

Peregrine

AssetCenter

---

# AssetCenterの高度な使い方



© Copyright 2002 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。

この製品はApache Software Foundation (<http://www.apache.org>) に開発されたソフトウェアを含んでいます。

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4.1.0に適用されます。

AssetCenter

Peregrine Systems, Inc.  
Worldwide Corporate Campus and Executive Briefing Center  
3611 Valley Centre Drive San Diego, CA 92130  
Tel 800.638.5231 or 858.481.5000  
Fax 858.481.1751  
[www.peregrine.com](http://www.peregrine.com)



# 目次

1. リストデータ . . . . .	19
不特定のリストデータ . . . . .	19
リストデータの値 . . . . .	20
オープンリストデータ . . . . .	21
クローズドリストデータ . . . . .	21
システムリストデータ . . . . .	21
システムリストデータの値 . . . . .	21
2. 履歴 . . . . .	23
レコードを作成する . . . . .	24
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） . . . . .	25
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） . . . . .	25
リンク（別のテーブルへのnリンク）を削除する . . . . .	25
リンク（別のテーブルへのnリンク）を変更する . . . . .	26
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 . . . . .	26
任意管理項目の追加 . . . . .	26
任意管理項目の削除 . . . . .	26
任意管理項目の値の変更 . . . . .	27
履歴項目の作成、削除、または変更 . . . . .	27

履歴項目を作成する . . . . .	27
<b>3. AQLクエリ . . . . .</b>	<b>29</b>
はじめに . . . . .	29
AssetCenterのクエリ . . . . .	29
AQL . . . . .	30
AQLクエリの記述に関する推奨事項 . . . . .	32
AQL結合について . . . . .	33
主キーが0のレコードの理由と有用性 . . . . .	34
NULLの使い方 . . . . .	36
例：Location=0 または ILocalId=0 . . . . .	36
CurrentUser . . . . .	37
システムリストデータ . . . . .	38
階層構造のテーブル . . . . .	39
AQLの簡単な表記法 . . . . .	40
並べ替えとインデックス . . . . .	42
例 . . . . .	42
インデックスを使う方法 . . . . .	42
並べ替え順 . . . . .	43
注意事項 . . . . .	44
クエリエディタ . . . . .	44
概要 . . . . .	44
クエリエディタにアクセスする . . . . .	45
クエリエディタを使ってクエリを作成する . . . . .	46
クエリで使うフィールド . . . . .	48
式を記述する . . . . .	48
定数 . . . . .	49
AQLのシンタックス . . . . .	50
表記法 . . . . .	50
クエリのシンタックス . . . . .	51
FROM句：クエリの対象となるテーブル . . . . .	52
クエリの要素 . . . . .	53
WHERE句 . . . . .	58
GROUP BY句 . . . . .	58
HAVING句 . . . . .	59
ORDER BY句 . . . . .	60
AQLの関数 . . . . .	60
集合型のAQL関数 . . . . .	61
文字列型のAQL関数 . . . . .	61
日付型のAQL関数 . . . . .	62
数値型のAQL関数 . . . . .	64
テスト型のAQL関数 . . . . .	64
クエリの例 . . . . .	65

メインテーブルのフィールドを特定の値と比較する . . . . .	65
メインテーブルのリンクを別のリンクと比較する . . . . .	65
メインテーブルのリンクを特定の値と比較する . . . . .	65
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 . . . . .	66
階層構造のテーブル . . . . .	66
2つの条件を組み合わせたクエリ . . . . .	66
数値、日付、テキスト型フィールドの比較 . . . . .	66
任意管理項目に関するクエリ . . . . .	67
式でレコードを検索する . . . . .	67
未入力のフィールドを検索する . . . . .	67
リンクなしのレコードを検索する . . . . .	67
エイリアスを使ったクエリ . . . . .	68
<b>4. 書式 . . . . .</b>	<b>71</b>
書式の定義 . . . . .	71
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 . . . . .	71
データベース作成時の書式のインポート . . . . .	72
既存のデータベース内に書式をインポートする . . . . .	72
書式を作成する . . . . .	72
基本的な情報 . . . . .	73
書式とオブジェクトを編集する . . . . .	73
固定テキスト . . . . .	74
計算式 . . . . .	74
リスト . . . . .	75
画像 . . . . .	75
書式内のオブジェクトのプロパティ . . . . .	75
位置とサイズ . . . . .	75
プロパティ . . . . .	76
グリッドを使う . . . . .	78
書式のページ設定 . . . . .	78
定期レポートを簡単に作成する方法 . . . . .	79
モジュールに関連する書式の識別 . . . . .	80
<b>5. Crystal Reports . . . . .</b>	<b>81</b>
レポート作成プログラムの操作とインストール . . . . .	81
概要 . . . . .	81
Crystal Reports完全版またはランタイムのインストール . . . . .	83
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 . . . . .	83
レポートの詳細設定 . . . . .	84
ファイル . . . . .	85
Seagate Crystal Reportsのレポートを変更する . . . . .	85

Crystal Reports統計	86
属性	86
【自動更新】ボタン	86
【ズーム】ボタン	86
詳細レポートを作成する	87
利用例	87
Crystal Reportsでレポートを設定する	87
モジュールに関連するCrystalレポートの識別	89
<b>6. アクション</b>	<b>91</b>
アクションの定義	91
機能ドメイン	92
アクションを作成する	92
アクションのタイプ	93
一般的な作成方法	96
[DDE] タブページに入力する	97
[メッセージ] タブページに入力する	99
アクションの例	101
【実行可能ファイル】タイプのアクション例	101
[DDE]タイプのアクション例	102
[メッセージ]タイプのアクション例	106
[スクリプト]タイプのアクション例	107
変数を使う	108
Sybase SQL Anywhereデータベースエンジンを使う場合	109
アクションをテストする	109
【計算】ボタン	109
【実行】ボタン	109
アクションを実行する	110
リストで複数のレコードを選択	110
「ウィザード」タイプのアクション	110
<b>7. メッセージ</b>	<b>113</b>
メッセージシステムの仕組み	113
メッセージを送信する方法	114
メッセージを参照する方法	114
配信済通知	115
<b>8. ワークフロー</b>	<b>117</b>
定義	117
ワークフロー	117
ワークフローのアクティビティ	118
ワークフローのイベント	119

ワークフローのトランジション	119
ワークフローのタスク	119
ワークフローのアクティビティの担当者	120
ワークフローの実行グループの定義	120
ワークフローの概要	120
ワークフローに関連するメインテーブル	121
ワークフローのグラフィカルエディタの使用法	123
アクティビティ	123
イベント	124
トランジション	125
その他の機能	125
ワークフローの活用法	125
ワークフローを使った依頼承認手続き例	126
目的	126
ワークフローチャートを使用する前に	127
アクティビティを作成する	134
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	139
開始イベントを作成する	140
トランジションを作成する	141
ワークフローの起動例	141
ワークフローのコンテキスト	144
ワークフローのコンテキストを定義する	144
ワークフローが参照するオブジェクト	145
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	145
ワークフローの役割	146
ワークフローの役割のタイプ	147
アクティビティの担当者を定義する	148
ワークフローのアクティビティ	148
「質問」タイプのアクティビティ	149
「ユーザのアクション」タイプのアクティビティ	150
「自動アクション」タイプのアクティビティ	151
「テスト/スクリプト」タイプのアクティビティ	152
[開始]アクティビティ	153
アクティビティテンプレート	153
アクティビティをトリガする	153
タスク	154
タスクを作成する	154
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	155
進行中のタスクのリストを表示する	155
ユーザタスクを実行する	156
ユーザにタスクを割り当てる	156
ワークフローのタスクを管理する	157
イベント	157
システムイベント	157

