に機能ドメインを選択すると、アクションは、機能とお気に入りの枠の機能ドメインの名前の下に表示されます。

 **注意:**

機能とお気に入りの枠の内容は状況（コンテキスト）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、従業員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アクションのコンテキスト（この例では従業員のテーブル）に一致する時のみ、アクションが表示されます。

機能ドメインを定義するには、次の手順に従います。

- 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。この名前は機能とお気に入りの枠内に表示されます。AssetCenter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:**

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

AssetCenterでは、各種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
- 導入
- 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。その他の文書の場合も、拡張子がファイルマネージャ内のアプリケーションに関連付けられていれば開くことができます。

[DDE] アクション

[DDE] アクションは、DDE要求を処理できる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要求を送ります。

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。AssetCenter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、次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。

- AssetCenter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、AssetCenterで次の情報を指定する必要があります。

- AssetCenterデータベースを開いた従業員（ [部署と従業員] テーブル ）の詳細画面の、 [メッセージ] タブページ / [アカウント] （SQL名：MailLogin）フィールドおよび [パスワード] （SQL名：MailPassword）フィールドで、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] （SQL名：EMail）フィールドで、メッセージを受け取るためのアドレスを指定します。

AssetCenter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、 [ログイン] フィールドと [パスワード] フィールドを指定します。

 **注意:**

AssetCenter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AssetCenter Server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、AssetCenter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
- AssetCenterデータベース内のオブジェクト（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、AssetCenter 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、AssetCenter APIから抽出した関数を使う必要があります。
- アクションのコンテキストとしてテーブルを指定した場合は、Set()関数を使うことができます。シンタックスは次の通りです。

```
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は、本マニュアルの「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、AssetCenter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、レポートや書式を印刷できます。

この種のアクションでは以下のフィールドに値を入力します。

- [タイプ] (SQL名: seFormType) フィールド: 印刷する文書タイプ (レポートまたは書式) を指定します。
- [レポート] または [書式] フィールド: どのレポートまたは書式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、レポートまたは書式に定義されているコンテキストにより決定されます。

[導入] アクション

[導入] アクションでは、コンピュータのテーブル内で定義されているコンピュータのリスト上に、導入を起動できます。

このアクションは、コンピュータのテーブル内の状況依存アクションです。

この種のアクションでは以下のフィールドに値を入力します。

- 導入ワークフロー: 作成またはインポートした導入ワークフローを選択します。
- 導入サーバ: 導入のアプリケーションサーバを選択します。

導入の実施や設定の詳細については、マニュアル『Desktop Administration』を参照してください。

[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、指定されたクエリに応じて別のアクションを実行できます。

この種のアクションでは以下のフィールドに値を入力します。

- [実行するアクション] : 起動するアクションを指定します。
- [選択クエリ] : アクションがどのコンテキストで実行されるのかを指定するクエリのスクリプトを入力します。

[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、次の手順に従います。

- 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]
 - DDE.
 - [メッセージ]
 - [スクリプト]
 - [ウィザード]
 - [印刷]
 - [導入]
 - [アクション]
 - [コントロールパネル]
- 5 必要に応じて、アクションの詳細画面の[SQL名] (SQL名: SQLName)フィールドに入力します。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 (AssetCenter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 次のようにして、[コンテキスト] (SQL名: ContextTable)フィールドに入力します。

- 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 必要に応じて、 [ドメイン] フィールドでアクションの所属先の機能ドメインを指定します。アクションは、機能とお気に入りの枠内のこのドメイン名の下に表示されます。
 - 8 必要に応じて、アクションにアイコンを付けます。
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。アクションリストから選択してアクションを実行すると、そのアクションのアイコンがツールバーに表示されます。
 - 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:

AssetCenter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

[DDE] アクションに関する情報は、アクションの詳細画面の [DDE] タブページに入力します。

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 [DDE] に設定した場合にのみ表示されます。

DDE機能は、ソフトウェアに内蔵されているサービス機能を使います。DDE機能を実行するには、コマンドの実行対象となるトピックを定義する必要があります。

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マニュアルを参照してください。
- [トピック] (SQL名 : DDETopic) フィールドに、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] (SQL名 : DDECommand) フィールドに、外部のアプリケーションが実行する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] (SQL名: ActionFile) フィールドに、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コマンドに回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を使う例は以下の通りです。

```
[FileOpen("c:tmp/test.txt")]
```

- アクションのコンテキストを指定する場合は、変数を使ってデータベースのフィールド値を参照します。この場合は、変数も大括弧で囲むので、AssetCenter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。前述の例を使うと、次のように記述してアクションのコンテキストを指定できます。

```
[/FileOpen("c:tmp/test.txt")]
```

また、次のようにコマンドと変数を組み合わせることもできます。この例では [資産] テーブルをコンテキストとして指定します。

```
[/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] タブページに入力します。

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、AssetCenter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、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場合は、[参照オブジェクト] (SQL名 : RefObject) フィールド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、AssetCenter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:**

AssetCenter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通りです。

<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 : AssetCenter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。

アドレスの例は以下の通りです。

- AM:Admin
- MAPI:CathyBernard@taltek.com
- VIM:Cathy Bernard / TALTEK
- SMTP:cbernard@taltek.com

<AssetCenter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AssetCenter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AssetCenter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AssetCenter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、AssetCenter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

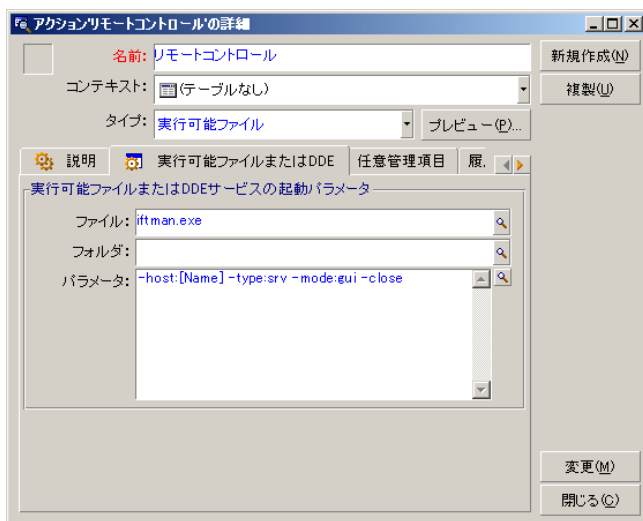
ここでは、AssetCenterによるアクションの実行例を紹介します。

- [実行可能ファイル] タイプのアクション例
- [DDE] タイプのアクション例
- [メッセージ] タイプのアクション例
- [スクリプト] タイプのアクション例

[実行可能ファイル] タイプのアクション例

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た例です。このアクションは、AssetCenter Serverを起動してデータベース記述ファイルの「Acdemo」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。

- AssetCenter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例

この例では、資産のリストから、ある資産の使用者に資産のリース期間が満了間近であることを知らせるメッセージを送信します。この資産は、取得方法が[リース]であり、リース期間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リース明細にするには、アクションを次のように設定します。

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

The screenshot shows the configuration for a 'Message' type action. The '名前' (Name) field is set to '期間満了'. The 'コンテキスト' (Context) is '資産 (am.Asset)'. The 'タイプ' (Type) is 'メッセージ'. The '参照オブジェクト' (Reference Object) is 'AcquContract.'. The '優先度' (Priority) is '普通'. There is an unchecked checkbox for '配信済通知'. The '受信者' (Recipient) is '[Portfolioitem.User.Email]'. The 'Cc' and 'Bcc' fields are empty. The '件名' (Subject) is '契約の期間満了'. The 'メッセージ' (Message) field contains the text: '契約[AcquContract.cf_self]は[AcquContract.dEnd]に期間満了します。'.

[スクリプト] タイプのアクション例


[スクリプト] タイプのアクションの作成では、AssetCenter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、マニュアル『プログラマーズリファレンス』、「関数の用途別一覧」の章、「組み込み関数」の節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以下の通りです。

- 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、Basicスクリプトのビルド画面を表示します。スクリプトタイプのアクションで使用可能なSuccess()関数には、明示的な戻りコードは不要です。以下の例では、次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。

フィールドのラベル	フィールドのSQL名	フィールドの値
[名前]	Name	PC
[作成]	seBasis	ポートフォリオ品目
[接続可能]	blsCnxClient	このチェックボックスはオンになっています。

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, "blsCnxClient", 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 [blsCnxClient]=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AssetCenter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]、[DDE]、および [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] (SQL名: ContextTable) フィールド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、そのアクションのボタ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] (SQL名: ContextTable) フィールド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。
- ポップアップメニュー (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、ポップアップ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

リストから選択した複数のレコードに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可能です。

「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 (移動先のページはトランジションの設定によります)。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、ウィザードの実行 (および関連するアクション) を完全に取り消すことができます。

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。

- マニュアル『ユーザインタフェース』の「クライアントワークステーションのカスタマイズ」の章の「ボタンのカスタマイズ」の節。
- マニュアル『管理』の「データベースのカスタマイズ」の章の「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 / アクションボタンの作成」の節。
- マニュアル『管理』の「データベースのカスタマイズ」の章の「新規オブジェクトの作成 / アクションボタンの作成」の節。

8 | メッセージ

AssetCenterでは、次の2種類のメッセージを管理できます。

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

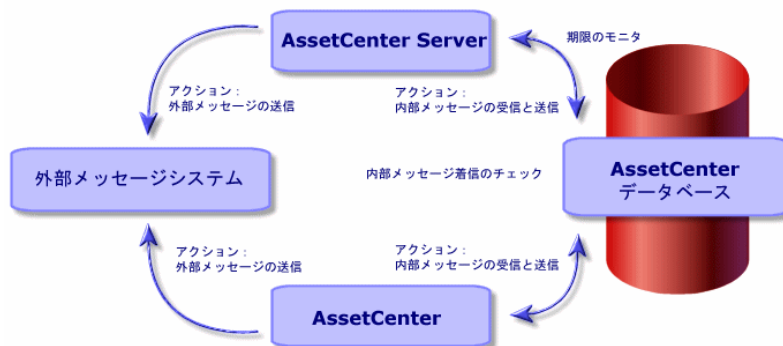
メッセージシステムの仕組み

AssetCenterは以下のタイプのプロトコルを使用してメッセージの送信を管理します。

- AM (AssetCenter)
- SMTP
- MAPI
- VIM

AssetCenterは、受信時にはAM (AssetCenter) 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、次の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。
- AssetCenter Serverによって自動的にトリガされる。
- AssetCenter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 [惨献03]の節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

AssetCenter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メッセージを参照できます。

配信済通知

注意:

AssetCenter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、AssetCenter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。

AssetCenterのワークフローシステムには次の特長があります。

- ワークフローチャートを使用して、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。
- 形式の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、次のようなプロセスのモデルを作成し、プロセスを自動的に処理することができます。

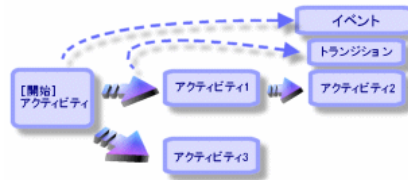
- 購入依頼の承認手続き
- 資産の移動
- その他

AssetCenterでワークフローチャートを作成するには、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）
- トランジション（イベントによって生成される）
- アクティビティ（トランジションによってトリガされる）
- コンテキスト
- 制限時間とアラーム

図 9.1. ワークフローチャートの概略



定義

ここでは、ワークフローで使われるいくつかの主要な用語について説明します。

- ワークフローインスタンスの定義
- 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
- ワークフローのトランジション
- 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の定義

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、ワークフローチャート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、AssetCenter 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トランジション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。ワークフローのイベントにより、別のアクティビティを引き起こすトランジションが起動されます。

ワークフローのトランジション

ワークフローのトランジションにより、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。トランジションは、イベントにより発生します。

1つのイベントを複数のトランジション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、ワークフローアクティビティから発生するタスクを実行する人です。

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チャートを分類できます。ワークフローチャートが属する実行グループは、ワークフローチャートの詳細で指定されます。

概要

AssetCenter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 / ワークフローチャート] メニューを選択して、グラフィカルエディタを表示し、ワークフローチャートを定義します。ワークフローチャートでは、アクティビティ、

イベント、およびトランジションを定義します。これらは、AssetCenter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。

ワークフローチャートは、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、AssetCenter ServerまたはAssetCenterのエージェントで実行されます。

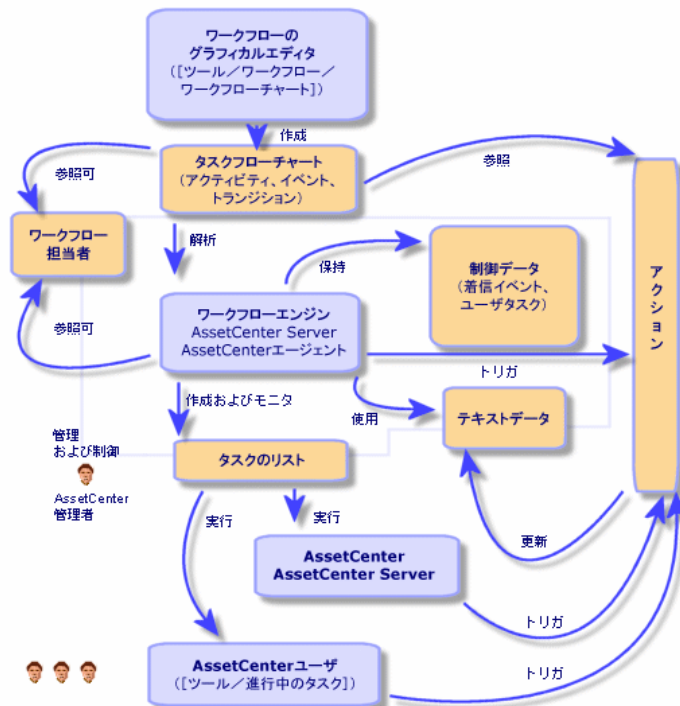
ワークフローエンジンは、イベントに反応して、ワークフローをトリガし、ワークフローインスタンスをモニタします。

- 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得して、イベントの推移を記録する。

ワークフローのタスクは、ワークフローエンジンまたはAssetCenter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。

AssetCenterでワークフローを実行する際の概要は、次の図の通りです。

図 9.2. AssetCenter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。

ワークフローの活用法

AssetCenterのワークフローを使って、会社の手続きを実装することができます。手順は以下の通りです。

- 1 自動化する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チャートを作成します。
 - 1 アクティビティ、イベント、およびトランジション
 - 2 アラーム
- 4 ワークフローの適切な実行グループを指定します。各ワークフローチャート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AssetCenter Serverを起動します。各AssetCenter Server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法

ワークフローチャートにアクセスするには、[ツール/ワークフロー/ワークフローチャート]メニューを使います。このメニューを使用できるのはAssetCenterの管理者だけです。


ワークフローチャートの詳細画面の[アクティビティ]タブページでは、ワークフローチャート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、ワークフローチャートを作成、変更、または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
- トランジション
- 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、次のいずれかの操作を行います。

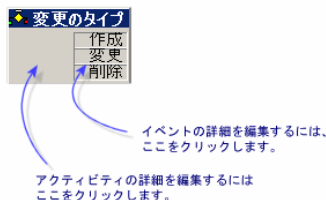
- [アクティビティ] タブページの空白の領域をマウスの右ボタンでクリックして、ポップアップメニューから [アクティビティの追加] コマンドを選択します。アクティビティの詳細画面が表示されます。
-  ボタンをクリックし、次にグラフィカルエディタ領域をクリックします。アクティビティの詳細画面が表示されます。

アクティビティを削除するには、次のいずれかの操作を行います。

- 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、ポップアップメニューから「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、次のいずれかの操作を行います。

- アクティビティをマウスの右ボタンでクリックし、ポップアップメニューから [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、次に示す2種類があります。