アラームイベント . . . . .	157
ユーザイベント . . . . .	158
起動の一般的な条件 . . . . .	160
イベントの処理 . . . . .	161
応用：同期ワークフローチャートを実行する . . . . .	165
終了イベント . . . . .	166
ワークフローのトランジション . . . . .	167
ワークフローのアラームと制限時間 . . . . .	167
制限時間 . . . . .	168
ワークフローのアラーム . . . . .	168
ワークフローの実行グループ . . . . .	169
ワークフローのトラッキング . . . . .	170
<b>9. データの出力およびSQLビューの作成 . . . . .</b>	<b>171</b>
定義 . . . . .	171
出力スクリプト . . . . .	171
出カクエリ . . . . .	172
AssetCenterデータベースからデータを出力する . . . . .	172
出力スクリプトを使ってデータを出力する . . . . .	172
[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 . . . . .	173
AssetCenterデータベースでSQLビューを管理する . . . . .	173
推奨事項 . . . . .	174
出力スクリプトを定義する . . . . .	174
出力スクリプトの作成方法 . . . . .	175
出カクエリを定義する . . . . .	175
出力スクリプトの出力書式 . . . . .	178
SQLビューに関するアクション . . . . .	179
出力スクリプトを実行する . . . . .	180
AssetCenter Exportから出力スクリプトを実行する . . . . .	180
DOSから出力スクリプトを実行する . . . . .	181
<b>10. スクリプト . . . . .</b>	<b>183</b>
スクリプトの定義 . . . . .	183
概要 . . . . .	183
本バージョンのBasicに関する情報 . . . . .	184
データアクセス表記 . . . . .	184
スクリプトの適用 . . . . .	185
関数の概要 . . . . .	186
関数の定義 . . . . .	186
組み込み関数とプログラマブル関数 . . . . .	186
関数とパラメータのデータ型 . . . . .	188
Basic関数を分類する . . . . .	190
スクリプト作成の基本 . . . . .	190



スクリプト例の内容	190
手順1：任意管理項目「チュートリアル」を作成する	191
手順2：編集ウィンドウを開く	191
手順3：アルゴリズムを分析および定義する	192
手順4：Basicスクリプトを作成する	192
手順5：Basicスクリプトをテストする	193
スクリプトライブラリ	193
概念	193
スクリプトライブラリを作成する	194
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	194
ヒントと注意事項	195
プログラマブル関数を使うときの注意事項	195
スクリプトで使用する「日付+時刻」定数の形式	195
スクリプトでの「Duration（時間）」の形式	196
システムリストデータの読取りと書込み	196
CurrentUser仮想リンク	197
Basicスクリプトにコメントを付ける	198
エラーメッセージをトリガする	198
スクリプト例1	199
スクリプトの目的	199
手順1：アルゴリズムを分析および定義する	199
手順2：Basicスクリプトを書く	201
手順3：Basicスクリプトをテストする	201
スクリプト例2	201
スクリプトの目的	202
手順1：アルゴリズムを分析および定義する	202
手順2：Basicスクリプトを書く	203
手順3：Basicスクリプトをテストする	203
<b>11. カレンダー</b>	<b>205</b>
カレンダーの概要	205
カレンダーの詳細画面	205
カレンダーの使用方法	206
カレンダーに影響される機能	206
カレンダーの作成方法	206
カレンダーの作成手順	207
一般情報を入力する	207
[ 予定表 ] タブページに入力する	207
[ 例外 ] タブページに入力する	208
カレンダーをチェックする	211
<b>12. タイムゾーン</b>	<b>213</b>
タイムゾーンを管理する理由	213

例	214
タイムゾーン機能を実装する	214
タイムゾーンを作成する	214
タイムゾーンを管理する	215
夏時間フィールドのフォーマット	215
<Year>引数の値	216
<DaylightInfo>引数の値	216
例	218
AssetCenter Serverでタイムゾーンを管理する	221
実行するテスト	221
テストの頻度	222
操作に与える影響	222
データベースの作成時	222
データベースへの接続時	224
インポートおよび出力時	225
<b>13. 特殊フィールド</b>	<b>227</b>
特殊フィールドの定義	227
特殊フィールドの用途	228
特殊フィールドを作成する	228
特殊フィールドの概要	228
特殊フィールドの作成方法	230
特殊フィールドの使用方法	233
リストの設定で特殊フィールドを使う	233
テーブルのレコードをフィルタを使って検索する	234
特殊フィールドを参照する	234
<b>14. ウィザード</b>	<b>237</b>
表記法	237
定義	238
Twip	238
コントロール	238
ノード	238
オブジェクト	239
オブジェクトと子オブジェクト	239
オブジェクトのフルネーム	239
変数	240
トランジション	240
構造モデル	240
ウィザードページのモデル	243
ウィザードに関する一般情報	243
一般的な構造とシンタックス	243
ノードのプロパティ	244

宣言的モデル	245
定数をプロパティの値として定義する	245
プロパティを参照する	245
スクリプトをプロパティの値として定義する	246
プロパティで適用できるメソッド	247
テーブルタイプのプロパティ	247
グローバル変数CurrentTableとCurrentSelectionを使う	248
ウィザードの連鎖化	249
実行	249
作成パラメータ	249
Basic関数	250
ROOTノードの定義	250
ROOTノードのシンタックス	251
ROOTノードのプロパティ	251
ROOTノードのサブノード	257
PAGEノードの定義	257
PAGEノードのシンタックス	257
PAGEノードのプロパティ	258
PAGEノードのサブノード	259
TRANSITIONノードの定義	259
TRANSITIONノードのシンタックス	260
TRANSITIONノードのプロパティ	260
TRANSITIONノードの特徴	261
ROOTノードでトランジションを定義する理由	261
FINISHノードの定義	261
STARTノードの定義	263
TIMERノードの定義	264
倍長整数および文字列ノードの定義	265
コントロールノードの定義	265
コントロールノードの一般的なシンタックス	266
コントロールのタイプと関連プロパティ	266
共通のプロパティ	266
CheckBoxコントロール	271
ComboBoxコントロール	271
OptionButtonsコントロール	272
ListBoxコントロール	273
Labelコントロール	278
ProgressBarコントロール	278
CommandButtonコントロール	278
DBListBoxコントロール	279
DBQueryBoxコントロール	282
DBEditコントロール	285
DBTableコントロール	286
DBPathコントロール	287

LinkEditコントロール . . . . .	287
TextBoxコントロール . . . . .	289
CHARTコントロール . . . . .	290
FILEEDITコントロール . . . . .	292
TICKEDITコントロール . . . . .	292
CALENDARコントロール . . . . .	293
TIMESPANEDITコントロール . . . . .	293
NUMBOXコントロール . . . . .	293
COMBOEDITコントロール . . . . .	294
DATETIMEEDITコントロール . . . . .	294
ウィザードの作成例 . . . . .	294
手順1：ニーズを分析する . . . . .	295
手順2：ウィザードの編成方法を定義する . . . . .	295
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 . . . . .	297
グラフィカルエディタの使用法 . . . . .	300
インタフェースの概要 . . . . .	301
新しいノードを作成する . . . . .	303
ノードのプロパティを編集する . . . . .	303
ウィザードをコンパイル、実行、およびデバッグする . . . . .	304
ウィザードに関する一般的な質問 . . . . .	304
質問 . . . . .	304
答 . . . . .	304
質問 . . . . .	305
答 . . . . .	305
質問 . . . . .	305
答 . . . . .	305
質問 . . . . .	305
答 . . . . .	305
質問 . . . . .	306
答 . . . . .	306
質問 . . . . .	306
答 . . . . .	306
質問 . . . . .	306
答 . . . . .	307
質問 . . . . .	307
答 . . . . .	307
質問 . . . . .	307
答 . . . . .	307
質問 . . . . .	307
答 . . . . .	308
15. ニュース . . . . .	309
ニュースの概要 . . . . .	309

ニュースの概要 . . . . .	310
新しいニュースを作成する . . . . .	310
ニュースを表示する . . . . .	310
ニュースの重要度 . . . . .	310
配信するメッセージ . . . . .	310
ニュースの配信先リスト . . . . .	310
<b>[すべての従業員グループ]</b> (SQL名:bAllGroups) チェックボックス . . . . .	311
<b>[サブグループを含む]</b> (SQL名:bChildGrps) チェックボックス . . . . .	311
ニュースを表示する . . . . .	311
ニュースマーカーを起動する . . . . .	311



# 図の一覧表

1.1. リストデータの詳細画面 . . . . .	20
3.1. AQLとSQLの比較 . . . . .	32
3.2. クエリエディタの活用法 . . . . .	45
4.1. 書式のヘッダー . . . . .	79
6.1. [実行可能ファイル] タイプのアクション . . . . .	102
6.2. 「LetterType.Doc」 . . . . .	103
6.3. 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 . . . . .	106
7.1. メッセージシステムの仕組み . . . . .	114
8.1. ワークフローチャートの概略 . . . . .	118
8.2. AssetCenterのワークフロー - 概要 . . . . .	121
8.3. ワークフローチャートを定義できるメインテーブル . . . . .	122
8.4. 実行中のワークフローに関連するメインテーブル . . . . .	122
8.5. ワークフローチャート - 依頼の承認 . . . . .	126
8.6. ワークフローチャート - 依頼の承認 . . . . .	141
8.7. [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 . . . . .	159
8.8. 同期ワークフローチャートの例 . . . . .	165
8.9. 非同期ワークフローチャートの例 . . . . .	166
8.10. 終了イベント付きのワークフローチャート . . . . .	167
10.1. スクリプトビルダ . . . . .	188
11.1. カレンダーの詳細画面の [予定表] タブページ . . . . .	207