[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、次のいずれかの操作を行います。

- 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、次のいずれかの操作を行います。

- 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、ポップアップメニューから [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、次のいずれかの操作を行います。
- イベントをダブルクリックします。
 - イベントを選択し、ポップアップメニューから [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。

トランジション

トランジションを作成するには、次の手順に従います。

- 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。

トランジションを削除するには、次のいずれかの操作を行います。

- トランジションをクリックして選択し、[Delete] キーを押します
- トランジションを選択し、ポップアップメニューから [削除] コマンドを選択します
- トランジションを選択し、 ボタンをクリックします。

トランジションの詳細設定を変更するには、次の手順に従います。


- 1 トランジションをクリックして選択します。
- 2 ポップアップメニューから [トランジションの詳細] を選択します。

トランジション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 トランジションを選択します。
- 2 変更するソースまたはターゲットをドラッグします。

その他の機能

グラフィカルエディタを使って、次の操作を行うこともできます。

- リンクしているアクティビティとトランジションのドラッグ&ドロップ
- [ズーム]スライダまたは  ボタンを使ったフローチャートの拡大と縮小

ワークフローを使った依頼承認手続き例

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 [[修献24](#)]
- 2 必要条件 [[修献25](#)]
- 3 アクティビティを作成する [[修献31](#)]
- 4 アクティビティと同時に作成されるイベントのパラメータ [[修献36](#)]
- 5 開始イベントを作成する [[修献37](#)]
- 6 トランジションを作成する [[修献38](#)]
- 7 ワークフローの起動例 [[修献38](#)]

目的

このワークフローチャートを使って、次の手順の購入依頼プロセスを自動化します。

図 9.3. ワークフローチャート - 依頼の承認



ワークフローチャートの手順は次の通りです。

- 1 ワークフローインスタンスは、購入依頼が承認されるべき状況になると（つまり、購入依頼の [依頼ステータス]（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。

- 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が終わる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。
- 3 責任者の承認をもらったら、財務承認手続きに移ります。
それ以外の場合は、依頼は却下されます。(3b)
- 4 財務承認では、依頼を会社の会計監査役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。
- 5 会計監査役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 [依頼ステータス] 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] に設定されます。

必要条件

「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](#) [俥献13])。

ワークフローの担当者を作成する

このワークフローチャートに関わる担当者は次の通りです。

- 依頼者の部署の責任者
- 会社の会計監査役、木村五郎氏

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承認されています。

これらのワークフローの担当者は [ワークフローの役割] (SQL名 : amWfOrgRole) テーブル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。これを定義するには、役割の詳細画面で次のように入力します。

The screenshot shows the '役割 '依頼者の部署の責任者' の詳細' (Role Details for 'Requester's Department Supervisor') dialog box. The '役割' (Role) list includes '依頼内容の承認', '責任者の承認', '財務の承認', and '依頼者の部署の責任者'. The '依頼者の部署の責任者' role is selected. The 'コンテキスト' (Context) is 'amRequest'. The '役割' (Role) is '依頼者の部署の責任者', and the '参照番号' (Reference Number) is 'D001001'. The 'コンテキスト' (Context) is '依頼 (amRequest)', and the 'タイプ' (Type) is '特定の個人に割当'. The 'スクリプト' (Script) field contains 'RetVal=[Requester.Parent.Supervisor]'. The dialog box has buttons for '新規作成(N)' (New), '複製(L)' (Copy), '削除(D)' (Delete), and '閉じる(C)' (Close).

会計監査役

この担当者は木村五郎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。

The screenshot shows the '役割 '会計監査役' の詳細' (Role Details for 'Accounting Auditor') dialog box. The '役割' (Role) list includes '責任者の承認', '財務の承認', '依頼者の部署の責任者', and '会計監査役'. The '会計監査役' role is selected. The 'コンテキスト' (Context) is 'amRequest'. The '役割' (Role) is '会計監査役', and the '参照番号' (Reference Number) is 'D001002'. The 'コンテキスト' (Context) is '(テーブルなし)', and the 'タイプ' (Type) is '個人に割当'. The '担当者' (Assignee) is '木村五郎'. The dialog box has buttons for '新規作成(N)' (New), '複製(L)' (Copy), '削除(D)' (Delete), and '閉じる(C)' (Close).

アクションを作成する

ワークフローチャートでは、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依頼

このアクションは、技術検証段階で使われます。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。

アクション: 技術検証依頼の詳細

名前: 技術検証依頼

コンテキスト: ワークフローのタスク (amWfWorkItem)

タイプ: メッセージ プレビュー(P)...

説明

メッセージ 任意管理項目 履歴 書類

参照オブジェクト:

優先度: 普通

配信済通知

受信者: [Assignee.Email] ✓ ✕ 🔍

Cc: 🔍

Bcc: 🔍

件名: 購入依頼の検証 🔍

メッセージ: 購入依頼の申込書の再確認をお願いします。 🔍

変更(M)

キャンセル

会計監査役に送られる財務検証依頼

「財務検証依頼」アクションは、財務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、財務検証担当者に送信されます。

The screenshot shows a dialog box titled "アクション財務検証依頼の詳細". It contains the following fields and controls:

- 名前:** 財務検証依頼
- コンテキスト:** ワークフローのタスク (amWfWorkItem)
- タイプ:** メッセージ (with a "プレビュー(P)..." button)
- 説明** (gear icon), **メッセージ** (envelope icon), **任意管理項目**, **履歴**, **書類** (tabs)
- 参照オブジェクト:** (empty dropdown)
- 優先度:** 普通
- 配信済通知
- 受信者:** [Assignee:EMail]
- Cc:** (empty)
- Bcc:** (empty)
- 件名:** 購入依頼の検証
- メッセージ:** 購入依頼の申込書の再確認をお願いします。
- Buttons:** 作成(C), 作成(R) (with icon), キャンセル

購入依頼の承認

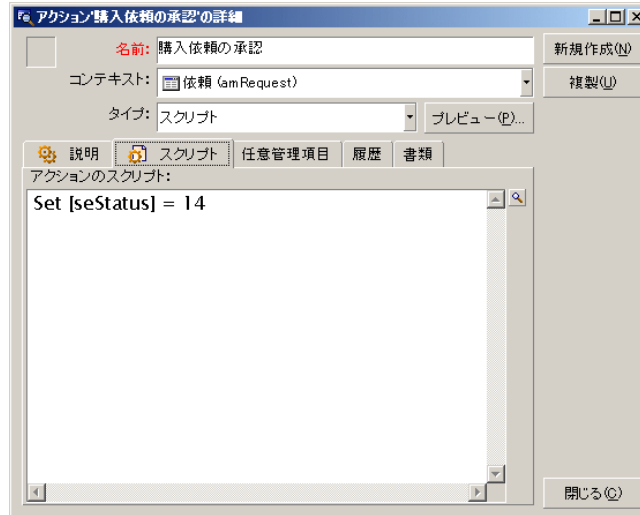
このアクションは、後に作成する「依頼の承認」アクティビティの時点で使用されます。

このアクションにより、購入依頼の詳細画面の [依頼ステータス] (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。

[依頼ステータス] 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 [依頼ステータス] フィールドをマウスの右ボタンでクリックします。
- 2 ポップアップメニューから [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。

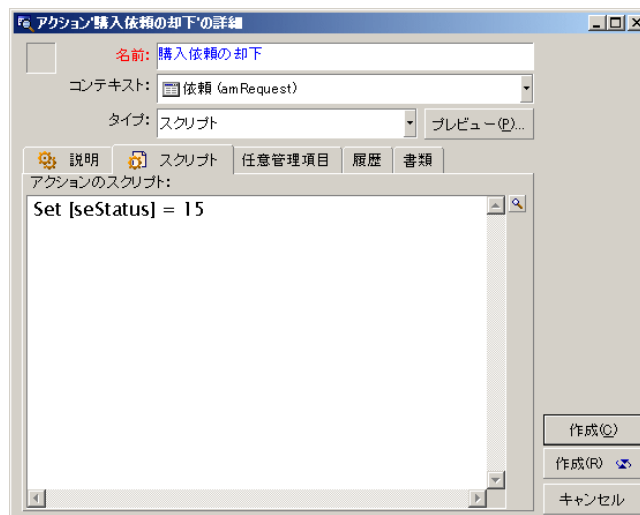
アクションの詳細は次のようになります。



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」アクティビティの時点で使用されます。

「購入依頼の承認」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 [依頼ステータス] (SQL名: seStatus) フィールドは、[却下] になります。



カレンダーを作成する

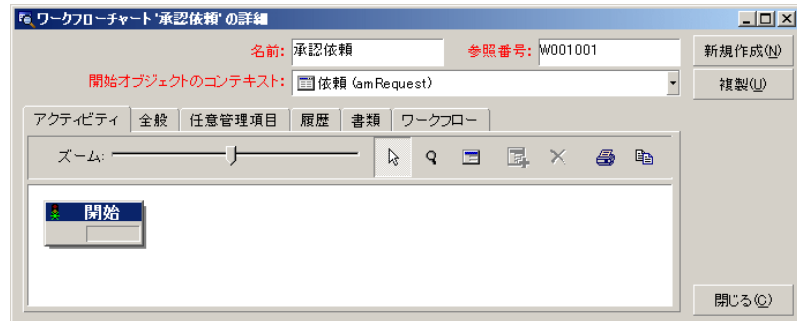
カレンダーのリストには、[ツール / カレンダー] メニューからアクセスできます。このカレンダーはワークフローチャートのアクティビティに関連付けられます。これにより、ワークフローチャートの処理の締切りを設定できます。



ワークフローチャートを準備する

- 1 [ツール / ワークフロー / ワークフローチャート] 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 デフォルトで、ワークフローチャートのすべてのアクティビティに適用されるフローチャートの開始テーブルが表示されます。この例では、[依頼] (SQL名: amRequest) のテーブルです。

- 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、ワークフローチャートの詳細画面の [アクティビティ] タブページ内で、グラフィックモードで作成できます。

- 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 ポップアップメニューから [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] (SQL名: seType) フィールド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」アクティビティのパラメータ

- 1 [パラメータ]タブページに次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、サブタブのラベル部分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] (SQL名: Assignee) フィールドに質問の宛先を指定します。この場合、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、前の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] (SQL名: bNotifAssignee) フィールドを [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、前の準備段階で作成した「技術検証依頼」アクションです。このアクションは、購入依頼の技術検証を開始すると、自動的にトリガされます。

「技術検証」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 [制限時間] 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ーを指定します。このカレンダー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「日本カレンダー」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。

ワークフローチャート「承認依頼」のアクティビティ

名前: 技術検証 テンプレート: [検索] [戻る] 新規作成(N)

タイプ: 質問 参照番号: A001003 複製(U)

全般 パラメータ 時間 アラーム 履歴 書類

カレンダー: 標準カレンダー - 日本

解決時間

期限のタイプ: 時間 0時間

閉じる(C)

「技術検証」アクティビティのアラームを定義する

[制限時間] 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「技術検証依頼」アクションをトリガすることにします。

ワークフローチャート「購入依頼の承認」のアクティビティ

名前: 技術検証 テンプレート: [検索] [戻る] 新規作成(N)

タイプ: 質問 参照番号: REQ APPR-ACTIV02 複製(U)

全般 パラメータ 時間 アラーム 履歴 書類

アラーム

名前: 検証の依頼

参照番号: REQ APPR-FUNCALRM01

トリガするアクション

タイプ: タスク終了前の時間

制限時間: 0時間

起動: アクション

アクション: 技術検証依頼

検証の依頼

閉じる(C)

ポップアップメニューの [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、AssetCenterエージェントは、これらを [ワークフローの基本イベント]（SQL名：amWfOccurEvent）のテーブルに記録します。

AssetCenterまたはAssetCenter Server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、AssetCenter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、AssetCenter Server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。

「財務検証」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。

ワークフローチャート「依頼の承認」のアクティビティ

名前: 財務検証 テンプレート: [目]

タイプ: 質問 参照番号: A001011

全般 パラメータ 時間 アラーム 履歴 書類

タスクを保存

コンテキスト: [依頼 (am Request)]

ワークフロー: 依頼の承認 [目]

入力条件: OR

コメント:

[追加(D)]
[追加(A)]
[キャンセル]

- ワークフローの担当者は異なります。この場合、会社の会計監査役である木村五郎氏です。この担当者は、前の手順で会計監査役の役割と共に [ワークフローの役割] 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。

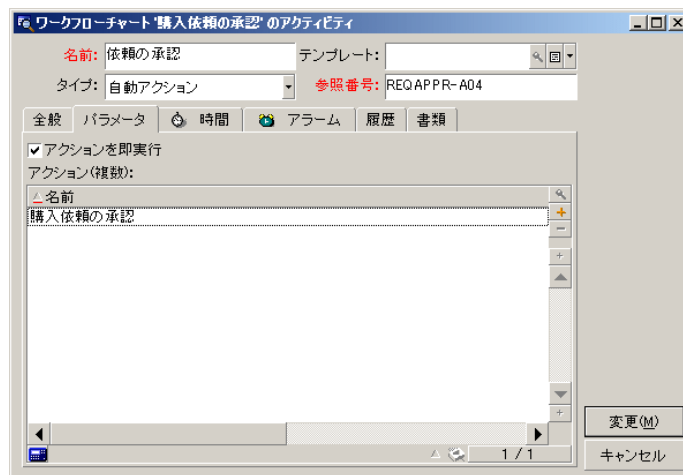
「依頼の承認」アクティビティを作成する

依頼が2つの検証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となります。

ワークフローチャートは、「依頼の承認」アクティビティで終了できます。

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。

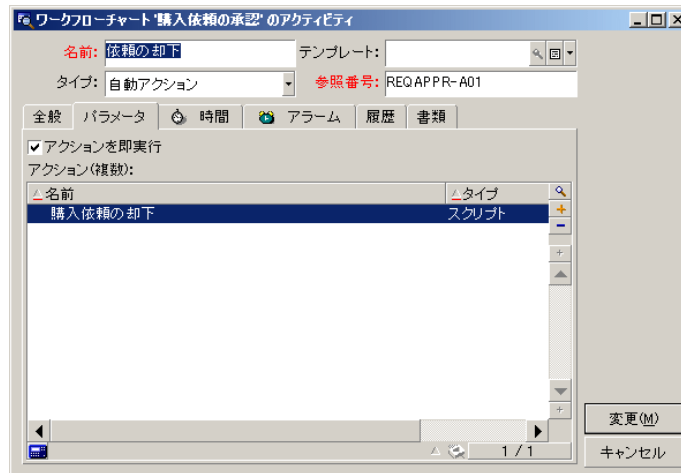
このアクティビティは [自動アクション] タイプのアクティビティです。実行されるアクションは、前の手順で作成した「購入依頼の承認」アクションです。



「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。

この場合は、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「購入依頼の却下」アクションです。



アクティビティと同時に作成されるイベントのパラメータ

前段階でアクティビティが作成されると、以下のイベントも同時に作成されます。

- 「技術検証」アクティビティ
 - 「承認」イベント
 - 「却下」イベント
- 「財務検証」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

AssetCenter Serverに依存しないようにするために、各イベントを選択し以下の操作を実行します。

- 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] タブページを選択します。
- 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。

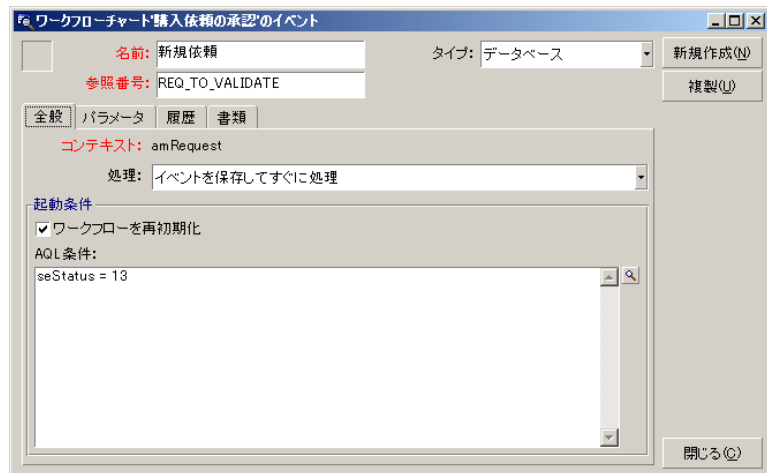
開始イベントを作成する

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領域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。

トランジションを作成する

アクティビティを作成したら、トランジションでリンクする必要があります。

トランジションを作成するには、次の手順に従います。

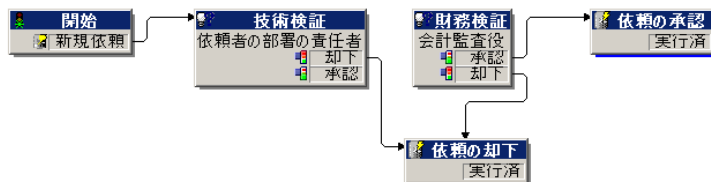
- 1 トランジションの開始イベントをクリックします。
- 2 マウスボタンを押したまま、ターゲットアクティビティまでイベントをドラッグします。

この例では、次のトランジションを作成します。

- 開始イベントから「技術検証」アクティビティまで
- 「技術検証」アクティビティの「承認」イベントから「財務検証」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
- 「技術検証」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、ワークフローチャートは次のようになります。

図 9.4. ワークフローチャート - 依頼の承認



ワークフローの起動例

ワークフローチャート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 [俊献38]
- 2 承認を受ける購入依頼を作成する [俊献39]
- 3 ワークフローを制御する [俊献39]

[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、業務を遂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ィールを指定します（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プロフィール「依頼者」を使用します。
- 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「鈴木俊彦」です。
- 5 後の操作を簡単にするために、「鈴木俊彦」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]（SQL名：bAdminRight）チェックボックスをオンにします。鈴木俊彦の [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:Suzuki」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 AssetCenter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]（SQL名：Requester）フィールドで、「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]（SQL名：seStatus）フィールドを [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。

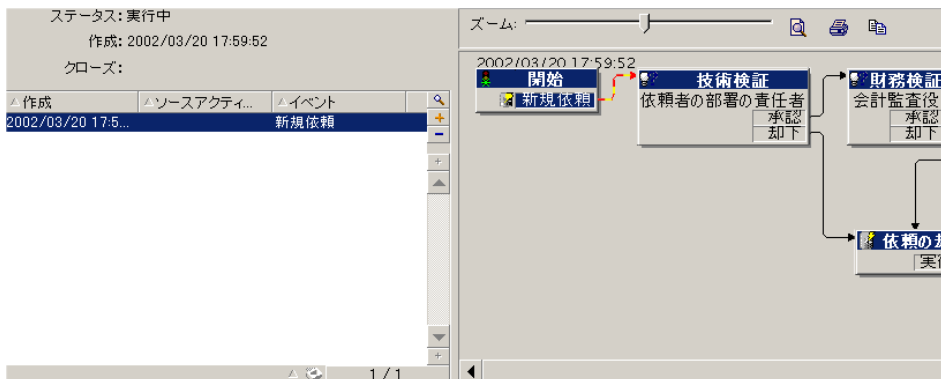
ワークフローを制御する

ワークフローが正常に機能することを確認するために、AssetCenterを起動し、「田中伸一」の部署の責任者である「鈴木俊彦」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

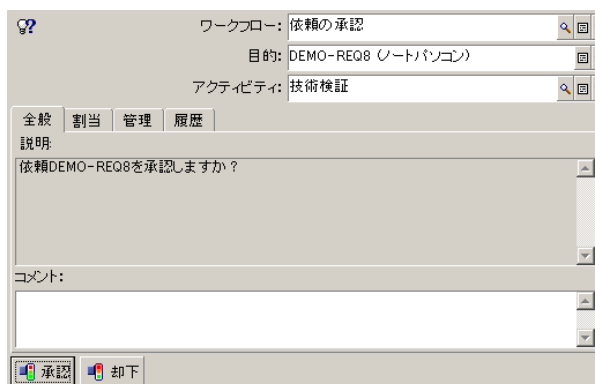
作成した購入依頼の詳細画面を表示します。[ワークフロー]タブページには、依頼の現在のワークフローが一覧表示されます。各ワークフローの詳細は、個々のサブタブページに表示されます。

- 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。ワークフローチャートの「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「鈴木俊彦」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 [制限時間] タブページで指定した情報と、タスクの作成日（つまりトランジションの起動日）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

各ワークフローには固有のコンテキストがあります。

ワークフローのコンテキストを定義する

ワークフローチャートを定義する場合は、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、トランジション、およびワークフローのアクティビティのコンテキスト（トランジションの詳細画面、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、ワークフローチャート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。

ワークフローが参照するオブジェクト

レコードが、ワークフローチャートで定義された起動条件を満たしたときに、ワークフローがトリガされます。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、およびワークフローのトランジション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進行中のワークフローを制限する

[1つの物件に付き1つのワークフローを起動] オプション (SQL名: bUniqueActive)

AssetCenterでは、ワークフローチャートの詳細画面の [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] 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。

あるオブジェクトで2番目のワークフローを起動する「開始」アクティビティの出力イベントが発生した場合、[1つの物件に付き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] タブページ内) によって結果が決まります。

オプションの選択に応じた結果は、次の表の通りです。

		ワークフローチャートの [全般] タブページの [1つの物件に付き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がある場合、そのワークフローは停止し、新しいワークフローが開始します。	
	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は開始しません)。	新しいワークフローが作成されます。

適用例

購入依頼の承認処理を目的とするワークフローチャートの場合は、次のように設定すると便利です。

- [1つの物件に付き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、[ツール/ワークフロー/役割] メニューを選択します。

ワークフローの役割のタイプ

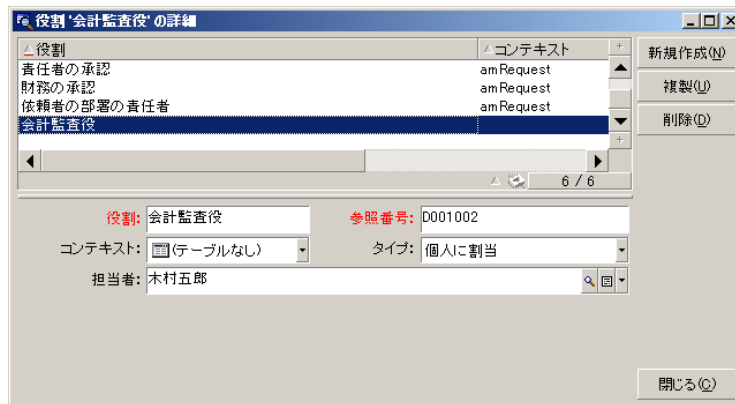
ワークフローの役割には次のタイプがあります (ワークフローの役割の詳細画面 / [タイプ] (SQL名: seType) フィールドで指定)。

- [個人に割り当]
- [特定の個人に割り当]
- [グループに割り当]
- [計算されたグループ]

[個人に割り当]

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



[特定の個人に割当]

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



[グループに割当]

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。

[計算されたグループ]

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] (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、AssetCenter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トランジション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、ポップアップメニューから [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、AssetCenter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トランジション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:**

「Executed」(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、AssetCenterまたはAssetCenter Server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、実行され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が表示されます。

 注意:

「Executed」（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

アクティビティを発生させるトランジション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。トランジションをトリガするイベント用に選択した処理モードに従って、AssetCenter ServerまたはAssetCenterのエージェントがアクションを実行します。

- [アクションを即実行]（SQL名：bExecImmediately）オプションをオンにした場合は、アクティビティを発生させるトランジション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AssetCenter Server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

「自動アクション」タイプのアクティビティは、AssetCenterまたはAssetCenter Server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には、スクリプトとその実行結果の選択肢を定義します。

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、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、ポップアップメニューから[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:

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

アクティビティを発生させるトランジション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。トランジションをトリガするイベント用に選択した処理モードに従って、AssetCenter ServerまたはAssetCenterのエージェントがアクションを実行します。

- [アクションを即実行] (SQL名: bExecImmediately) オプションをオンにした場合は、アクティビティを発生させるトランジション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。トランジションを起動するイベント用に選択した処理モードに従って、AssetCenter ServerまたはAssetCenter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AssetCenter Server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、ワークフローチャートの開始点です。

開始アクティビティは必須で、ワークフローチャート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、ワークフローチャート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、AssetCenter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]タブページにある[入力条件]（SQL名：selnCond）フィールドに入力する必要があります。これは、アクティビティを起動するトランジションに関する条件です。

- アクティビティを起動するトランジションが1つだけの場合は、（AssetCenterまたはAssetCenter Serverによって）トランジションを起動するだけで、アクティビティがトリガされます。
 - アクティビティを起動するトランジションが複数ある場合は、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、すべてのトランジション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、トランジションを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、連続した中間[テスト/スクリプト]タイプのアクティビティを作成して代用することもできます。

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

タスクを発生させるトランジションが起動すると、トランジション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: WkEle) 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] 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]、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 [制限時間] 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]、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、タスクを生成するトランジションの起動元のワークフローエンジンによって実行されます。このエンジンは、AssetCenter ServerまたはAssetCenterエージェントです。

それ以外の場合は、AssetCenter Server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定期的に検証され、必要に応じて実行されます。

AssetCenter Serverがワークフロー関数をモニタする頻度は、AssetCenter Serverのオプションで定義します。

進行中のタスクのリストを表示する

実行する必要があるタスクのリストを表示するには、 [ツール/進行中のタスク] メニューを選択します。

リストには、データベースに接続しているユーザに応じて、次のタスクが表示されます。

- AssetCenter管理者は、すべてのワークフローの進行中の全タスクを表示できます。
- ワークフローの担当者は、次のタスクを表示できます。
 - 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

管理者は、ワークフローチャートの詳細からも特定のアクティビティに関する進行中のタスクのリストにアクセスできます。次の手順に従います。

- 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 ポップアップメニューから [タスク] を選択します。

 **注意:**

表示されるリストは、[ワークフローのタスク] (SQL名: amWfWorkItem) テーブルだけに限定されたビューです。実行しなければならないタスクが表示されません。

ユーザタスクを実行する

ワークフローの担当者は、[ツール / 進行中のタスク] メニューを選択して、実行するタスクのリストにアクセスできます。

 **警告:**

データベースに接続しているユーザがAssetCenter管理者の場合は、[ツール / 進行中のタスク] メニューを選択すると、実行する必要があるすべてのタスクが表示されます。接続ユーザが管理者ではない場合は、[ツール / 進行中のタスク] メニュー項目を選択すると、ユーザに割り当てられたタスクとユーザが属するグループに割り当てられたタスクのみが表示されます。

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。

ユーザタスクの割当て

ユーザタスクの割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

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トランジションを起動します。

アクティビティのレベルには、3種類のシステムタイプイベントがあります。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AssetCenter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、「Executed」(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

アクティビティ用アラームのイベントは、イベントを起動するアクティビティ用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] ポップアップ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、コンテキストレコードまたはリンクしているレコードのテーブル内の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す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、次の方法があります。
 - [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トランジションを有効にすることができます。
 - [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トランジションを有効にすることができます。
 - [削除前] オプション(SQL名: bDelete)では、レコードを削除する前にトランジションを有効にできます。
 - [削除後] オプション(SQL名: bPostDelete)では、レコードを削除した後にトランジションを有効にできます。
 - [更新前] オプション(SQL名: bPreUpdate)を選択すると、トランジションが行われるフィールドをフィールド更新前に指定できます。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] オプション(SQL名: bUpdate)を選択すると、変更時にモニタしなければならないフィールドを [モニタするフィールド(後)] フィールド(SQL名: MonitFields)で指定できます。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、変更されたフィールドがモニタされません。

**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) ゾーン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せん。

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 AssetCenter 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、本マニュアルの「ワークフロー」の章の「イベントの処理」の節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.IUserId] <> IUserId 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、AssetCenter Serverによってイベントがトリガされます。

AssetCenter Server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] タブページで定義します。

イベントの処理方法については、本マニュアルの「ワークフロー」の章、「イベントの処理」の節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、ワークフローチャート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、同じワークフローチャート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、ワークフローチャートの[全般]タブページにある[1つの物件につき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、次の表の通りです。

表 9.1. ワークフローを制限する場合

ワークフローチャートの [全般] タブページの [1つの物件につき1つのワークフローを起動] オプション	
オン	オフ

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	<p>起動条件が満たされたときに、AssetCenter Server 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 ページで定義します。</p> <p>発生するとすぐに、AssetCenter 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。</p> <p>トランジションは、AssetCenter Server によって後で起動されます (AssetCenter Server がトランジションの起動をモニタする頻度は、AssetCenter Server のオプションで定義します)。</p>	<p>起動条件が満たされたときに、AssetCenter Server 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 ページで定義します。</p> <p>発生するとすぐに、AssetCenter 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。</p> <p>トランジションは、AssetCenter Server によって即座に起動されます。</p>	<p>起動条件が満たされたときに、AssetCenter Server 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 ページで定義します。</p> <p>イベントが発生しても、イベントは AssetCenter Server によって、[amWfOccurEvent] というSQL名のテーブルに即座に記録されません。ただし、トランジションは、AssetCenter Server により即座に起動されます。</p>

	[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AssetCenter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、AssetCenter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、AssetCenter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。トランジションは、AssetCenter Serverによって後で起動されます (AssetCenter Serverがトランジションの起動をモニタする頻度は、AssetCenter Server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、AssetCenter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。トランジションは、AssetCenter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、トランジションは、AssetCenterクライアントマシンによって即座に起動されます。
AssetCenter Serverによってトリガされるシステムイベント (AssetCenter Server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、AssetCenter 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。トランジションは、AssetCenter Serverによって後で起動されます (AssetCenter Serverがトランジションの起動をモニタする頻度は、AssetCenter Server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、AssetCenter 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。トランジションは、AssetCenter Server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AssetCenter Serverによって、[amWfOccurEvent] というSQL名のテーブルに即座に記録されません。ただし、トランジションは、AssetCenter Server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	<p>起動条件が満たされたときに、AssetCenter Server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] タブページで定義します。</p> <p>発生するとすぐに、AssetCenter 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。</p> <p>トランジションは、AssetCenter Serverによって即座に起動されます。</p>	<p>起動条件が満たされたときに、AssetCenter Server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] タブページで定義します。</p> <p>イベントが発生しても、イベントはAssetCenter Serverによって、 [amWfOccurEvent] というSQL名のテーブルに即座に記録されません。ただし、トランジションは、AssetCenter Serverにより即座に起動されます。</p>
[データベース] タイプのイベント、またはAssetCenter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、AssetCenter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)	<p>イベントが発生するとすぐに、AssetCenter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。</p> <p>トランジションはAssetCenterクライアントマシンによって即座に起動されます。</p>	<p>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、トランジションは、AssetCenterクライアントマシンによって即座に起動されます。</p>
AssetCenter Serverによってトリガされるシステムイベント (AssetCenter Server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	<p>イベントが発生するとすぐに、AssetCenter 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。</p> <p>トランジションは、AssetCenter Serverによって即座に起動されます。</p>	<p>イベントが発生しても、イベントはAssetCenter Serverによって、 [amWfOccurEvent] というSQL名のテーブルに即座に記録されません。ただし、トランジションは、AssetCenter Serverにより即座に起動されます。</p>

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チャートを定義できます。また、両方のアプローチを組み合わせることもできます。