11.2. カレンダの詳細画面の [ プレビュー ] タブページ . . . . .	211
14.1. ウィザードの構造 . . . . .	242
14.2. ウィザードの編成 . . . . .	296
14.3. 実行およびデバッグ ボタン . . . . .	302



# 表の一覧表

1.1. システムリストデータの値 . . . . .	21
3.1. 表記法 . . . . .	50
3.2. 論理演算子 . . . . .	55
3.3. 比較演算子 . . . . .	56
3.4. 集合型のAQL関数 . . . . .	61
3.5. 文字列型のAQL関数 . . . . .	61
3.6. 日付型のAQL関数 . . . . .	62
3.7. 日付型のAQL関数 . . . . .	63
3.8. 数値型のAQL関数 . . . . .	64
3.9. テスト型のAQL関数 . . . . .	64
8.1. ワークフローを制限する場合 . . . . .	160
8.2. イベントの様々な処理方法 . . . . .	161
8.3. イベントの様々な処理方法 . . . . .	163
10.1. 使用可能なデータ型 . . . . .	189
13.1. 特殊フィールドのタイプ . . . . .	228
14.1. 表記法 . . . . .	237
14.2. ROOTノードの論理プロパティ . . . . .	251
14.3. ROOTノードの物理プロパティ . . . . .	253
14.4. ROOTノードのサブノード . . . . .	257
14.5. PAGEノードの論理プロパティ . . . . .	258
14.6. PAGEノードの物理プロパティ . . . . .	258
14.7. PAGEノードのサブノード . . . . .	259

14.8. TRANSITIONノードの論理プロパティ . . . . .	260
14.9. FINISHノードの論理プロパティ . . . . .	261
14.10. FINISHノードの物理プロパティ . . . . .	262
14.11. STARTノードの論理プロパティ . . . . .	263
14.12. TIMERノードの論理プロパティ . . . . .	264
14.13. 倍長整数および文字列ノードの論理プロパティ . . . . .	265
14.14. すべてのコントロールに共通の論理プロパティ . . . . .	266
14.15. すべてのコントロールに共通の物理プロパティ . . . . .	268
14.16. CHECKBOXコントロールのプロパティ . . . . .	271
14.17. COMBOBOXコントロールの物理プロパティ . . . . .	271
14.18. OPTIONBUTTONSコントロールの物理プロパティ . . . . .	272
14.19. LISTBOXコントロールの物理プロパティ . . . . .	273
14.20. LISTBOXコントロールのメソッド . . . . .	276
14.21. LISTBOX コントロールの必須論理プロパティ . . . . .	277
14.22. LABELコントロールの物理プロパティ . . . . .	278
14.23. PROGRESSBARコントロールの物理プロパティ . . . . .	278
14.24. COMMANDBUTTONコントロールの物理プロパティ . . . . .	279
14.25. DBLISTBOXコントロールの物理プロパティ . . . . .	280
14.26. DBQUERYBOXコントロールの物理プロパティ . . . . .	282
14.27. 標準モードのDBEDITコントロールの物理プロパティ . . . . .	285
14.28. DBPATHコントロールの必須論理プロパティ . . . . .	287
14.29. LINKEDITコントロールの論理プロパティ . . . . .	287
14.30. TEXTBOXコントロールの物理プロパティ . . . . .	289
14.31. CHARTコントロールの論理プロパティ . . . . .	290
14.32. CHARTコントロールの物理プロパティ . . . . .	290
14.33. FILEEDITLコントロールのプロパティ . . . . .	292
14.34. TICKEDITコントロールのプロパティ . . . . .	292
14.35. NUMBOXコントロールのプロパティ . . . . .	293
14.36. COMBOEDITコントロールのプロパティ . . . . .	294
14.37. コントロールのプロパティ . . . . .	294
15.1. マーキーツールバーのボタン . . . . .	311

# 1 | リストデータ

---

リストデータは、優先度、場所、国、メーカなど、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

## 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。




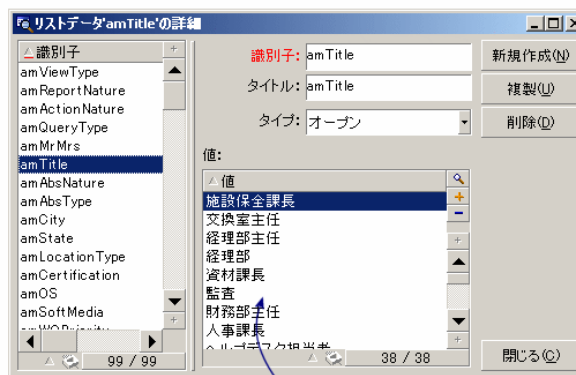
管理者は、リストの右側に表示される 、 および  ボタンを使って値の削除、変更、および追加を行うことができます。

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。

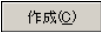
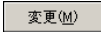
ユーザが新しい値を入力すると、この値がリストデータの値の一覧に追加されます（値の一覧はすべてのユーザで共有できます）。追加する際、作成を確認するメッセージが表示されます。

## 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  または  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。

システムリストデータは、【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。

資産の詳細画面の【割当】（SQL名：seAssignment）フィールドの例は、次の通りです。

表 1.1. システムリストデータの値

データベースに保存される値	表示される値
0	使用中
1	在庫中
2	除却済み
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。

## 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  をクリックして変更を確定します。

### 注意:

フィールドの設定画面で  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）：変更を行った担当者（ログイン名と氏名）
- **【フィールド】**（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、実行する操作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】**：「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】**：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】**：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】**：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

各操作について以下の内容が記録されます。

### 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】**：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】**：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】**：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

---

本章では、AQLでクエリを書く方法について説明します。内容は次の通りです。

## はじめに

ここでは、クエリが必要になる状況と、AQL言語について説明します。内容は次の通りです。

- AssetCenterのクエリ
- AQL

## 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、組み合わせることができます。

AssetCenterのパッケージプログラムでは、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とAssetCenter WebKit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に使うことはできません。

## 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。

#### 注意:

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、互換性がありません。AQLは、これらのデータベースエンジンに関係なく使うことができます。

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

これは、どのデータベースエンジンを使用しているても、AQLがデータベースと同一の関数セットを使用しているためです。

例えば、AQLのSubstring関数はOracle SQL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。

並べ替えと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、資産を検索する（メーカーで並べ替えたり、インデックスを使用する）場合は、次のAQLを記述します。

```
SELECT FIRST_ROWS Brand FROM amAsset ORDER BY Brand
```

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 7.3では、次のSQLコードが生成されます。

```
SELECT /*+ FIRST_ROWS INDEX_ASC(a1 Ast_Brand) */ a1.Brand FROM amAsset a1 WHERE a1.Brand >= CHR(0)
```

SYBASE 11では、次のSQLコードが生成されます。

```
SELECT a1.Brand FROM amAsset a1(Index Ast_Brand)
```

Microsoft SQL Server 6.5では、次のコードが生成されます。

```
SELECT a1.Brand FROM amAsset a1(Index=Ast_Brand)
```

## AssetCenterデータベースへのアクセスを簡略化

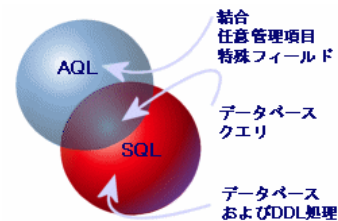
AQLは、リンクと結合の管理を簡略化します。そのため、AQLでクエリを記述すると、直接SQLを使う場合に比べ、速くデータベースにアクセスできます。

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とSQLを比較する

図 3.1. AQLとSQLの比較



上の図は、AQLとSQLとの比較を示しています。

- AQLは、SQLのデータベースクエリステートメント（SELECT）と互換性があります。
- AQLには、データベースに記述するSQLステートメント（INSERT、UPDATE、DELETE）に相当するものも、DDL（Data Definition Language：データ定義言語）ステートメントに相当するものもありません。