応用：同期ワークフローチャートを実行する

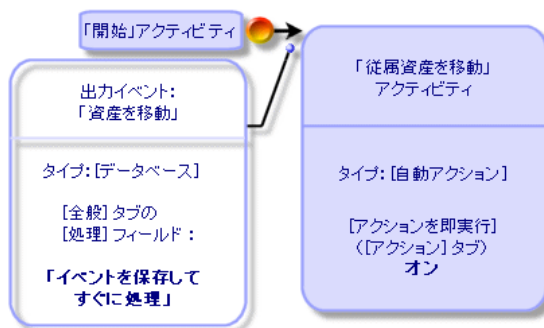
同期ワークフローチャートを作成するには、次の内容を定義する必要があります。

- [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 : bExecImmediatly) 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チャートを使うと、資産の設置場所が変更されると同時に、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. 同期ワークフローチャートの例



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。

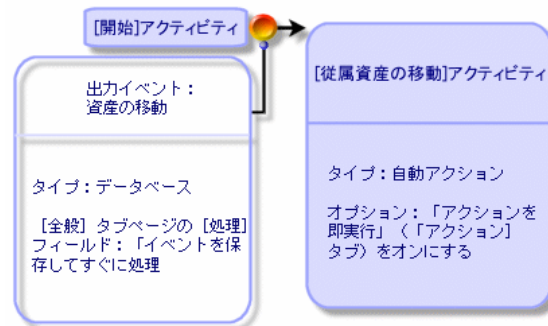
- 4 ワークフローのトランジションが起動します。
- 5 従属資産の設置場所が変更されます。
- 6 トランザクション全体が検証されます。

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チャート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チャートの例



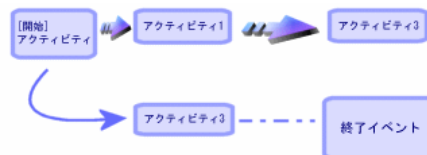
終了イベント

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チャート



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

ワークフローチャートの [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、ワークフローチャートを作成する場合は、次の方法で終了イベントを指定します。

- 1 ワークフローチャート内のイベントをマウスの右ボタンでクリックします。
- 2 ポップアップメニューから [終了イベント] を選択します。

ワークフローのトランジション

トランジションは、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。

イベントは複数のトランジションにリンクできます。

必要に応じて、トランジションの詳細画面の [AQL条件] (SQL名: AQLCond) フィールドで、トランジション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、業務用カレンダー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、AssetCenter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、AssetCenter Serverで変更します。モニタの頻度は、AssetCenter Server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。

- AssetCenter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、ワークフローチャートを分類できます。ワークフローチャート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。

AssetCenter Serverは、新しいワークフロー実行グループの作成をモニタします。

AssetCenter Serverは、新しいワークフロー実行グループを検出し次第、 [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、別のAssetCenter Server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、AssetCenter Serverは実行グループの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、トランジションの起動、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

AssetCenterのあるテーブルを、ワークフローチャート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。

各ワークフローは、それぞれの進行状況を示す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。
- サブタブページの右側には、ワークフローチャートが表示されます。進行中の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。

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

ワークフローチャート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] （SQL名：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] フィールド、SQL名：seStatus）でも自動的に削除されません。

ワークフローチャート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、AssetCenter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 [ワークフローインスタンス] (SQL名: 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、ワークフローチャート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チャートを作成します。
- 4 ワークフローチャートが自動的に実行されるように、AssetCenter Serverを設定します。

[ワークフローチャート] テーブルにフィールドを1つ追加する

[ワークフローチャート] (SQL名: amWfScheme) テーブルに以下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。
説明 ([ヘルプの本文] タブ)	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。

パラメータ	値
例	-1：ワークフローインスタンスは決して削除されません。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、マニュアル『カスタマイズ』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」の節の「フィールド、リンクとインデックスの作成」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チャートの、[インスタンス削除の期限] (SQL名: 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 AssetCenterを起動します。
- 2 本番用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])。
- 2 [新規作成] をクリックします。
- 3 以下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	ワークフローチャート (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 IRc As Long
Dim i As Long

i = 0

If [AutoCleaningDelay] >= 0 Then
```

```

Dim hqWfInstance As Long
hqWfInstance = AmQueryCreate()
IRc = AmQueryExec(hqWfInstance, "SELECT IWfInstanceId FROM amWfInstance
WHERE IWfSchId = "& [IWfSchId] & " And seStatus = 1 AND ADDSECONDS(dt
Completed, " & [AutoCleaningDelay] & ") < GetDate()" )

Do While (IRc = 0 And i < NumberOfInstanceToDelete)
Dim hrWfInstance As Long
hrWfInstance = AmGetRecordHandle(hqWfInstance)
IRc = AmDeleteRecord(hrWfInstance)
IRc = AmReleaseHandle(hrWfInstance)
IRc = 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チャートを作成する

- 1 ワークフローチャートのリストを表示します（[ツール/ワークフロー/ワークフローチャート]メニュー）。
- 2 [新規作成] をクリックします。
- 3 以下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
参照番号	Ref	ADM_CLEAN_WF_INSTC
開始オブジェクトのコンテキスト	StartContextTable	ワークフローチャート（amWfScheme）
実行グループ	GroupName	AssetCenter Server内でワークフローチャート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] タブページを選択します。

- 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 以下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 以下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	ワークフローチャート (amWfScheme)
入力条件	selnCond	OR

- 12 [追加] をクリックします。
- 13 [パラメータ] タブページを選択します。
- 14 以下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	bExeclmmediately	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 以下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。

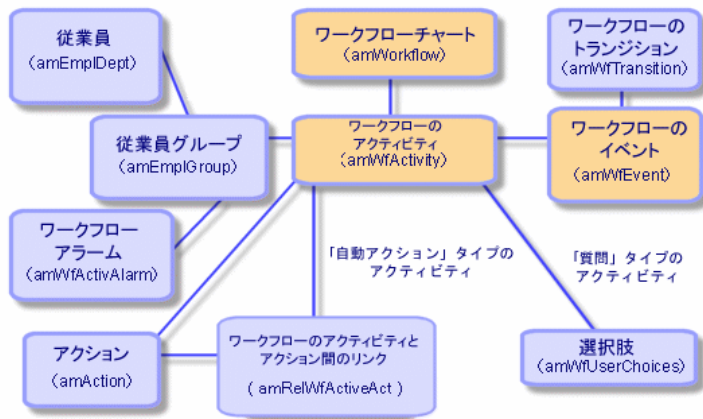
ワークフローチャートが自動的に実行されるように、AssetCenter Serverを設定する

- 1 AssetCenter Server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、ワークフローチャートの[実行グループ]（SQL名：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、AssetCenter Serverを稼動状態にしておきます。

技術情報：データモデル

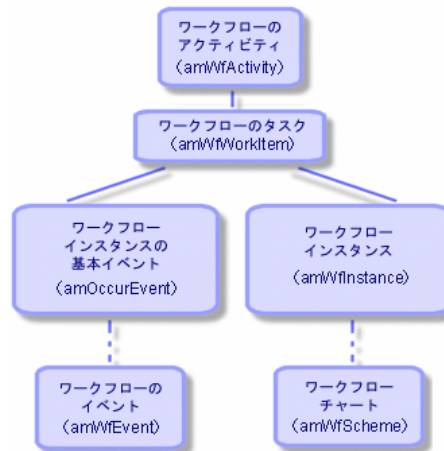
ワークフローに関連するメインテーブルとこれらの関係は、次の図の通りです。テーブルは、ラベルとSQL名で表記されています。

図 9.9. ワークフローチャート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、次の図の通りです。

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

本章では、AssetCenter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

出力スクリプトを使うと、データの出力、AssetCenter Export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。出力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

出力クエリは、AssetCenter Export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


AssetCenterデータベースからデータを出力する

AssetCenter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を使う。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AssetCenterプログラムグループからAssetCenter Export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。
 - AssetCenter Export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:

AssetCenterで定義したアクセス制限の整合性を維持する理由から、AssetCenter Export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AssetCenter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] ポップアップ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/リストの出力] メニューの詳細については、マニュアル『はじめに』の「AssetCenterを初めて使用する」の章、「レコードリスト」の節の「リストを出力する」を参照してください。

AssetCenterデータベースでSQLビューを管理する

AssetCenter Exportを使うと、AssetCenter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、AssetCenter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。

AssetCenter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、次の手順に従います。

- 1 AssetCenter Export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。
 - AssetCenter Exportで直接実行する。
 - amexpl.exeを実行する。

推奨事項

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

出力スクリプトを定義する

データの出力またはデータベースのSQLビューの作成を行うには、出力スクリプトと其中で使う出力クエリを定義する必要があります。これを行うには、AssetCenter Export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、次の手順に従います。

- 1 AssetCenter Export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 AssetCenter Export画面の上部で、データを出力するか（出力モード）、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

出力スクリプトのクエリは、AssetCenter Export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。


- 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。
- コメントを入力することは可能ですが、出力されません。
- [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。
- 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、AssetCenter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(IModelId) 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、このマニュアルの「AQLクエリ」の章、「並べ替えとインデックス」の節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」オプションで、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。

- AssetCenter Exportから出力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

AssetCenter Exportから出力スクリプトを実行する





AssetCenter Exportから出力スクリプトを実行するには、次の手順に従います。

- 1 AssetCenter Export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、AssetCenter Export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、AssetCenter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 : AssetCenterデータベースへの接続名 ([ファイル / データベース接続の管理] 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:biensibm.scx  
-cnx:GeneraleDatabase -login:Gerald -password:PAssword
```


11 | スクリプト

本章では、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。AssetCenter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、AssetCenter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、マニュアル

『管理』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、AssetCenterのウィザードで使われます。

本バージョンのBasicに関する情報

AssetCenter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 Applications™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のマニュアル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、ポップアップメニューが表示されます。

データアクセス表記

AssetCenter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.ILocald]
```

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

AssetCenter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（ポップアップメニューの [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タイプの特特殊フィールド
- フィールドの設定（ポップアップメニューの [オブジェクトの設定] コマンドまたはAssetCenter Database Administrator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
```

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

AssetCenter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用します）

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:

AssetCenter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。


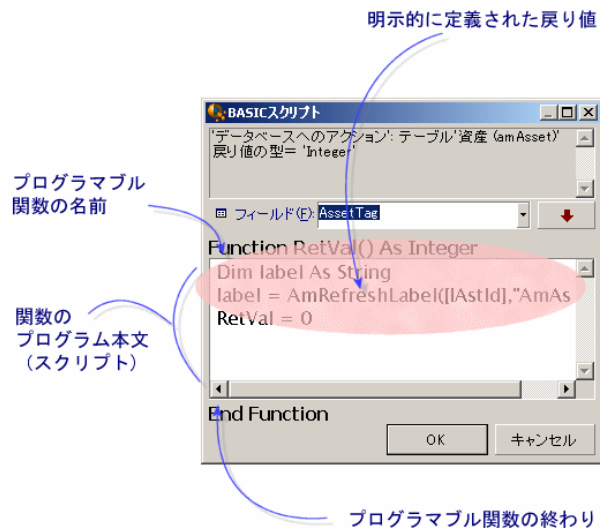
プログラマブル関数には、スクリプトビルダからアクセスできます（スクリプトを作成できるフィールドのボタンをクリックすると表示されます）。スクリプトビルダは、関数に対応するソフトウェアアイテムを、ユーザが簡単に作成できるように設計されています。スクリプトビルダには、プログラマブル関数を作成するためのテンプレートがあります。

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せん。

パラメータのデータ型

組み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、次の表の通りです。

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	任意の文字からなるテキスト
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
- AssetCenter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。AssetCenter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。

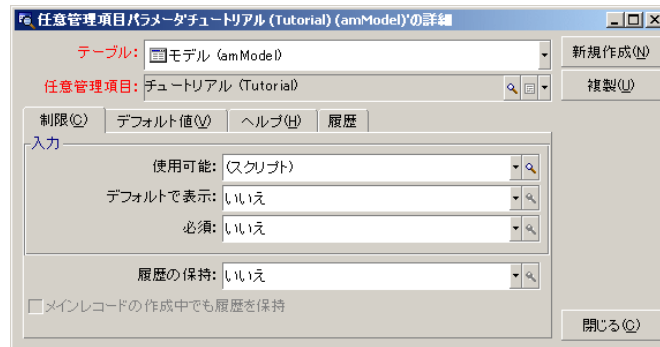
手順1：任意管理項目「チュートリアル」を作成する

[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「(スクリプト)」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] (SQL名：seAvailable) 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] (SQL名：seAvailable) 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] (SQL名：seAvailable) 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が

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、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

AssetCenter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、『プログラマーズリファレンス』の「関数の説明」の節を参照してください。

概念

AssetCenter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、次の手順に従います。

- 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] をクリックして作成を確定します。

例えば、「biblio」という名のライブラリを作成し以下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「amEvalScript」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、AssetCenterの様々な機能で使われます。可能な限りスクリプトを最適化して、AssetCenter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:

AssetCenter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

AssetCenter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] タブページにある [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通りです。

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]	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 リストデータの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=<リストデータ値に対応する整数>
```

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ド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、AssetCenter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、部署と従業員のテーブルについての記述文字列を返します。
- 「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.IPersId]

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：

AssetCenter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。AssetCenter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通りです。

```
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、AssetCenter Database Administrator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。

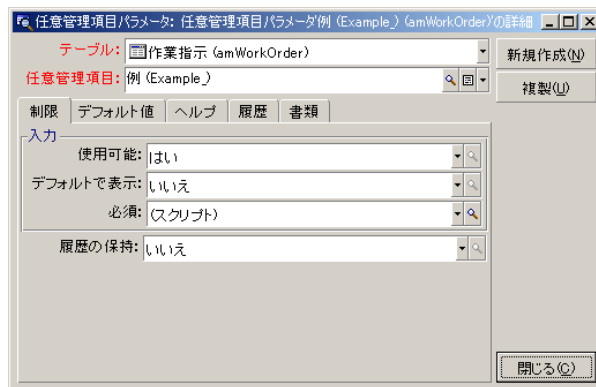
スクリプト例1

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際は、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。

```
If the work order is closed then
Populating the feature is mandatory
Else
Populating the feature is not mandatory
```

作業指示の [ステータス] (SQL名: seStatus) フィールドが [クローズド] 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。

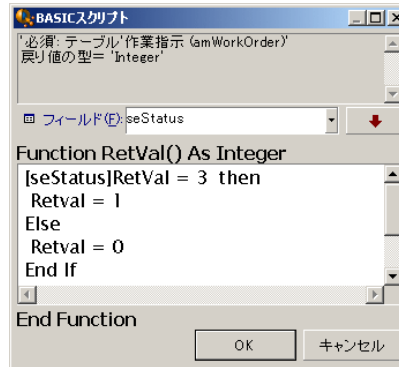
リストデータの値	対応する整数
[報告済]	0
[作業日確定済]	1
[進行中]	2
[クローズド]	3

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 [クローズド]

手順2：Basicスクリプトを書く

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 [クローズド] 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] タブページを選択します。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 [クローズド] 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。

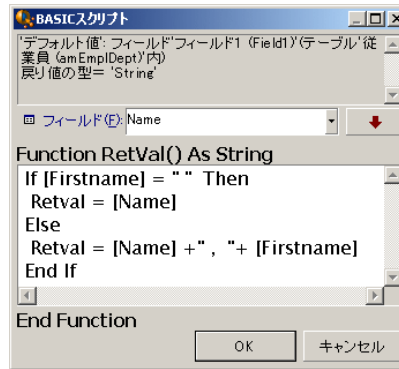
```
If the first name of the employee does not exist then
The default value of "Field1" is the name of the employee
Else
The default value of "Field1" is "Name, First"
```

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スクリプトを書く

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/部署と従業員]メニューを選択して[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となります。

12 | カレンダー

カレンダーのリストにアクセスするには、[ツール/カレンダー]メニューを使います。

カレンダーの概要

カレンダーの詳細画面

カレンダーの詳細画面には、次の情報が表示されます。

- カレンダーを固有に識別できる一般情報
 - カレンダーの [名前] (SQL名: Name)
 - カレンダーがリンクしている [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

カレンダーの使用方法

カレンダー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ーを変更すると、ユーザがAssetCenterを終了し再起動した時に、カレンダー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。

カレンダーに影響される機能

カレンダーは、AssetCenterの特定の機能に影響を与えます。カレンダーを変更すると、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。カレンダー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

カレンダーの作成方法

次の手順でカレンダーを作成します。

- 1 カレンダーに [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、カレンダーをタイムゾーンに関連付けます。
- 3 カレンダーの詳細画面の [予定表] タブページで、通常の業務時間を定義します。
- 4 カレンダーの詳細画面の [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ーの機能を確認できます。

カレンダーの作成手順

カレンダー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] タブページを入力する
- 3 [除外] タブページを入力する
- 4 カレンダーをチェックする

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、カレンダー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、カレンダーに使うタイムゾーンを指定することもできます。

[予定表] タブページを入力する

カレンダーの詳細画面の [予定表] タブページでは、カレンダーの週単位の予定表を定義します。このタブページで定義した業務時間によって週単位の予定表が決まります。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. カレンダーの詳細画面の [予定表] タブページ

予定表(T)	例外(E)	プレビュー(V)
週間予定表の定義		
		0 3 6 9 12 15 18 21 24
月曜日(M):		9:00-12:00;13:00-18:00
火曜日(T):		9:00-12:00;13:00-18:00
水曜日(W):		9:00-12:00;13:00-18:00
木曜日(T):		9:00-12:00;13:00-18:00
金曜日(F):		9:00-12:00;13:00-18:00
土曜日(S):		
日曜日(S):		

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ーコントロールに自動的に入力されません。

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

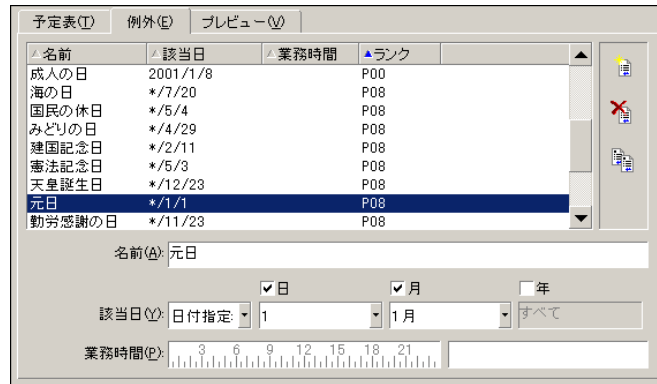
カレンダーの詳細画面の [除外] タブページでは、[予定表] タブページで定義した週単位の業務時間を適用しない期間を定義します。

出力スクリプトの作成方法




例外の期間の作成方法は次の通りです。

- 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できます。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。

[実行日] フィールドの値	例外期間
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
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。

カレンダーをチェックする

[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. カレンダーの詳細画面の [プレビュー] タブページ

The screenshot displays a software interface with three tabs: '予定表(T)', '例外(E)', and 'プレビュー(W)'. The 'プレビュー(W)' tab is active. Under the 'テスト' section, there are three fields: '開始日' (Start Date) set to '2002/03/14 14:00:30', '終了日' (End Date) set to '2002/03/14 14:00:30', and '経過時間' (Elapsed Time) set to '0秒'. Below this is a section titled '開始日に関連するカレンダー' (Calendar related to start date). It shows a list of dates from 2002年3月11日 to 2002年3月17日. For each date, there is a horizontal bar representing the day. The bar for 2002年3月14日 is highlighted in red. To the right of each bar, the working hours are listed as '9:00-12:00;13:00-18:00'. A scale at the top of the calendar grid shows 0 and 24 hours.

- [開始日に関連するカレンダー] 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

本章では、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AssetCenter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、AssetCenter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

AssetCenter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 AssetCenter Database Administrator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ーを定義します。



タイムゾーンを作成する

AssetCenter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

す。AssetCenter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、AssetCenter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。AssetCenter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、AssetCenter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以下の表の通りです。

	<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。
<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通りです。

```
<StdShift>,<DltShift>,<SDay>
,<SMonth>,<SDayPos>,<SHour>
,<DDay>,<DMonth>,<DDayPos>,<DHour>
```

この引数は、下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（この場合は、<DDayPos>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" <DDay>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

副引数	説明	可能な値
<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、<SDayPos>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。
<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" <SDay>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
```

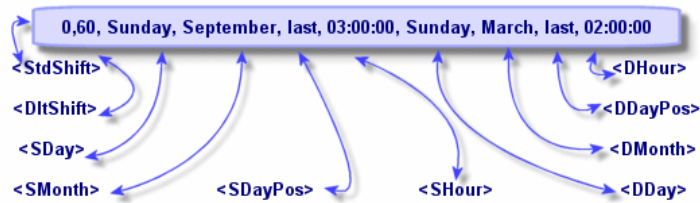
使用されているパラメータは以下の通りです。

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。

AssetCenter Serverでタイムゾーンを管理する

AssetCenter Serverを使って、タイムゾーンのテスト方法を設定できます。[オプション/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AssetCenter Serverをインストールしたマシンの時刻を比較します。時間差は $(n * 30分) + m$ で表されます。ただし、 m は-15から+15分です。

2つのテストに共通する特徴

時間差が5分を超えると、AssetCenter Serverがインストールされているマシンのローカル時刻を更新するメッセージが表示されます。

この更新を拒否すると(例えば、サーバの時刻を更新する場合)、接続が拒否されます。データベースサーバの時刻とAssetCenter Serverをインストールしたマシンの時刻のいずれかを変更して、2つの時刻の差を5分以内になると、再接続できるようになります。

[データベースサーバのタイムゾーンの検証]オプションの特徴

注意:

この機能を正しく実行するには、AssetCenter Serverが稼働しているマシンで、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。

必要に応じて、AssetCenterのオプションテーブルで指定されているサーバのタイムゾーンに関する情報が更新されます($n * 30分$)の数値がサーバのタイムゾーンに対応しない場合)。

[サーバと比較したローカルマシン時間の検証] オプションの特徴

AssetCenterの内部操作に必要なサーバのタイムゾーンを取得します。

テストの頻度

テストは次の場合に実行されます。

- 1 まず、AssetCenter Server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、AssetCenter 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、定期的に行われます。

操作に与える影響

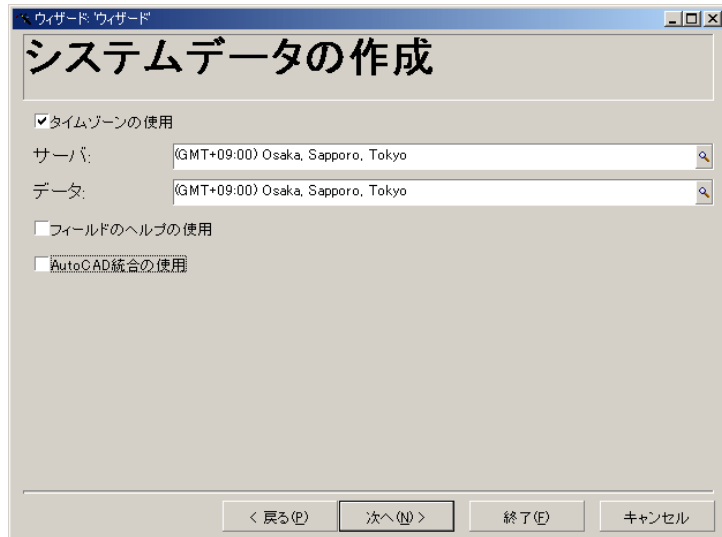
タイムゾーンは、次のような様々な操作に影響を与えます。

- 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。AssetCenter Database Administrator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]セクション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。

[サーバのタイムゾーン]と[データのタイムゾーン]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

AssetCenter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：▶ AssetCenter - インストールガイドの.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、AssetCenterはシステムのタイムゾーン（Windowsで定義）を使います。

次に、AssetCenter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、以下の表の通りです。

「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」（「LocalTimeZone」エントリ）に保存される情報
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、AssetCenter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。

AssetCenter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。

$$\text{クロック差} = \text{剰余}((\text{サーバとローカルマシン間の分単位の時差})/30)$$

 **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）。

$$\text{クロックの差} = \text{母数} (43/30) = 13\text{分}$$

この差が5分（固定値）を超えると、AssetCenter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。

AssetCenterはこのチェックが定期的に行われ、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、次のように「Am.ini」の [option] セクションにある「g_ITimeZoneCheckInMns」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：▶ AssetCenter - インストールガイドの.iniおよび.cfgファイルの章。

```
[option]
g_ITimeZoneCheckInMns = 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、AssetCenter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
- 特殊文字列

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、AssetCenter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、AssetCenter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通りです。

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。

特性	対応するAQL引数
表示可能	SELECT 句

特性	対応するAQL引数
並べ替え可能	SELECT ORDER BY GROUP BY 句
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、このマニュアルの「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。
 - 特殊文字列
 - 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。

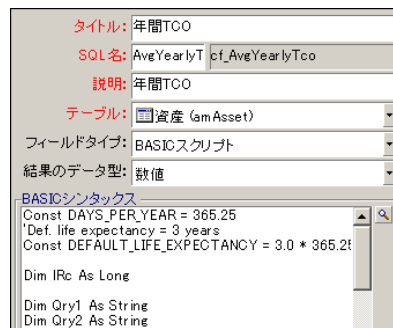
例

- 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。



タイトル: 年間TCO
 SQL名: AveYearlyT cf_AveYearlyTco
 説明: 年間TCO
 テーブル: 資産 (amAsset)
 フィールドタイプ: BASICスクリプト
 結果のデータ型: 数値
 BASICシンタックス
 Const DAYS_PER_YEAR = 365.25
 'Def. life expectancy = 3 years
 Const DEFAULT_LIFE_EXPECTANCY = 3.0 * 365.25!
 Dim IRc As Long
 Dim Qry1 As String
 Dim Qry2 As String

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) 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、次のマニュアルを参照してください。


- このマニュアルの「スクリプト」の章。使用する関数はRetVal()です。
- AQL言語の場合は、このマニュアルの「AQLクエリ」の章。
- 特殊文字列の場合は、マニュアル『管理』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」の節、「テーブルの表記名」。

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

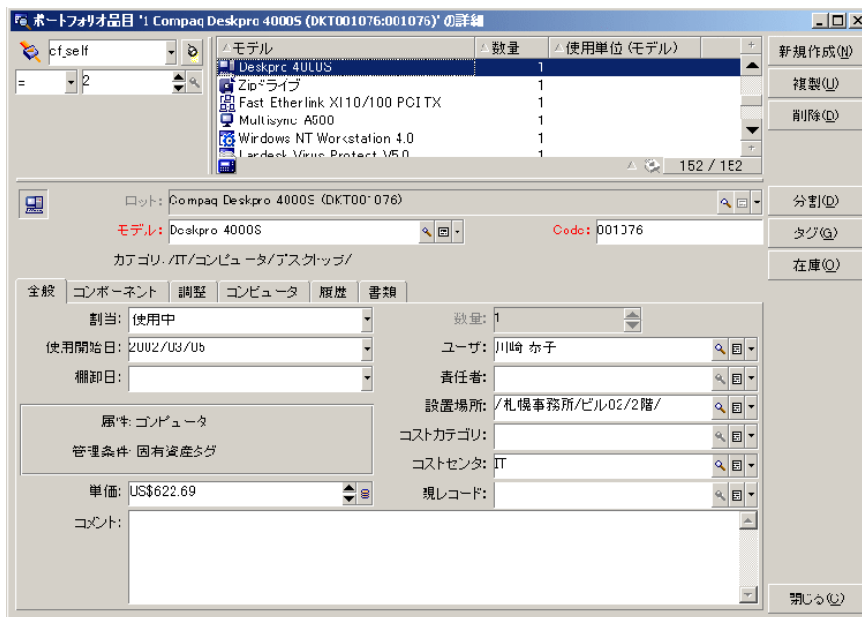
ポップアップメニューの[リストの設定]コマンドを使って、テーブルのすべてのレコード用の特殊フィールドの値を表示できます。

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AssetCenter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、ポップアップメニューから[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名をフィルタにを使って、レコードを検索した例です。



さらに、特殊フィールドは、AssetCenterの次のモジュールまたは機能でも使用できます。

- AssetCenter Web
- AssetCenter API
- レポート
- 書式

これらのモジュールや機能については、対応するマニュアルを参照してください。

15 | ウィザード

AssetCenter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。

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AssetCenter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、AssetCenter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

オブジェクトと子オブジェクト

オブジェクト「A」にオブジェクト「B」が含まれる場合

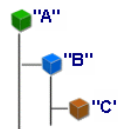
- オブジェクト「A」は、オブジェクト「B」の親オブジェクトになります。
- オブジェクト「B」は、オブジェクト「A」の「子オブジェクト」になります。

 **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。



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。

AssetCenter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、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AssetCenter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、AssetCenterによって自動的に再計算されることはありません。

トランジション

トランジションとは、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トランジションを定義できます。それぞれのトランジションには、トランジション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、最初に有効なトランジション（条件が満たされたトランジション）が実行されます。有効なトランジションがない場合は、[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

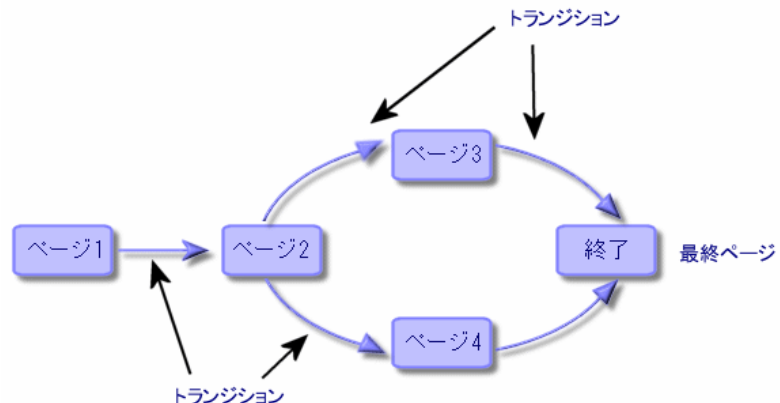
AssetCenterウィザードは、ウィザードを実行するタイプのウィザードです。ウィザードは、構造スクリプトで定義します。

- 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通りです。
 - 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トランジションを記述します。
 - 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、トランジション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通りです。

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、本章の「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
 - $\text{<プロパティ名>} = \text{<テキスト>}$
- ブール型のプロパティ
 - $\text{<プロパティ名>} = \text{TRUE}$
 - $\text{<プロパティ名>} = \text{FALSE}$
 - $\text{<プロパティ名>} (\text{<プロパティ名>} = \text{TRUEに相当})$
- 数値型のプロパティ
 - $\text{<プロパティ名>} = 42$
- $\text{<プロパティ名>} = \{ \text{<Basic変数またはプロパティのフルネーム>} \}$

 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="Le nom est  ": & {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}="Chose 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)
...

 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、AssetCenterにより自動的に入力されません。ユーザが値を手動で入力することはできません。
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、AssetCenterにより自動的に入力されません。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。

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

AssetCenter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()
- 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, "IUserId", {IEmplDeptId})
```

正しいスクリプトは次の通りです。

```
Dim IValue as Long  
IValue=AmGetFieldLongValue(hRecord, "IUserId", {IEmplDeptId})
```