### 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。

- 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。

## 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」と「AQLの関数」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。

## 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、資産タグごとにそのユーザと責任者の名前を返します。

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

次は、Oracle SQLで記述した同じクエリです。

```
SELECT B1.AssetTag, U2.Name, R3.Name FROM amAsset B1, amEmpDept U2,
amEmpDept R3 WHERE B1.UserId = U2.EmpDeptId AND B1.SuperVId =
R3.EmpDeptId
```

資産のテーブルと部署と従業員のテーブル間の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。

 **注意:**

Oracle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
```

## 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数値（32ビットの整数）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、それぞれ資産タグと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 amAsset
```

このクエリの結果には、ユーザや責任者に割り当てられない資産が含まれます。データベースでは、このような資産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、資産ごとにその資産タグとユーザの所在地の名前を検索します。結果には、ユーザのない資産とユーザの所在地名がない資産が含まれます。

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

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 amAsset a, amEmpDept e, amLocation l WHERE a.UserId *= e.EmpDeptId AND e.LocId *= l.LocId
```

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 amAsset a, amEmpDept e, amLocation l WHERE a.UserId = e.EmpDeptId AND e.LocId = l.LocId
```

上記のクエリでは、「ユーザ」と「所在地」のリンクが依然として部署と従業員または所在地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 amAsset
```

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 of 「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 <text field> = NULL
```

```
WHERE <text field> IS NULL
```

```
WHERE <text field> = "
```

未入力 of 「日付」または「日付+時刻」型のフィールドについては、次のシンタックスを使ってデータベースにNULL値を保存することができます。

```
WHERE <date or date+time field> = NULL
```

```
WHERE <date or date+time field> IS NULL
```

```
WHERE <date or date+time field> = []
```

 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 lLocald=0

## 例：Location=0 または lLocald=0

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 amEmpDept
```

これは、次のクエリに相当します。

```
SELECT Name + "," + FirstName + "(" + Phone + ")" FROM amEmpDept
```

## 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資産を検索します。

```
SELECT IAstId FROM amAsset WHERE User = CurrentUser
```

### リンクとして使う

CurrentUserは、ま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。

## 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。

## 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

- 資産のテーブルの【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/'
```

- 場所のテーブルの【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、部署と従業員のテーブルを場所のテーブルへつなぐ「所在地」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が次の表記文字列で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 amAsset
```

これは次のクエリと同等です。

```
SELECT user.self FROM amAsset
```

これ自体は次と同等です。

```
SELECT User.Name + ", " + User.FirstName + "(" + User.Phone + ")" FROM amAsset
```

## 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、資産のテーブルでSQL名が「XXX」の任意管理項目の値を検索します。

```
SELECT fv_XXX FROM amAsset
```

## 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（ORDERBY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。

### 注意:

SQL Anywhereでは、上記のいずれかを選択できるわけではありません。データベースエンジンが最適な方法を自動的に選択します。

## 例

次のクエリについて説明します。

```
SELECT IAstId, Brand FROM amAsset 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。

- 設定するリストに移動します。
- マウスの右ボタンをクリックしてポップアップメニューを表示します。
- [ 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
インデックスを使わないリスト	NULL NULL A a B b C D

## Microsoft SQL Server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。

## Sybase SQL Anywhere

Sybase SQL Anywhereでは、AQLクエリでインデックスを使うことはできません。

データにアクセスしてデータを並べ替える最適な方法は、データベースエンジンが決定します。

## 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有利であるかを、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。

## 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、クエリの実行と結果を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。

## 概要

クエリエディタでは、次のいずれかの方法でクエリを設計できます。

- 簡単にクエリを設計できるグラフィカルユーザインターフェースを使う。
- AQLで直接クエリを記述する。

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 3.2. 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] <champ>[, <champ>...] FROM <table> [WHERE <clause>]
[ORDER BY <clause>]
```

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='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='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。表示されるウィンドウには、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。

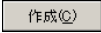
### 手順1：クエリの詳細画面の上部のフィールドに入力する

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、 [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。


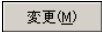
クエリの基本的な情報を入力してから  をクリックすると、クエリの詳細を指定するタブページを使うことができます。

## 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



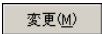
AssetCenterのクエリエディタを使うと、フィールド、計算式、定数、演算子を組み合わせる条件を定義できます。

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  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、 ボタンまたは  ボタンで条件を連結します。
- 3  をクリックし、クエリの変更を確定します。

 **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

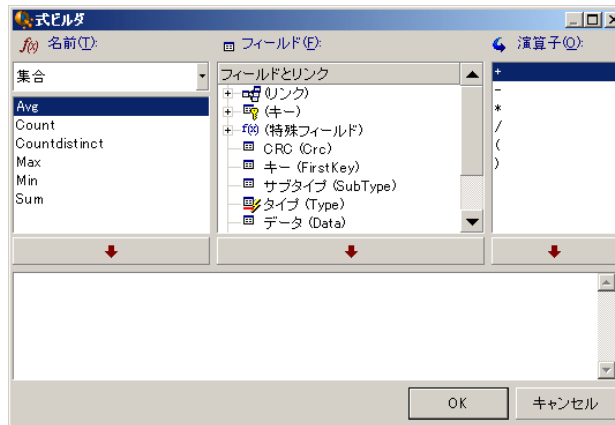
式  を使うとクエリで計算を実行できます。例えばCount関数を使うと、クエリによって検索されたレコード数をカウントできます。

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。




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、 をクリックすると、クエリの詳細画面の【フィルタ (WHERE句)】タブページに画面が切り替り、定義した式が表示されます。

## 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」のすべての資産を検索する場合は、資産のテーブルの【メーカー】（SQL名：Brand）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>]
[WHERE <句>]
[GROUP BY <句>]
[HAVING <句>]
[ORDER BY <句>]
```

### 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> ]
[WHERE <句> ]
[GROUP BY <句>]
[HAVING <句>]
)
```

 **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 dPrice >= (SELECT Max(dPrice)/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>]
[WHERE <句>]
```

```
[GROUP BY <句>]
[HAVING <句>]
[ UNION | UNION ALL | INTERSECTS | MINUS
SELECT <選択リスト>
[FROM <句>]
[WHERE <句>]
[GROUP BY <句>]
[HAVING <句>]...
[ORDER BY <句>]
```

## FROM句：クエリの対象となるテーブル

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。

### シンタックス

AQLでは、テーブル名のエイリアスを利用できます。

FROM句の書式は次の通りです。

```
FROM <テーブル> [<エイリアス>][, <テーブル> [<エイリアス>] ... ]
```

例

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

### 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は見なしません。AQLのFROM句はSQLのFROM句とは異なります。

例えば、次のサブクエリでは、AQLは【資産】テーブル（SQL名：amAsset）の【AssetTag】フィールドを検索します。

```
SELECT AssetTag FROM amAsset
```

### 使用例

次のクエリはすべて同じように機能します。

```
SELECT AssetTag FROM amAsset
SELECT AssetTag FROM amAsset a
SELECT a.AssetTag FROM amAsset a
SELECT amAsset.AssetTag FROM amAsset a
SELECT amAsset.AssetTag
SELECT amAsset:AssetTag
```

## 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。

## 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。

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

モデルのテーブルの例

```
Brand
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

'Computer '

'Monitor'

### 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#
#1999-01-01 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	意味
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]の資産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	意味
=	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。



演算子	意味
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	重要 : 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。

#### 注意:

SQL Anywhereでは、LIKE X句のXが128文字を超える場合は処理できません。Xが128文字を超えた場合にクエリを適用すると、ODBCエラーメッセージが表示されます。このエラーは、LIKE句で【FullName】フィールドを指定して、リストをツリー構造で表示する場合などに発生する可能性があります。

## 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

- 次のクエリでは、「府中支社」で使用しているメーカーの資産のリストを取得できます。

```
SELECT IAstId, Model.Brand FROM amAsset WHERE Model.Brand = ANY (SELECT Model.Brand FROM amAsset WHERE PortfolioItem.location.fullName=/府中支社')
```

## 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ステートメントに指定します。

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 amEmpIDept
```

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ステートメントに特定の数の式しか記述できない場合があります。

## WHERE句

AQLのWHERE句は、SQLのWHERE句と同様に機能します。

WHERE句は、検索条件を指定する時に使用します。

WHERE <検索条件>

検索条件には、データベースから抽出する項目を指定します。検索条件ではWHEREまたはHAVING句を使うことができます。

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。

GROUP BY <集合を使わない式>

[, <集合を使わない式>]...