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

ROOT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="Move Wizard"	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、この名前で作成されます。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>	Basicスクリプト（またはコード）が含まれます。このプロパティは、グローバル変数（すべてのウィザードに適用可能）およびウィザードのすべてのノードで使用できるグローバル関数などを定義する時に役立ちます。 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。 文字列型のプロパティ	{GLOBAL="Dm Filter As String}	

プロパティ名=値	プロパティの説明	例	コメント
COMMON	<p>全ウィザード内に自動的に入るBasic関数が含まれます。</p> <p>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。</p>		
SERIALIZE=<TRUE FALSE>	<p>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。</p> <p>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。</p> <p>ブール型のプロパティ.iniファイルの場所：▶ AssetCenter - インストールガイドの.iniおよび.cfgファイルの章。</p>	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>	<p>ウィザードがモーダルか（=TRUE）否か（=FALSE）を定義します。</p>		

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、AssetCenterの「Config」フォルダを基準にします。 AssetCenter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	

プロパティ名=値	プロパティの説明	例	コメント
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。	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。		
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。		

プロパティ名=値	プロパティの説明 例	コメント
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。	
CONFIRMCANCEL=<TRUE FALSE>	キャンセルの確認メッセージを表示するか (=TRUE) しないかを (=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トランジション名 (終了)。このトランジション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ヘ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。 文字列型のプロパティ	{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>	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。	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。
IMAGE16=<ビットマップファイルのパス>	文字列型のプロパティ		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 <トランジション名>	ウィザードの現在のページと別のページ間のトランジション名を定義します。
TIMER	ウィザードのページヘ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:

トランジションは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トランジションは、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。

```
' トランジションの宣言
{ TRANSITION0 <トランジション名>
' 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=<トランジションのリンク元のページの名前>	トランジション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、トランジションが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>	トランジション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>	トランジション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>	トランジション時に実行されるスクリプトを定義します。 文字列型のプロパティ	{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 ノードでトランジションを定義する理由

PAGE ノードの外でトランジションを定義すると、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トランジションを定義します。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。 文字列型のプロパティ		
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。 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	
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。 ブール型のプロパティ	

INTERVAL=間隔	<p>タイマを実行する時間の間隔を定義します。</p> <p>間隔はミリ秒単位で指定します。</p>
TIMER="Script"	<p>タイマの間隔 ("INTERVAL") ごとに実行される内容を定義します。</p> <p>スクリプトタイプのプロパティ</p>
VALUE=tickcount	<p>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。</p> <p>tickcountの値は自動的に増加します。</p>

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>	<p>ノード名を名前に持つ変数の値を定義します。</p> <p>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</p>	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、AssetCenter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>	<p>トランジションを確認するために、ユーザにコントロールへの入力を強制します。</p> <p>以下の場合このプロパティは無視されます。</p> <ul style="list-style-type: none"> ■ 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。
VALUE=<値>	<p>作成時にコントロールのデフォルト値を定義します。<値>はそのコントロールによって決まります。</p> <p>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）。</p>	例えば、CHECKBOXコントロールの場合、<値>はTRUEまたはFALSE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
PERMANENT=<TRUEFALSE>	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ず。 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=<TRUEFALSE>	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。[Shift+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	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)、ユーザー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		

プロパティ名=値	プロパティの説明	例	コメント
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		
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=A1B=2C=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。「タイトル」にはオプションボタンのテキストを定義し、「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"
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	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。
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"	

プロパティ名=値	プロパティの説明	例	コメント
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%', ' ', ',', '=')
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"	「VALUES」プロパティを「VALUE」プロパティ混同しないでください。
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"	

プロパティ名=値	プロパティの説明	例	コメント
COLNAME=" <code><名前 名前></code> "	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 Name Name Name	
MULTISEL = <code><TRUE FALSE></code>	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)	

プロパティ名=値	プロパティの説明	例	コメント
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)	
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

プロパティ名=値	プロパティの説明	例
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、AssetCenter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、AssetCenter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
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、AssetCenter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"	
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"	

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>	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	このプロパティはデフォルトでは「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、AssetCenter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 = "User.IEmpIDeptId = '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。		
VALUES.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。	VALUES.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 {dblestbox1.IsSelection() (ユーザがリストか ら何も選択しない と、トランジション 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。 <ul style="list-style-type: none"> ■ 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 = {dblestbox1.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、AssetCenter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"	


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>	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	このプロパティはデフォルトでは「FALSE」に設定されます。
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) かない (=FALSE) かを指定します。 ブール型のプロパティ	MULTISEL=TRUE	

プロパティ名=値	プロパティの説明	例	コメント
DBLCLICK=<TRUE FALSE>	このプロパティをTRUEに設定する場合、AssetCenter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	
[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。		

DBEDITコントロール

DBEDITコントロールでは、AssetCenter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"
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。 <ul style="list-style-type: none"> ■ 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 [俊蔵84]
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。 <ul style="list-style-type: none"> ■ 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、AssetCenterデータベースのテーブル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

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、AssetCenter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、AssetCenter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>	 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])	<p>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。</p> <p>タイトルタイプ (iNameType) は以下の通りです。</p> <ul style="list-style-type: none"> ■ 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) <p>TABLEプロパティには値が入力されなければなりません。</p>	▶DBQUERYBOXコントロールの物理プロパティ [修版84]
LINK.LABEL([iNameType])	<p>このメソッドはリンクのタイトルを返します。</p> <p>タイトルタイプ (iNameType) は以下の通りです。</p> <ul style="list-style-type: none"> ■ 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) <p>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。</p>	

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"

プロパティ名= 値	プロパティの説明	例
FORMAT	データ型を定義します。 <ul style="list-style-type: none"> ■ 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。	
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。 <ul style="list-style-type: none"> ■ MODE=0 : 縦の棒グラフ ■ MODE=1 : 横の棒グラフ ■ 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。	

プロパティ名= 値	プロパティの説明	例
POPOP	ポップアップ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。	
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。

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。 <ul style="list-style-type: none"> ■ 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。	
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ーを挿入できます。

TIMESPANEDITコントロール

このコントロールでは、期間の入力ゾーンを挿入できます。

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。 <ul style="list-style-type: none"> ■ 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

このコントロールでは日付タイプのコントロールを挿入できます。

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。 <ul style="list-style-type: none"> ■ 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法

AssetCenter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 ）。

このゾーンをクリックし、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:

ポップアップメニューを使って、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IPAGE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、テキスト編集ゾーンに入力します。


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:

ポップアップメニュー（プロパティをマウスの右ボタンでクリックして表示）から [デフォルト値に戻す] を選択すると、固定値またはスクリプトが取り消され、デフォルト値に戻ります。この操作は、値またはスクリプトがユーザによって定義されたプロパティ（黒で表示される）でのみ使用可能です。

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 手順1：ニーズを分析する
- 2 手順2：ウィザードの編成方法を定義する
- 3 手順3：スクリプト言語を使ってウィザードの構造を記述する

手順1：ニーズを分析する

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に、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2：ウィザードの編成方法を定義する

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。

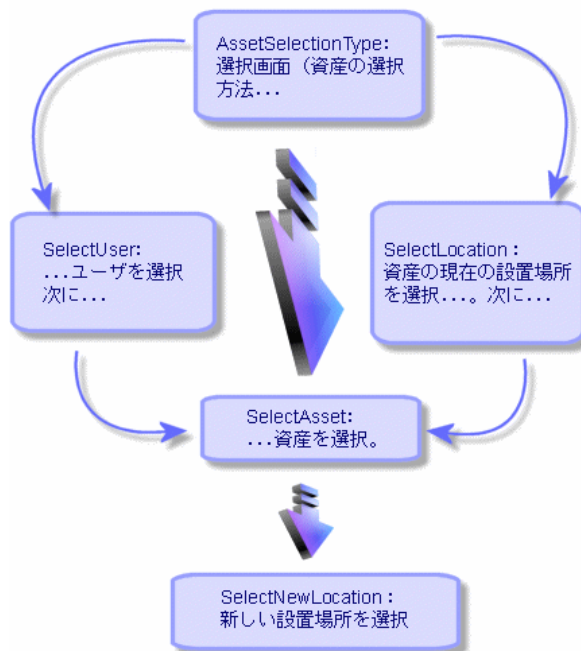
- 1 ページ数
- 2 ページ同士のリンク方法
- 3 各ページの内容

 **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、ページごとのトランジションを定義できます。

ページ	トランジションのリンク先ページ
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3 : スクリプト言語を使ってウィザードの構造を記述する

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。

```

=====
==
;(c) Peregrine Systems 1999
=====
==
NAME = "移動"

```

```

TITLE = "ユーザの移動"
VERSION = "699"
;=====
;移動するユーザを指定するページ。デフォルトでは、このテーブルがコンテキ
;ストの場合、 amEmpIDept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 = "移動する従業員"
MULTISEL = 1
TABLE = "amEmpIDept"
{ VALUE =
if [CurrentTable] = "amEmpIDept" then
RetVal = [CurrentSelection]
else
RetVal = ""
end if
}
VISIBLE = 1
}
{ TRANSITION trPersonToNewLoc
TO = "pgNewLoc"
}
}
;=====
;新しい設置場所を指定するページ
;=====
{ PAGE pgNewLoc
TITLE = "新しい設置場所の選択"
{ STRING UserName
VALUE = AmDbGetString("SELECT FirstName + ' ' + Name FROM amEmpIDept WH
ERE IEmpIDeptID 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"
}
}

;=====
;選択内容の確認ページ
;=====

{ PAGE pgRecap
TITLE = "Recapitulatif"
{ LISTBOX Users
COLTITLE = "名前"
COLWIDTH = "100"
LABEL = "移動する従業員"
MANDATORY = 0
MULTISEL = 1
READONLY = 1
VALUE = ""
VALUES = AmDbGetList("SELECT FullName FROM amEmpIDept WHERE FullNam
e LIKE LikeParam(amEmpIDept_2:FullName)+'%' AND amEmpIDept_2:IDeptID I
N(" & {pgUser.Users} & ")","|","|","|","=")
}
}

;=====
;最後のページ
;=====

{ FINISH FINISH
{ DO =
On Error Goto ErrHandler
Dim IErr as long

```

```
dim hRecord as Long

dim iEmplCount as Integer
iEmplCount = {pgRecap.Users.VALUES.Count()}
dim iMax as Long
iMax = iEmplCount

dim lLocald as long
lLocald = {pgNewLoc.NewLocId}

lErr = amStartTransaction()

dim i as Integer
For i = 1 To iEmplCount
lErr = AmProgress((100 * i) / iMax)
lErr = AmLog("Deplacement de la personne " + {pgRecap.Users.VALUES(i,1)})
hRecord = AmGetRecordFromMainId("amEmplDept", {pgRecap.Users.VALUES(i,0)})

If hRecord <> 0 then
lErr = AmSetFieldLongValue(hRecord, "lLocald", lLocald)
lErr = AmUpdateRecord(hRecord)
lErr = AmReleaseHandle(hRecord)
End If
Next i

lErr = amCommit()

RetVal = 0
Exit Function

ErrorHandler:
On Error Goto 0
AmLog(AmLastError() & " - " & AmLastErrorMsg())
AmLog("La transaction a ete annulee")
RetVal = 1
Exit function
}
SUMMARY = 1
}
```

クエリウィザード（QBE）の作成の例

クエリウィザードは、所定の画面またはビューにあらかじめフィルタをかける特殊なウィザードです。

次の例では、標準のクエリウィザードの1つについて説明します。

 **注意:**

このサンプルのウィザードは、画面上で機能します。ロジックはビューの場合と同じです。

手順1：ニーズを分析する

このウィザードの目的は、作業指示画面にあらかじめフィルタをかけることです。

このため、以下を行う必要があります。

- 1 クエリウィザードで使用するフィールドを決める。
- 2 フィールドのタイプを決める。

手順2：ウィザードの編成方法を定義する

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
フィルタ手順にあまり負荷をかけないように、1ページだけを作成することをお勧めします。
- 2 各ページの内容
- 3 ページ同士のリンク方法

ページ	このページの目的	使用するコントロール
Filter	Présente les différents champs permettant un préfiltrage de l'écran ou de la vue à afficher	TEXTBOX COMBOBOX LABEL LINKEDIT DATETIMEEDIT CHECKBOX

ページ	このページの目的	使用するコントロール
Advanced	オプションのフィールドを提示して、さらに詳細なフィルタを可能にします。	COMBOBOX LINKEDIT DATETIMEEDIT LABEL

手順3：スクリプト言語を使ってウィザードの構造を記述する

ここでは、作業指示画面で用いられるウィザードスクリプトの一部について説明します。

これらの部分は、新しいIQBEウィザードの基礎として使用できます。

重要項目:

ウィザードでオブジェクトの名前を定義するときには、ウィザードのシンタックスの予約語（「name」、「title」など）は使用できません。

- ```
NAME = "searchamWorkOrder"
TITLE = "Filter the work orders"
VERSION = "3152"
{ PAGE Filter
TITLE = "Quick search"
```

ウィザードのタイトルページの定義。
- ```
{ TEXTBOX WONo
LABEL = AmGetFieldLabelFromName("amWorkOrder","WONo")
LABELLEFT = 1
VALUE = ""
XOFFSET = 1500
}
```

[作業指示番号] オブジェクトを作業指示テーブルに作成し、ウィザードにそのタイトルを表示します。
- ```
{ COMBOBOX seStatus
LABEL = AmGetFieldLabelFromName("amWorkOrder","seStatus")
LABELLEFT = 1
VALUE = ""
{ VALUES =
Dim strFormat As String
strFormat = AmGetFieldFormat(AmGetFieldFromName(AmGetTableFromName("amWorkOrder"), "seStatus"))
RetVal = SysEnumToComboBox(strFormat)
```

```

}
XOFFSET = 1500
}
{ COMBOBOX Priority
LABEL = AmGetFieldLabelFromName("amWorkOrder","Priority")
LABELLEFT = 1
VALUE = ""
{ VALUES =
Dim strValues As String
strValues = AmDbGetList("SELECT Value FROM amItemListVal WHERE Itemize
dList.Identifier = 'amWOPriority'", "", "", "", "")
RetVal = EnumToComboBox(strValues)
}
XOFFSET = 1500
}

```

#### リストデータの定義

##### 1 [ステータス] オブジェクトの場合 :

- 1 作業指示テーブルの [ステータス] フィールドのタイトルを使用してウィザードに表示します。
- 2 フィールドをテキスト型のフィールドとして定義
- 3 リストデータタイプのオブジェクトを作成
- 4 このフィールドに関連するシステムリストデータの内容を使用し、ウィザード設定ライブラリに定義されているSysEnumToComboBox関数を使ってフォーマット。

##### 2 [優先度] フィールドの場合 :

- 1 フィールドをテキスト型のフィールドとして定義
- 2 [優先度] フィールドに関連するリストデータの内容を使用
- 3 リストデータタイプのオブジェクトを作成し、ウィザード設定ライブラリに定義されているEnumToComboBox関数を使ってそのオブジェクトの内容をフォーマット。

```

■ { LINKEDIT Model
FILTER = "Nature.seBasis = 3"
LABEL = AmGetFieldLabelFromName("amWorkOrder","Model")
LABELLEFT = 1
TABLE = "amModel"
VALUE = 0
XOFFSET = 1500
YOFFSET = 0
YOFFSET2 = 0
}

```

リンクのコントロールオブジェクトを作成 :

- 1 属性によって作業指示が作成されるすべてのモデルに対してフィルタを作成
- 2 作業指示テーブルの [ モデル ] フィールドのタイトルを使用してウィザードに表示
- 3 リンクが選択されるテーブルを定義

```

■ { LABEL Empty2
}
{ COMBOBOX OPdtResolLimit
LABEL = AmGetFieldLabelFromName("amWorkOrder", "dtResolLimit")
LABELLEFT = 1
VALUE = ""
VALUES = "Before=<≠,After=>≠,Le=≠,After=<>"
XOFFSET = 1500
}
{ DATETIMEEDIT dtResolLimit
FORMAT = "DateTime"
LABEL = ""
LABELLEFT = 1
VALUE = ""
XOFFSET = 1500
}

```

リストデータタイプのオブジェクトの作成 :

- 1 作業指示テーブルの [ 解決予定期限 ] フィールドのタイトルを使用してウィザードに表示
- 2 可能なオペレータのリストを定義  
日付タイプのオブジェクトの作成

```

■ { CHECKBOX bAdvanced
CAPTION = "Advanced search"
VALUE = 0
}
{ TRANSITION Transition
CONDITION = 1
{ TO =
If {Filter.bAdvanced} = 0 Then
RetVal = "Finish"
Else
RetVal = "Advanced"
End If
}
}
}

```

チェックボックスの作成 :



- 1 チェックボックスのラベルの定義
- 2 チェックボックスの動作の定義：デフォルトでクリア
- 3 トランジションの定義：チェックボックスが選択されると、ウィザードの [次へ] ボタンが有効になる

```

■ { FINISH FINISH
 { DO =
 Dim strFilter As String

 Dim strOperator As String
 strOperator = ""

 'WONo condition
 If {Filter.WONo} <> "" Then
 strFilter = strFilter & " AND WONo LIKE " & AmSqlTextConst({Filter.WONo})
 End If

 'Model condition
 If {Filter.Model} <> "0" Then
 strFilter = strFilter & " AND IModelId = " & {Filter.Model}
 End If

 If strFilter<> "" then
 strFilter = RightPartFromLeft(strFilter, "AND", 0)
 RetVal = AmOpenScreen("amWorkOrder","",strFilter,1,"")
 Else
 'ERR CANCEL
 RetVal = 2
 End if
 }
}

```

フィルタ条件の作成（FINISH）：

- 1 FINISHオブジェクトが作成されます。
  - このオブジェクトは、SQLステートメントのWHERE節を作成するために用いられます。
- 2 各オブジェクトに対してフィルタ条件が定義されます。
  - WONo条件のテキストフィールドタイプのオブジェクト：
    - 1 フィールドが空でなければ、挿入された値に基づいてフィルタ。
  - リンクタイプのオブジェクトのモデル条件：
    - 1 フィールドが空でなければ、リンクのIDに基づいてフィルタし、モデルのリンクを表示。
- 3 表示する画面の値に対するフィルタ条件が定義されます。

この条件は定義済のWHERE句を使用し、最初のANDの後から始まる部分だけを考慮します（クエリを単純にするため）。

- 4 フィルタされる画面は、AmOpenScreen関数から呼び出されます。

## 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(IRow)}
```

### 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(IRow)}
```

## 質問

次のコードが機能しません。

```
{IbxMyListBox.Values.Line({IbxTmp})}
```

## 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 IRow As Long
IRow = {IbxTmp}
{IbxMyListBox.Values.Line(IRow)}
```

## 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 AssetCenter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き込みできる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を確認されます。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", "AssetCenter", "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 = "amEmpIDept"
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

本章では、AssetCenter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：「2002/02/10、11：00と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せん。

## 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。従業員グループのテーブルは階層構造になっているため、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。

## ニュースを表示する

ニュースを表示するには、次の操作を行います。

- 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。

次の内容を定義できます。

- 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

AssetCenter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。                                   |

# インデックス

PEREGRINE

- \$C (書式), 69
- \$D (書式), 69
- \$N (書式), 69
- \*= (AQL), 47
- < (AQL), 47
- <> (AQL), 47
- != (AQL), 47
- = (AQL), 47
- =\* (AQL), 47
- =< (AQL), 47
- = ALL (AQL), 47
- = ANY (AQL), 47
- = SOME (AQL), 48
- > (AQL), 47
- >= (AQL), 47
- アクション, 95-111
  - (参考ウィザード)
  - タイプ, 97
  - テスト, 110
  - ボタン - 関連付け, 111
  - メッセージ, 114
  - 印刷 - コンテキスト, 100
  - 機能ドメイン, 96
  - 作成, 97
  - 実行, 110
  - 定義, 95
  - 変数, 109
  - 例, 106
    - DDE, 106
    - スクリプト, 107
    - メッセージ, 107
    - [実行可能ファイル], 106
    - [DDE] タブ, 102
    - [メッセージ] タブ, 103
  - アクションのタイプ, 97
  - アクションのテスト, 110
  - アクション (アクションのタイプ), 101, 100
  - アクセス権の定義
    - Basic関数, 190
    - FINISH (ウィザードのノード), 263
    - LONG (ウィザードのノード), 266
    - ROOT (ウィザードのノード), 252
    - START (ウィザードのノード), 264
    - STRING (ウィザードのノード), 266
    - TIMER (ウィザードのノード), 265
    - TRANSITION (ウィザードのノード), 261
  - twip, 240
  - アクション, 95
  - オブジェクト, 241
  - オブジェクトのフルネーム, 241

- コントロール, 240
- コントロール (ウィザードのノード), 267
- トランジション, 242
- ニュース, 317
- ノード, 240
- ページ (ウィザードのノード), 258
- ワークフローインスタンス, 118
- ワークフローのアクティビティ, 118
- ワークフローのアクティビティの担当者, 119
- ワークフローのイベント, 119
- ワークフローのタスク, 119
- ワークフローのトランジション, 119
- ワークフローの実行グループ, 119
- 規定レポート, 86
- 子オブジェクト, 241
- 終了ワークフローイベント, 163
- 出力クエリ, 176
- 出力スクリプト, 175
- 書式, 65
- 親オブジェクト, 241
- 特殊フィールド, 231
- 変数, 242
- アドレス - メッセージタイプのアクション, 104
- インデックス  
(参考クエリ)
- ウィザード, 239-315
  - { - シンタックス, 245
  - Basic
    - 連結, 251
  - Basic関数, 251
  - FINISH (ノード), 263
  - LONG (ノード), 266
  - ROOT (ノード)
    - サブノード, 258
    - シンタックス, 252
    - プロパティ, 252
    - 定義, 252
  - START (ノード), 264
  - STRING (ノード), 266
  - TIMER (ノード), 265
  - TRANSITION (ノード)
    - シンタックス, 261
    - プロパティ, 261
  - 定義, 261
  - 特異性, 262
- エディタ
  - アクセス, 296
  - インタフェース, 296
  - ツールバー, 297
  - テキスト - 検索, 297
  - 使用, 296
- クエリウィザード, 306
- コントロール, 267
  - CALENDAR, 294
  - CHART, 291
  - CHECKBOX, 271
  - COMBOBOX, 272
  - COMBOEDIT, 295
  - COMMANDBUTTON, 280
  - DATETIMEEDIT, 295
  - DBEDIT, 286
  - DBLISTBOX, 281
  - DBPATH, 288
  - DBQUERYBOX, 283
  - DBTABLE, 288
  - FILEEDIT, 293
  - LABEL, 279
  - LINKEDIT, 288
  - LISTBOX, 274
  - NUMBOX, 295
  - OPTIONBUTTONS, 273
  - PROGRESSBAR, 279
  - TEXTBOX, 290
  - TICKEDIT, 294
  - TIMESPANEDIT, 294
- コントロール (ノード), 267, 267
- スクリプト - 制限, 251
- セル - 値, 281
- デバッグ, 299
- ノード
  - CurrentSelection, 249
  - CurrentTable, 249
  - スクリプト, 247
  - スクリプトまたは固定値 - 優先順位, 299
  - テーブル, 248
  - プロパティ, 298, 247, 245
  - プロパティ - 制限, 246
  - メソッド, 248

- 構造とシンタックス, 244
  - 作成, 298
  - 数値, 246
  - 宣言的モデル, 246
  - 定数, 246
  - 名前 - 制限, 241
- プロパティ, 267
- ページ - モデル, 244
- ページ (ノード)
  - サブノード, 260
  - シンタックス, 258
  - プロパティ, 259
  - 定義, 258
- 開始 - ページ, 265
- 概要, 244
- 構造テンプレート, 242
- 作成パラメータ, 264
- 作成 - 例, 300
- 質問, 311
- 実行, 299
- 実行 - 制限, 299
- 表記 - 規則, 239
- 連鎖, 250
- ウィザードのトランジション, 242
- ウィザード (アクションのタイプ), 100
- エラー - スクリプト, 203
- エラーメッセージ - スクリプト, 203
- オブジェクト, 241
- オブジェクトのフルネーム, 241
- カレンダー, 209-216
  - コントロール, 215
  - プレビュー, 209
  - 一般情報, 211
  - 概要, 210
  - 作成, 211, 210
  - 精度, 212
  - 他の機能への影響, 210
  - 変更 - 考慮, 210
  - [ 予定表 ] タブ, 211
- クエリ, 23-64
  - AQLとSQLの比較, 25
  - AQLの有用性, 24
  - CurrentUser, 35
  - DELETE, 54
  - DUPLICATE, 53
  - FROM, 48
  - FullName (フィールド), 37
  - GROUP BY, 50
  - HAVING, 51
  - INSERT, 52
  - NULL, 34
  - ORDER BY, 52
  - Self, 35
  - sLvl (フィールド), 37
  - UPDATE, 53
  - WHERE, 49
  - インデックス, 40
  - インデックス - 注意事項, 42
  - エディタ, 26
    - アクセス, 27
    - 概要, 26
  - システムリストデータ, 36, 36
  - シンタックス, 42
    - UNION, 44
    - サブクエリ, 43
    - シンプルクエリ, 43
    - 表記法, 42
  - テーブル数, 49
  - はじめに, 23
  - フィールド, 29
  - フィルタ, 28
    - 直接入力, 29
  - プレビュー, 29
  - リストデータ, 31
  - 階層構造のテーブル, 37
  - 外部キー, 38
  - 外部結合
    - 演算子, 34
    - 数値, 32
  - 概要, 25
  - 管理者, 28
  - 簡単な表記法, 38
  - 関数 (参考 AQLの関数)
  - 共有, 28
  - 結合, 31
  - 作成, 28
  - 式, 29
  - 主キーが0のレコード, 32
  - 推奨事項, 31
  - 数値フィールド, 35

- 定数, 30
- 特殊フィールド, 39
- 任意管理項目, 39
- 必要な技術, 24
- 表記文字列, 39
- 並べ替え, 40
  - インデックスの使用, 40
  - リストデータ, 41
  - 順序, 41
  - 注意事項, 42
- 要素, 44
  - フィールドとリンク, 44
  - 演算子, 46
  - 式, 46
  - 選択リスト, 48
  - 定数, 45
- 例, 60, 60
  - 2つの条件, 62
  - alias, 64
  - FullName, 61
  - FullNameおよびsLvl, 61
  - メインテーブルにリンクしたテーブルのフィールド, 61
  - リンク, 61, 61
  - リンクなし, 63
  - 階層構造のテーブル, 61
  - 式, 62
  - 数値、日付、テキスト, 62
  - 任意管理項目, 62
  - 未入力フィールド, 62
  - [ウィザード], 306
- 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
- グリッド, 73
- コメント - スクリプト, 202
- コントロール, 240
- コントロールパネル, 89-94
  - インジケータ, 90
  - ナビゲーション, 94
  - レポート機能, 89
- 概要, 89
- 作成, 89
- 統計とレポート, 90
- 例, 91
- コントロール (ウィザードのノード), 267, 267
- 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), 229, 226
- サブクエリ, 43
- システムリストデータ - スクリプト, 200
  - ウィザード, 307
- システムリストデータ - 値, 37
- スクリプト, 187-207
  - Basic関数
    - コールに回答する, 189
    - タイプ - 制約, 193
    - パラメータの型, 192
    - プログラマブル関数, 191
    - ヘルプ, 188
    - 関数の型, 192
    - 組み込み関数, 190
    - 分類, 193
    - 名前 - 制限, 198
  - CurrentUser, 201
  - エラーメッセージ, 203
  - コメント, 202
  - システムリストデータ, 200
  - ノード (ウィザード), 247
  - ヒント, 198
  - ライブラリ, 197
  - リンク - ID, 188
  - 期間, 199
  - 時刻, 199
  - 手順1, 193
  - 大文字と小文字の区別, 196
  - 注意事項, 198
  - 日付, 199
  - 表記 - 規則, 188
  - 編集, 187
  - 用途, 189
  - 例, 206, 203
- スクリプトライブラリ, 197
- スクリプト (アクションのタイプ), 99
- タイムゾーン
  - AssetCenter Server - 制限, 225
  - インポート, 229
  - クライアントマシンの時刻, 228
  - 夏時間, 224, 219
  - 概要, 225

- 管理, 219
- 作成, 218
- 使用可能性, 217
- 出力, 229
- 接続, 227
- 操作に与える影響, 226
- 有用性, 217
- タイムゾーンのインポート, 229
- タイムゾーンの使用 (オプション), 227, 218
- タイムゾーン - 実装, 218
- タスクを保存 (オプション), 164
- テーブル
  - (参考 履歴)
  - ワークフロー, 172
- テキスト (書式のプロパティ), 71
- テスト/スクリプト (ワークフローのアクティビティ), 148
- 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), 299
- データベース
  - (参考 履歴)
  - 既製の書式, 65
- データベースサーバのタイムゾーンの検証 (オプション), 225
- データ - 出力 (参考 出力)
- ニュース
  - メッセージ, 318
  - 概要, 318
  - 作成, 318
  - 重要度, 318
  - 定義, 317
  - 読取り, 318
  - 配信先リスト, 319
  - 表示, 319
- ニュースの重要度, 318
- ノード (ウィザード)
  - 構造とシンタックス, 244
  - 定義, 240
- フィールド
  - (参考 履歴)
  - クエリ, 61, 44
  - 未入力フィールド, 62
- 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, 237
- フォント (書式), 72
- フッター - 書式, 74
- プログラマブル関数, 198
- ページ設定 - 書式, 73
- ページ内の配置 (書式), 72
- ページ (ウィザードのノード), 258
- メソッド - ウィザード, 248
- メッセージ, 113-115
  - メッセージの参照, 114
  - メッセージの発送, 114
  - 概要, 113
  - 内部メッセージシステム - アドレス, 99
  - 配信済通知, 115
- メッセージ (アクションのタイプ), 98
- メッセージ (タブ), 103
- メッセージ (メニュー), 115
- モジュール
  - レポート, 87
  - 書式, 75
- ユーザのアクション (ワークフローのアクティビティ), 146
- ユーザ権限 - 特殊フィールド, 236
- リスト
  - 書式, 70
  - 特殊フィールド, 237
- リストデータ, 13-320, 13-16
  - ウィザードのスクリプト, 307
  - システムリストデータ, 15, 15
  - 不特定のリストデータ, 13
    - オープンリストデータ, 14
    - クローズドリストデータ, 15
  - 値, 14
  - 変更 - 考慮, 15
- リストの出力 (メニュー), 177
- リストの設定 (メニュー), 40, 27
- リストの内容 (書式), 73
- リスト内のリンク (書式), 72
- リンク
  - (参考 履歴)
  - クエリ, 61, 61, 44
  - リンクなし, 63
- レポート
  - (参考 書式)