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句を使った例と使わない例は以下の通りです。

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) 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, count(IAstId) 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。

HAVING <検索条件>

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せんが、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せんが、結果として得られる項目数に影響はありません。

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, count(IAstId) FROM amAsset GROUP BY Model.Brand
HAVING Model.Brand > '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, count(IAstId) FROM amAsset WHERE Model.Brand > 'B'
GROUP BY Model.Brand
```

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, count(IAsId) FROM amAsset GROUP BY Model.Brand  
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。

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

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

## 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。この結果得られたコードを移植することはできません。

---

## 集合型のAQL関数

表 3.4. 集合型のAQL関数

関数	説明
Avg( <列> )	数値型の列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。
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

関数	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。
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。
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)。

関数	説明
Upper( <文字列> )	<文字列>を大文字で返します。

## 日付型のAQL関数

表 3.6. 日付型のAQL関数

関数	説明
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 ( 1997 など )。
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 )。
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 )。
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。
Hour( <時間> )	時刻または日付+時刻型のフィールドの1日の時間を表す数字を返します ( 0-23 )。
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 )。
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 )。
Getdate()	サーバの現在のシステム日付を返します。
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。
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。
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 )

関数	説明
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)
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)
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)
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。
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。

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(dtActualFixStart)) =
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(dtActualFixStart)) = DayOfYear(DbToLocalDate(GetDate()))
```

## 数値型のAQL関数

表 3.8. 数値型のAQL関数

関数	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。
Floor( <数値> )	「数値」以下で一番大きい整数を返します。
Mod( <a>, <b> )	「a」を「b」で割った余りを返します( $a = qb + r$ 、ただし $q$ は整数で $0 \leq r < q$ )。
Round( <a>, <n> )	「a」を小数点以下第「n」位で四捨五入します。
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, -1) = 30.00.

Round (31.16, -1) = 30.00.

Trunc (31.16, 1) = 31.1.

## テスト型のAQL関数

表 3.9. テスト型のAQL関数

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



## クエリの例

次に、特定のクエリの例を紹介します。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。

次のように完全なシンタックスで記述されたクエリがあるとします。

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

以下に示すクエリ例は、AssetCenterプログラムに付属しているデモ用データベース（サンプルデータベース）に格納されています。

### 注意:

使用中のDBMSのSQLコードに書き換えたクエリは、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」のすべての資産

```
SELECT Self FROM amAsset WHERE Model.Brand = '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」に直接リンクしたすべての部署と従業員

```
SELECT Self FROM amEmpIDept 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：場所名に「大阪支社」が付いているすべての関連場所

```
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には、[ FullName ] フィールドと [ sLvl ] フィールドを頻繁に使います。

例：「大阪支社」に関連する従属場所で、階層のレベルが3より低いレコードツリー構造のルートの階層レベルは「0」です。

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/大阪支社/%') 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「会計部長」で場所が「府中支社」のすべての従業員

```
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：1995年1月1日から1995年12月31日までに実行したすべての作業指示

```
SELECT Self FROM amServiceCall WHERE (dtFirstCall >= #95/01/01 00:00:00#) AND (dtFirstCall <= #95/12/31 00:00:00#)
```

## 任意管理項目に関するクエリ

例：SQL名 [ Size ] フィールドの値が150 cm以上のすべての任意管理項目

```
SELECT Self FROM amAsset 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 amEmpDept 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 = 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。親資産のない資産を選択します。
- nリンク (0 = (SELECT COUNT(a.IAstId) FROM amAsset a WHERE a.IParentId = IAstId)) のテスト。従属資産のない資産を選択します。nリンクのテストは、各資産の識別子 [IAstId] を選択し、[IParentId] 識別子が [IAstId] に等しいすべての資産をカウントします。

## 別の組み合わせの例

「サウンドカード」属性の従属コンポーネント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 = 'サウ
ンドカード')))
```

 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：部署と従業員のテーブル

クエリは次の通りです。

```
SELECT Self FROM amEmplDept WHERE (Training_1.Title = 'Peregrine') AND
(Training_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リンクでリンクしている2つの異なるレコードごとに別々の条件を定義できます。

以下のように記述すると、

```
SELECT Self FROM amEmplDept WHERE (Training.Title = 'Peregrine') AND
(Training.Title = 'データベース')
```

クエリは、2つの研修コースを両方とも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 (Training.Title = 'Peregrine') OR  
(Training.Title = 'データベース')
```

クエリは、2つの研修コースのいずれかを受講したすべての従業員を選択します。



# 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.dbb」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 『データベース管理』マニュアルの、「AssetCenterデータベースの作成」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。
- 7 **【インポートシナリオの使用】**オプションを選択します。
- 8 **【インポートするデータ】**枠内で**書式**を選択します。
- 9 **【作成】**をクリックし、データベースの作成を確定します。

## 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.dbb」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 **【データベース】**フィールドで、データベースを選択します。
- 7 **【データベースの作成】**と**【システムデータの作成】**オプションをオフにします。
- 8 **【インポートシナリオの使用】**オプションを選択します。
- 9 **【インポートするデータ】**枠内で**書式**を選択します。
- 10 **【作成】**をクリックし、書式のインポートを確定します。

## 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。

### ! 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

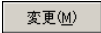
書式を編集するには、まずオブジェクトを定義してから、そのオブジェクトをページ内に配置します。

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）。
	レコードのリストを挿入します。 このツールを使うと、ページ内にリストを配置できます。 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。

- 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  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

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:" AssetTag " / " Brand
```

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。

## 画像

画像（ロゴなど）を挿入できます。

 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。

## 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。

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

テキストを直接入力します。

## 計算式

計算式の特殊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、文字の色

16色から選択できます。

## 行揃え


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


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

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

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

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

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

 **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います。

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



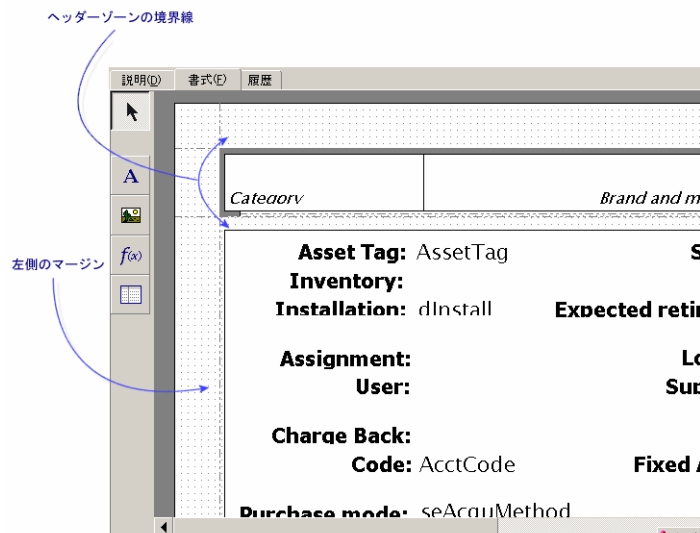
### 注意:

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(M)** 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。

## 定期レポートを簡単に作成する方法

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

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。

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 [ 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/



# 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。

---

 注意:

Crystalレポートは、AssetCenterのUnixバージョンでは使用できません。

---

## 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/rptフォルダにあります。

「reports.txt」ファイルはレポートのインポートスクリプトに使用されるため、このファイルの構造は変更しないでください。ただし本番用データベースへ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。

## 既存のレポートの印刷にフルバージョンのCrystal Reportsは不要

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。

## 既存のレポートの変更や新しいレポートの作成にはフルバージョンのCrystal Reportsが必要

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。

<b>サポートされるCrystal Reportsバージョン</b>	バージョン7.xと8.0 日本語版：バージョン7.0
------------------------------------	-------------------------------

<b>サポートされるCrystal Reportsの言語</b>	<ul style="list-style-type: none"> <li>• フランス語</li> <li>• 英語</li> <li>• ドイツ語</li> <li>• スペイン語</li> <li>• イタリア語</li> <li>• 日本語</li> <li>• ポーランド語</li> </ul>
----------------------------------	--

Crystal ReportsとAssetCenterではそれぞれ別の言語を使用できます。

## 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.dbb」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 『データベース管理』マニュアルの、「AssetCenterデータベースの作成」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。
- 7 **【インポートシナリオの使用】**オプションを選択します。
- 8 **【インポートするデータ】**枠内でCrystal Reportsレポートを選択します。
- 9 **【作成】**をクリックし、データベースの作成を確定します。

## 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、

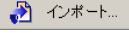
- 1 AssetCenter Database Administratorを起動します。
- 2 **【ファイル/開く】**メニューを選択します。
- 3 **【データベース記述ファイルを開く（新規データベースの作成）】**オプションを選択します。
- 4 AssetCenterのインストール先フォルダのサブフォルダ「config」に格納されている「gbbase.dbb」を選択します。
- 5 **【アクション/データベースの作成】**メニューを選択します。
- 6 **【データベース】**フィールドで、データベースを選択します。
- 7 **【データベースの作成】**と**【システムデータの作成】**オプションをオフにします。
- 8 **【インポートシナリオの使用】**オプションを選択します。
- 9 **【インポートするデータ】**枠内でCrystal Reportsレポートを選択します。
- 10 **【作成】**をクリックし、レポートのインポートを確定します。

## レポートの詳細設定

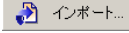

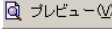
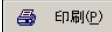
レポートのリストを表示するには、**【ツール/レポート機能/レポート】**メニューを選択します。

AssetCenterのレポートの詳細画面で、次の情報を指定します。

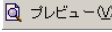
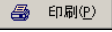
## ファイル

このフィールドを直接編集することはできません。このフィールドには、 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。

-  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。
-  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。
-  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、レポートを印刷する前にプレビューすることができます。
-  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。

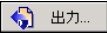
### 注意:

 ボタンまたは  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  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/FormulaField】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名の太文字と小文字を区別して入力してください。

例えば、資産の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、資産の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、 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  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。

## 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/



# 6 | 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】

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。デフォルトでは数種の機能ドメインがソフトウェア内に作成されています。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】**枠内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内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 **【新規作成】**をクリックします。
- 2 **【管理/機能ドメイン】**メニューを選択します。
- 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。

## アクションを作成する

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、様々タイプのアクションを定義できます。

### 注意:

AssetCenterを使うと、[実行可能ファイル]、[メッセージ]、または[印刷]タイプのアクションを作成できます。[DDE]、[スクリプト]、および[ウィザード]タイプのアクションの場合は、既定のアクションを実行することしかできません。

## [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」形式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（部署と従業員のテーブル）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]** フィールド（SQL名：EMail）で、メッセージを受け取るためのアドレスを指定します。

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 **[プロフィール]** 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、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。

 注意:

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。

## [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『Peregrine Automated 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。

## 一般的な作成方法

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 ]**
- 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）に使用します。

 **注意:**

【SQL名】フィールドに入力しないと、AssetCenterが自動的に標準的なSQL名を生成して入力します。

- 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  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コマンドに応答するメインアプリケーションになります。

## 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。この例では資産の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] フィールド (SQL名: seActionType) 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と、メッセージの詳細画面で  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 (部署と従業員のテーブル) 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手書き形式で指定します。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:GeraldColombo/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：資産の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

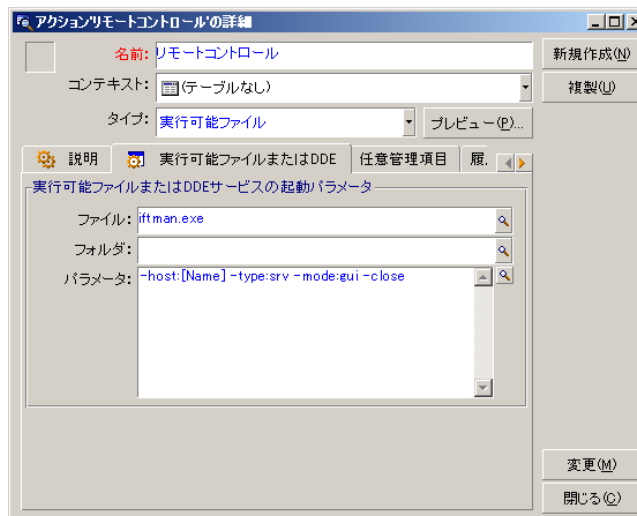
ここでは、AssetCenterによるアクションの実行例を紹介します。

- **【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 6.1. [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。
- その他

ここでは、簡単な [ DDE ] アクションについて説明します。

### アクションの目的

このアクション例では、購入依頼の確認を送信します。

このアクションは、購入依頼の詳細画面からトリガされます。

このアクションでは、Microsoft Word 7とAssetCenterとのDDEリンクを使います。依頼に関する情報（契約担当者の詳細と依頼番号）をWord文書に挿入し、この文書を印刷します。

## 準備：Wordでレターを作成する

まず、印刷するWord文書ファイル「LetterType.doc」を作成する必要があります。

このレターの構成は次の通りです。

### 図 6.2. 「LetterType.Doc」

```

!
TALTEK SERVICES
San Mateo Site
IT Department

<MrMrs> <FirstName> <Name>
<Adr1>
<Adr2>
<Zip> <City>

San Mateo. (DATE)

Dear <MrMrs>,

Following our conversation, I am pleased to confirm that your request has been attributed the
following request number: <ReqNo>.

Regards,

Procurement Manager
```

「LetterType.doc」文書を、AssetCenterのインストール先フォルダに格納します。  
対応するWordテンプレート「Normal.dot」を「LetterType.doc」にリンクします。  
このテンプレートには、次に示すマクロ「mymacro.bas」が含まれています。

```
Attribute VB_Name = "MyMacro"
Sub PrintLetterType(MrMrs, FirstName, Name, Adr1, Adr2, Zip, City, ReqNo)
'
' PrintLettreType Macro
'
    Application.WindowState = wdWindowStateMinimize 'Run Winword in the back end

    Documents.Open ("LetterType.doc") 'Open letter pattern
    Documents("LettreType.doc").Activate

    Selection.Find.ClearFormatting 'Clear parameters for Find function
    Selection.Find.Replacement.ClearFormatting 'Clear parameters for Replace function

    With Selection.Find
        .Text = "<MrMrs>"
        .Replacement.Text = MrMrs
        .Forward = True
    End With
End Sub
```

```
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll 'Execute replacement

With Selection.Find
.Text = "<FirstName>"
.Replacement.Text = FirstName
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<Name>"
.Replacement.Text = Name
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<Adr1>"
.Replacement.Text = Adr1
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<Adr2>"
.Replacement.Text = Adr2
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<Zip>"
.Replacement.Text = Zip
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<City>"
.Replacement.Text = City
```



```

End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
    .Text = "<ReqNo>"
    .Replacement.Text = ReqNo
End With
Selection.Find.Execute Replace:=wdReplaceAll

ActiveDocument.PrintOut 'Print document
ActiveWindow.Close (wdDoNotSaveChanges) 'Close document w/o updating
MsgBox ("Your document is being printed.") 'Notify user
End Sub

```

## 手順1 : AssetCenterでアクションを作成する

AssetCenterで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】**メニューを選択して、アクションのテーブルを開きます。
- 2 **新規作成(N)** をクリックし、アクションを作成します。
- 3 このアクションで表示する購入依頼のテーブルを **【コンテキスト】** (SQL名: ContextTable) フィールドに指定します。
- 4 **【タイプ】** フィールドを **【DDE】** に設定します。
- 5 **【DDE】** タブページで、AssetCenterとMicrosoft Word7とのDDEリンクを定義します。

DDEタブページの各フィールドに次のように指定します。

- 1 **【サービス】** (SQL名: DDEService) フィールドを「Winword」に設定します。
- 2 **【トピック】** (SQL名: DDETopic) フィールドを「System」に設定します。
- 3 **【サービス開始パラメータ】** セクションでは、Winword.exeとそのパスを指定します。
- 4 **【コマンド】** (SQL名: DDECommand) フィールドには、次のように起動するマクロとそのパラメータを指定します。