- CurrentUserId, 85
- ID, 87
- SQL名 (制限), 84
- インストール
  - Crystal Reportsのバージョン (フルまたはランタイム), 79
  - 既製のレポート, 79
  - 既存のデータベース, 80
  - 新規データベース, 79
- プレビュー, 82
- ボタン - 関連付け, 87
- リスト - 制限, 73
- 概要, 77
- 既製のレポート, 79
- 規定レポート, 86
- 詳細, 81
- 詳細レポート
  - 作成, 84
  - 使用 - 例, 84
  - 設定, 84
- 統計, 83
- 変更, 83
  - [印刷], 86
- レポートのプレビュー, 82
- レポートの印刷, 86
- ワークフロー, 117-172
  - SQL名 (制限), 149
  - アクション - トリガ, 146
  - アクセス権の定義, 118
  - アクティビティ, 145
    - テスト/スクリプト, 148
    - テンプレート, 149
    - トリガ, 150
    - ユーザのアクション, 146
  - 開始, 149
  - 自動アクション, 147
  - 質問, 145
  - 担当者 - アクティビティのタイプ, 143
    - [アラーム] タブ - 制限, 165
    - [制限時間] タブ - 制限, 165
- アクティビティテンプレート - 制限, 150
- アラーム, 164
- イベント, 153
  - アラーム, 154
  - システム, 153
  - データベース, 154
  - ユーザ, 154
  - 起動, 157
  - 終了イベント, 163
  - 処理, 158
  - 定期的, 156
    - [開始] アクティビティ, 154
- インスタンス
  - 削除, 167, 167
  - 自動削除, 168
- グラフィックエディタ, 121
- コンテキスト, 141
- タスク, 150
  - アラーム - 制限, 151
  - スクリプト, 151
  - テスト, 151
  - ユーザタスク, 152
  - 割当, 153
  - 管理者, 153
  - 作成, 151
  - 自動アクション, 151
  - 進行中のタスク, 151
- テーブル, 172
- トラッキング, 167
- トランジション, 164
- トリガ - 制限, 156
- 概要, 119
- 期間, 165
- 期限, 164
- 実行グループ, 166
- 制限時間, 166, 165
- 同期ワークフロー, 162
- 導入, 121
- 複雑な入力条件, 150
- 役割, 143
- 例, 124
  - アクティビティ, 131
  - イベントの設定, 136
  - トランジション, 138
  - トリガ, 138
  - 開始イベント, 137
  - 必要条件, 125
- ワークフロー - アクション - スクリプト, 202
- ワークフローインスタンス
  - 削除, 167

- 制限, 142
- 定義, 118
- ワークフローエディタ, 121
- ワークフローチャート(参考 ワークフロー)
- ワークフローチャート(メニュー), 120
- ワークフローのアクティビティ, 145
  - グラフィックエディタ, 122
  - 定義, 118
- ワークフローのアクティビティのテンプレート, 149
- ワークフローのアクティビティの担当者
  - 割当, 145
  - 定義, 119
- ワークフローのアラーム, 165, 164
- ワークフローのイベント, 153
  - エディタ, 122
  - 定義, 119
- ワークフローのコンテキスト, 141, 141
- ワークフローのタスク, 150, 119
- ワークフローのトランジション, 164
  - エディタ, 123
  - 定義, 119
- ワークフローの期限, 165, 164
- ワークフローの実行グループ, 166, 119
- ワークフローの制限時間, 165
- ワークフローの役割, 143
- 印刷(アクションのタイプ), 100
- 演算子 - クエリ, 46
- 横 - 書式, 74
- 夏時間, 222
- 夏時間(フィールド), 219
- 画像(書式), 72
- 開始(ワークフローのアクティビティ), 149
- 階層構造のテーブル - クエリ, 61
- 外部キー - クエリ, 38
- 外部結合
  - 演算子, 34
  - 数値, 32
- 期間 - スクリプト, 199
- 機能ドメイン - アクション, 96
- 計算式 - 書式, 70
- 計算式(書式), 71
- 計算(ボタン), 110
- 結合, 31
- 固定テキスト - 書式, 69
- 行揃え(書式), 72
- 参照オブジェクト - ワークフロー, 141
- 参照オブジェクト(フィールド), 104
- 子オブジェクト, 241
- 時刻 - スクリプト, 199
- 自動アクション(ワークフローのアクティビティ), 147
- 式 - クエリ, 46
- 質問(ワークフローのアクティビティ), 145
- 実行可能(アクションのタイプ), 98
- 実行(ボタン), 110
- 終了ワークフローイベント, 163, 163
- 縦 - 書式, 74
- 出力, 175-185
  - SQLビュー, 177
    - 作成, 183
    - 削除, 183
  - アクセス権限, 177
  - クエリ, 181
  - スクリプトの出力形式, 182
  - タイムゾーン, 229
  - ビュー(タブ) - 表示, 183
  - フィルタ, 181
  - プレビュー, 181
  - プロセス, 176
    - スクリプト, 176
    - ポップアップメニュー, 177
  - 出力クエリ, 179
  - 出力スクリプト
    - 実行, 184
    - 実行 - DOS, 184
    - 実行 - プロセス, 184
    - 書込み, 178
  - 出力スクリプトの作成方法, 179
  - 出力するデータ, 180
  - 出力する列, 180
  - 書式設定(タブ) - 表示, 182
  - 推奨事項, 178
  - 並べ替え, 180
- 出力クエリ
  - 設計, 179
  - 定義, 176
- 出力スクリプト
  - 実行, 184
  - 書込み, 178

- 定義, 175
- 書式
  - (参考 レポート)
  - アイコン, 68
  - オブジェクト
    - テキスト, 71
    - フォント, 72
    - プロパティ, 71
    - ページ内の配置, 72
    - リストの内容, 73
    - リスト内のリンク, 72
    - 位置とサイズ, 71
    - 画像, 72
    - 計算式, 71
    - 行揃え, 72
    - 色, 72
    - 複数選択, 71
    - 枠, 72
  - オブジェクトのプロパティ, 71
  - グリッド, 73
  - ページ設定, 73
    - フッター, 74
    - ヘッダー, 74
    - マージン, 74
  - ボタン - 関連付け, 76
  - モジュール, 75
  - リスト, 70
  - 引用符, 69
  - 画像, 70
  - 既製の書式, 65
  - 固定テキスト, 70, 69
  - 作成, 67
  - 書式のフィルタ, 68
  - 定期レポート, 75
  - 定義, 65
  - 変数, 69
  - 編集, 68
- 書式のアイコン, 68
- 書式のヘッダー, 74
- 書式の画像, 70
- 親オブジェクト, 241
- 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), 157
- 数値 - クエリ, 62
- 宣言的モデル - ウィザード, 246
- 選択リスト - クエリ, 48
- 値のリスト (参考 リストデータ)
- 定数
  - クエリ, 45
  - ノード (ウィザード), 246
- 導入 (アクションのタイプ), 100
- 特殊フィールド, 231-238
  - (参考 スクリプト)
  - SQL名 - 制約, 235
  - クエリ, 39
  - タイプ, 232
    - 使用する言語, 236
    - 推奨事項, 235
    - 用途, 233
  - データベース - 値の格納, 231
  - フィルタ, 237
  - ユーザ権限の詳細画面, 236
  - リスト, 237
  - 計算式, 236
  - 作成, 232
  - 参照, 238
  - 使用, 237
  - 使用のコンテキスト, 236
  - 定義, 231
  - 有効性, 232
- 日付
  - クエリ, 62
  - スクリプト, 199
- 任意管理項目
  - (参考 履歴)
  - クエリ, 62, 39
- 背景色 (書式), 72
- 配信済通知, 115, 104
- 配信先リスト - ニュース, 319
- 表記文字列 - クエリ, 39
- 文字の色 (書式), 72
- 並べ替え
  - (参考 クエリ)
- 変数
  - アクション, 109
  - 書式, 69
  - 定義, 242
  - 予定表, 211
  - 履歴, 17-22
    - 1リンク, 19

- nリンク, 20, 19, 19
- データベース - インポート, 18
- フィールド, 19
- レコード, 18
- 概要, 17
- 削除, 21
- 任意管理項目, 20
  - 起動, 21
  - 削除, 21
  - 追加, 20
  - 変更, 21
  - 履歴項目の作成、削除、または変更, 21
  - 履歴項目を作成する, 17
- 履歴項目 (参考 履歴)
- 例外 (タブ), 212
- 枠 (書式), 72
  
- A**
- Abs (AQL), 59
- AddDays (AQL), 56
- AddHours (AQL), 56
- AddMinutes (AQL), 57
- AddSeconds (AQL), 57
- Advanced Query Language (参考 クエリ)
- alias (AQL), 64
- am.ini, 227
- amexpl.exe, 177
- AND (AQL), 46
- AQL (参考 クエリ)
- AQLの関数, 54
  - テスト, 60
    - IsNull, 60
- 集合型, 54
  - Avg, 54
  - Count, 54
  - Countdistinct, 54
  - Max, 55
  - Min, 55
  - Sum, 55
- 数値, 59
  - Abs, 59
  - Ceil, 59
  - DataLength, 59
  - Floor, 59
  - Length, 59
  - Mod, 59
  - NullNumeric, 59
  - NumberToNumber, 59
  - Round, 59
  - Sign, 59
  - TextToNumber, 59
  - Trunc, 59
- 日付, 56
  - AddDays, 56
  - AddHours, 56
  - AddMinutes, 57
  - AddSeconds, 57
  - Day, 57
  - DayOfYear, 57
  - DaysDiff, 57
  - DbToLocalDate, 57
  - Getdate, 57
  - Hour, 57
  - HoursDiff, 57
  - LocalToDbDate, 57
  - Minute, 57
  - MinutesDiff, 57
  - Month, 57
  - NumberToTime, 57
  - Second, 58
  - SecondsDiff, 58
  - TextToTime, 58
  - WeekDay, 58
  - Year, 58
- 文字列, 55
  - Ascii, 55
  - Char, 55
  - CharIndex, 55
  - DateToText, 55
  - EmptyString, 55
  - Left, 55
  - Length, 55
  - LikeParam, 55
  - Lower, 55
  - Ltrim, 55
  - NullBlob, 56
  - NullDate, 57
  - NullMemo, 56
  - NullString, 56

NullTime, 57  
 NullTimeStamp, 57  
 NumberToText, 56  
 Right, 56  
 Rtrim, 56  
 Substring, 56  
 TimeStampToText, 56  
 TimeToText, 56  
 Upper, 56  
 Ascii ( AQL ), 55  
 AssetCenter Server - タイムゾーン, 225  
 Avg ( AQL ), 54

## B

Basic (参考 スクリプト)  
 (参考 ウィザード)  
 情報, 188  
 Basic関数  
 (参考 ウィザード)  
 (参考 スクリプト)  
 定義, 190

## C

CALENDAR (ウィザードのコントロール),  
 294  
 Ceil ( AQL ), 59  
 Char ( AQL ), 55  
 CharIndex, 55  
 CHART (ウィザードのコントロール), 291  
 CHECKBOX (ウィザードのコントロール),  
 271  
 COMBOBOX (ウィザードのコントロール),  
 272  
 COMBOEDIT (ウィザードのコントロール),  
 295  
 COMMANDBUTTON (ウィザードのコント  
 ロール), 280  
 Count ( AQL ), 54  
 Countdistinct ( AQL ), 54  
 Crystal Reports (参考 レポート)  
 Crystal Reports - レポートの設定, 84  
 Crystal Reports統計, 83  
 CurrentSelection (ウィザードのノード), 249  
 CurrentTable (ウィザードのノード), 249

CurrentUser ( AQL ), 35  
 CurrentUserId ( レポート ), 85  
 CurrentUser ( スクリプト ), 201

## D

database.txt, 200, 31, 16  
 DataLength ( AQL ), 59  
 DATETIMEEDIT (ウィザードのコントロー  
 ル), 295  
 DateToText, 55  
 Day ( AQL ), 57  
 DayOfYear ( AQL ), 57  
 DaysDiff ( AQL ), 57  
 DBEDIT (ウィザードのコントロール), 286  
 DBLISTBOX (ウィザードのコントロール),  
 281  
 DBPATH (ウィザードのコントロール), 288  
 DBQUERYBOX (ウィザードのコントロー  
 ル), 283  
 DBTABLE (ウィザードのコントロール), 288  
 DbToLocalDate ( AQL ), 57  
 DDE (アクションのタイプ), 98  
 DDE (タブ), 102  
 DELETE ( AQL ), 54  
 DUPLICATE ( AQL ), 53

## E

EmptyString, 55

## F

FILEEDIT (ウィザードのコントロール), 293  
 FINISH (ウィザードのノード), 263  
 Floor ( AQL ), 59  
 FROM ( AQL ), 48  
 FullName ( AQL ), 61  
 FullName (フィールド), 37  
 クエリ, 61

## G

g\_ITimeZoneCheckInMns (am.ini), 228  
 gbbase.wiz, 254  
 gbbase.xml, 80, 66  
 Getdate ( AQL ), 57  
 GROUP BY ( AQL ), 50

**H**

HAVING (AQL), 51  
 Hour (AQL), 57  
 HoursDiff (AQL), 57

**I**

INSERT (AQL), 52  
 IS NOT NULL (AQL), 47  
 IsNull (AQL), 60  
 IS NULL (AQL), 47

**L**

LABEL (ウィザードのコントロール), 279  
 Left (AQL), 55  
 Length, 55  
 Length (AQL), 59  
 LIKE (AQL), 47  
 LikeParam, 55  
 LINKEDIT (ウィザードのコントロール), 288  
 LISTBOX (ウィザードのコントロール), 274  
 LocalToDbDate (AQL), 57  
 LONG (ウィザードのノード), 266  
 Lower (AQL), 55  
 Ltrim (AQL), 55

**M**

MAPI (参考メッセージ)  
 Max (AQL), 55  
 Microsoft SQL Server  
   クエリ  
     並べ替え順, 42  
 Min (AQL), 55  
 Minute (AQL), 57  
 MinutesDiff (AQL), 57  
 Mod (AQL), 59  
 Month (AQL), 57

**N**

NOT LIKE (AQL), 47  
 NULL (AQL), 34  
 NullBlob, 56  
 NullDate, 57  
 NullMemo, 56  
 NullNumeric (AQL), 59

NullString, 56  
 NullTime, 57  
 NullTimeStamp, 57  
 NumberToNumber (AQL), 59  
 NumberToText, 56  
 NumberToTime (AQL), 57  
 NUMBOX (ウィザードのコントロール), 295

**O**

Old, 202  
 OPTIONBUTTONS (ウィザードのコントロール), 273  
 OR (AQL), 46  
 Oracle for WorkGroups  
   クエリ  
     並べ替え順, 41  
 ORDER BY (AQL), 52

**P**

PROGRESSBAR (ウィザードのコントロール), 279

**Q**

QBE  
   クエリウィザードの作成の例, 306

**R**

reports.txt, 78, 78  
 Right (AQL), 56  
 ROOT (ウィザードのノード), 252  
 Round (AQL), 59  
 Rtrim (AQL), 56

**S**

Second (AQL), 58  
 SecondsDiff (AQL), 58  
 SELECT (AQL)  
   サブクエリ - 警告, 43  
   制限, 48  
 Self (AQL), 35  
 Sign (AQL), 59  
 sLvl - クエリ, 37  
 sLvl (フィールド) - クエリ, 61  
 SMTP (参考メッセージ)

## SQL

クエリ, 54

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), 264

STRING (ウィザードのノード), 266

Substring (AQL), 56

Sum (AQL), 55

Sybase SQL Server

クエリ

並べ替え順, 42

## T

TEXTBOX (ウィザードのコントロール), 290

TextToNumber (AQL), 59

TextToTime (AQL), 58

TICKEDIT (ウィザードのコントロール), 294

TIMER (ウィザードのノード), 265

TIMESPANEDIT (ウィザードのコントロール), 294

TimeStampToText, 56

TimeToText, 56

TRANSITION (ウィザードのノード), 261

Trunc (AQL), 59

twip, 240

tz.scr, 219

## U

UNION (AQL), 44

UPDATE (AQL), 53

Upper (AQL), 56

useSQL92Join (amdb.ini), 32

## V

VIM (参考 メッセージ)

## W

WeekDay (AQL), 58

WHERE (AQL), 49

## Y

Year (AQL), 58