```

/[MyMacro.ImprimeLettreType "[Requester.MrMrs]", "[Requester.FirstName]",
"[Requester.Name]", "[Requester.Location.Address1]", "[Requester.Location.Address2]",
"[Requester.Location.ZIP]", "[Requester.Location.City]", "[ReqNumber]" /]

```

**作成(C)** をクリックし、アクションの作成を確定します。

## 手順2：アクションを起動する

アクションを起動するには、次の手順に従います。

- 1 購入依頼のテーブルを開きます。
- 2 購入依頼を選択します。
- 3 **[ツール/アクション]**メニューからアクションを起動します。  
アクションを起動すると、次の作業が行われます。

- 1 Microsoft Wordが起動し、「LetterType.doc」が読み込まれます。
- 2 連絡担当者の詳細と依頼番号がレターに挿入されます。
- 3 レターが印刷されます。

## [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、リース明細にリンクされている必要があります（**[取得]**タブページ）。参照するオブジェクトを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 6.3. 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 fields are as follows:

- 名前:** 期間満了
- コンテキスト:** 資産 (m.Asset)
- タイプ:** メッセージ
- 参照オブジェクト:** AcquContract.
- 優先度:** 普通
- 配信済通知
- 受信者:** [PortfolioItem.User.Email]
- 件名:** 契約の期間満了
- メッセージ:** 契約[AcquContract.cf\_self]は[AcquContract.dEnd]に期間満了します。

## [スクリプト]タイプアクション例




[スクリプト]タイプ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  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。  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スクリプトのビルダ画面を表示します。スクリプトタイプ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
<b>[名前]</b>	Name	PC
<b>[作成]</b>	seBasis	リポジトリ品目
<b>[接続可能]</b>	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 Irec As Long
Dim Ires As Long
Irec=AmCreateRecord("amNature")
Ires=AmSetFieldStrValue(Irec, "Name", "PC")
Ires=AmSetFieldStrValue(Irec, "seBasis", 1)
Ires=AmSetFieldStrValue(Irec, "blsCnxClient", 1)
AmInsertRecord(Irec)
```

 **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の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、ユーザが属性のテーブルを開き、 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、 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].docl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。

## Sybase SQL Anywhereデータベースエンジンを使う場合

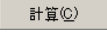
Sybase SQL AnywhereをAssetCenterのデータベースエンジンとして使う場合は、アクションのフィールドの先頭に「{d}」または「m」を記述することはできません。

アクションのフィールドの先頭に「{d}」または「m」を記述する必要がある場合は、文字列の前にスペースを1つ挿入してください。

## 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【プレビュー】ボタンを使います。

### 【計算】ボタン





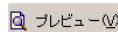
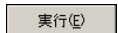
コンテキストを選択し、 ボタンをクリックします。これで【実行可能ファイル】、【DDE】、および【メッセージ】のいずれかのタブページのフィールドに値が入ります。【コンテキスト】(SQL名: ContextTable) フィールドで選択したレコードから、変数が正しく抽出されていることを確認します。

### 【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  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。
  -  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(N) >** 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(B)** 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(F)** 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。





# 7 | 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

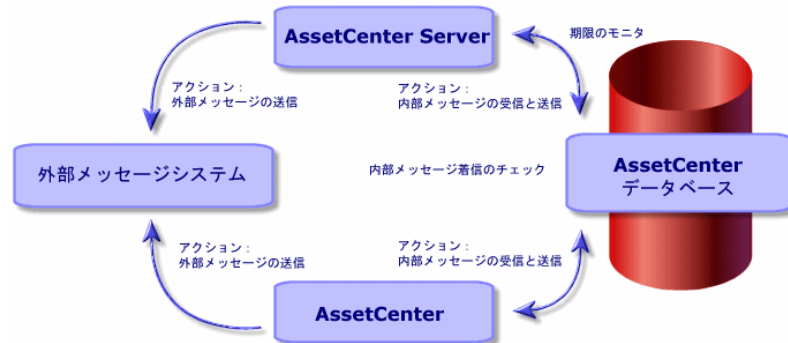
## メッセージシステムの仕組み

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 7.1. メッセージシステムの仕組み



## 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、本マニュアルの「アクション」の章、「アクションを作成する/[メッセージ]タブページに入力する」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 (部署と従業員のテーブル) 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。



# 8 | ワークフロー

---

本章では、ワークフローチャートを定義し、管理する方法について説明します。

## 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
- 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

ワークフローは、業務処理を標準化し、自動化する機能です。

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

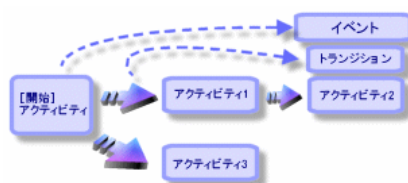
AssetCenterを使うと、ワークフローチャートを定義し、その進捗状況を管理することができます。

## ワークフローチャート

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。

- アクティビティ
- トリガイベント（アクティビティの結果発生したイベントで、トランジションを起動するイベント）
- トリガトランジション（アクティビティを起動するトランジション）
- コンテキスト
- 制限時間とアラーム

図 8.1. ワークフローチャートの概略



ワークフローチャート（SQL名：mWfScheme）のテーブル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]メニューを選択します。

## ワークフローインスタンス

「ワークフローインスタンス」とは、実行する定義済みワークフローチャートを指します。

## 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

ワークフローのアクティビティは、ワークフローのアクティビティのテーブル（SQL名：amWfActivity）に格納されます。

ワークフローのアクティビティの詳細画面を表示するには、[ツール/ワークフロー/ワークフローチャート]メニューを選択し、[アクティビティ]タブページのグラフィカルエディタを使います。

## 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。

ワークフローチャートに属するイベントは、ワークフローのイベント（SQL名：amWfEvent）のテーブルに格納されます。

これらのイベントが発生すると、ワークフローインスタンス（SQL名：amWfOccurEvent）の基本イベントのテーブルに記録されます。

ワークフローのイベントの詳細画面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]メニューを選択し、[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います。

## 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、1つ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。

ワークフローチャートに属するトランジションは、ワークフローのトランジションのテーブル（SQL名：amWfTransition）に格納されます。

ワークフローのトランジションの詳細画面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]メニューを選択し、詳細画面の[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います。

## 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。

ワークフローのタスクをワークフローのタスクのテーブル（SQL名：amWfWorkItem）に記録するには、アクティビティの詳細画面の[全般]タブページにある[タスクを保存]（SQL名：bLogWorkItem）チェックボックスをオンにする必要があります。

実行するワークフローのタスクのリストにアクセスするには、[ **ツール/進行中のタスク** ]メニューを選択します。

## ワークフローのアクティビティの担当者

担当者は、「質問」または「ユーザのアクション」タイプの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。担当者は、「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関わりません。ワークフローの担当者は、ワークフローの役割テーブル（SQL名：amWfOrgRole）に格納されます。ワークフローの役割の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の実行グループは、[ **ツール/ワークフロー/ワークフローチャート** ]メニューを選択し、[ **全般** ]タブページの[ **実行グループ** ]（SQL名：GroupName）フィールドで指定します。

## ワークフローの概要

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。これらの要素を、AssetCenterのアクションと従業員（ワークフローの担当者）に関連付けることができます。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モニタします。ワークフローエンジンは、次の内容を実行します。

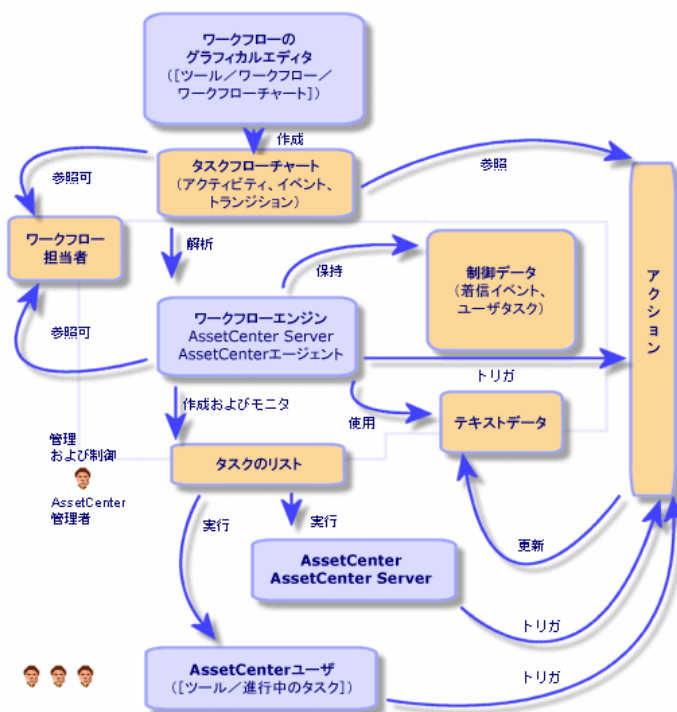
- 実行するタスクを生成する。
- これらのタスクとアクティビティを起動するイベントをモニタする。
- 受信するイベントと実行するユーザタスクの履歴を作り、イベントの推移を記録する。

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。



次の図に、AssetCenterでワークフローを実行する概要を示します。

図 8.2. AssetCenterのワークフロー - 概要

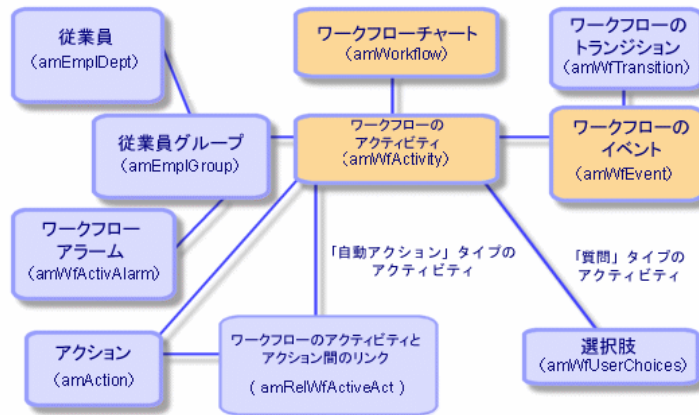


ワークフローインスタンスの処理は、アクティビティやイベントの属性、およびこれらを制御するワークフローエンジンの設定によって異なります。

## ワークフローに関連するメインテーブル

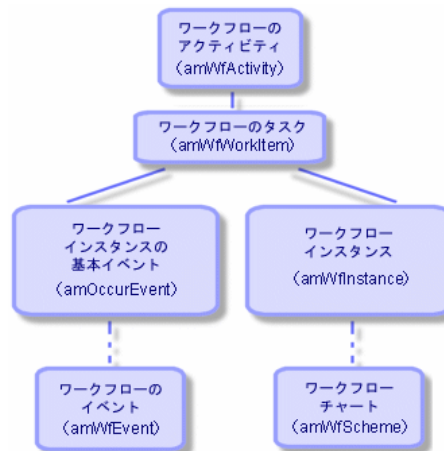
ワークフローに関連するメインテーブルとこれらの関係は、次の図の通りです。テーブルは、ラベルとSQL名で表記されています。

図 8.3. 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 8.4. 実行中のワークフローに関連するメインテーブル



## ワークフローのグラフィカルエディタの使用方法

ワークフロー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。

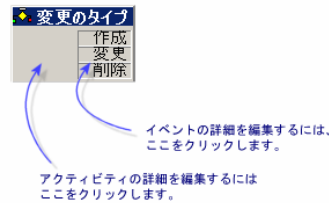
- [アクティビティ]タブページの空白の領域をマウスの右ボタンでクリックして、ポップアップメニューから[アクティビティの追加]コマンドを選択します。アクティビティの詳細画面が表示されます。
-  ボタンをクリックし、次にグラフィカルエディタ領域をクリックします。アクティビティの詳細画面が表示されます。

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

### [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 [ 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

[ システム ] イベントは、各イベントの起点のアクティビティから作成および削除できます。

[ システム ] イベントを変更するには、必要に応じて次のいずれかの方法を使います。

- イベントの処理方法 ( イベントの詳細画面の [ 処理 ] ( SQL 名 : seProcessingMode ) フィールド ) を変更するには、[ データベース ] または [ 定期的 ] タイプのイベントを変更するのと同じ方法をとります。


- それ以外の場合は、イベントの起点であるアクティビティの詳細設定を編集し、変更します。

## 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 ]** スライダーまたは  ボタンを使ったフローチャートの拡大と縮小

## ワークフローの活用方法

AssetCenterのワークフローを使って、会社の業務処理を自動化することができます。次にその手順を示します。

- 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
- 2 ワークフローチャートを使用する前に
- 3 アクティビティを作成する
- 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
- 5 開始イベントを作成する
- 6 トランジションを作成する
- 7 ワークフローの起動例

### 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 8.5. 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翌日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します。
- 5 会計監査役が購入依頼を許可すると、購入依頼は承認されます。  
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、【依頼ステータス】フィールド（SQL名：seStatus）は【却下】に設定されます。

## ワークフローチャートを使用する前に

「Admin」ログインを使ってデータベースに接続し、メッセージを設定する必要があります。（詳細は本マニュアルの「メッセージ」の章を参照してください。）

## 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

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）のテーブルに保存されます。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') window. The role is selected in the list, and its details are shown below:

- 役割: 依頼者の部署の責任者
- 参照番号: D001001
- コンテキスト: 依頼 (amRequest)
- タイプ: 特定の個人に割当
- スクリプト: Retval=[Requester.Parent.Supervisor]

## 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') window. The role is selected in the list, and its details are shown below:

- 役割: 会計監査役
- 参照番号: D001002
- コンテキスト: (テーブルなし)
- タイプ: 個人に割当
- 担当者: 木村五郎

## 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 window 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
- プレビュー(P)...
- 説明 (gear icon)
- メッセージ (envelope icon)
- 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。
- 作成(C)
- 作成(R)
- キャンセル

## 購入依頼の承認

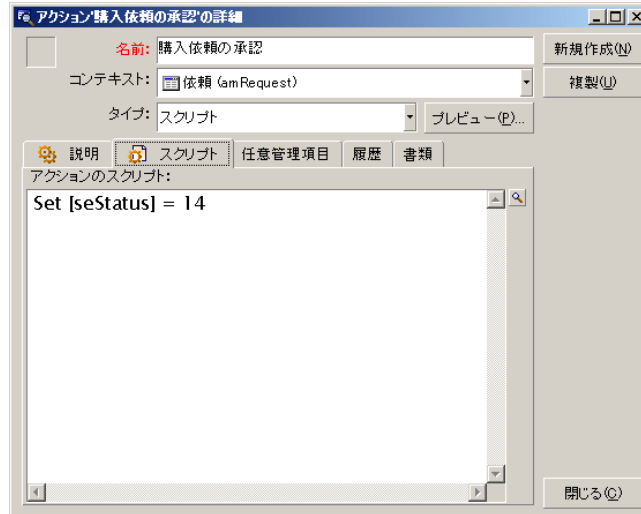
このアクションは、後に作成する「依頼の承認」アクティビティの時点で使用されます。

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。

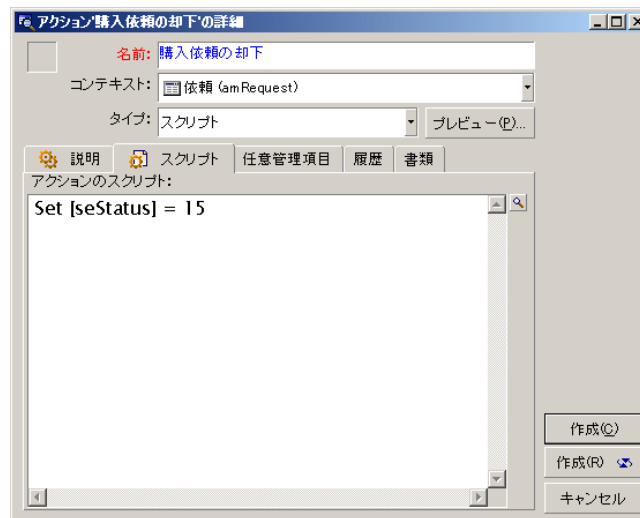
アクションの詳細は次のようになります。



## 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。



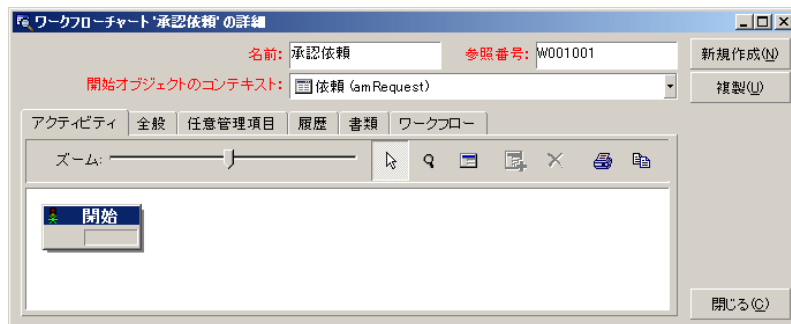
## カレンダーを作成する

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 : 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

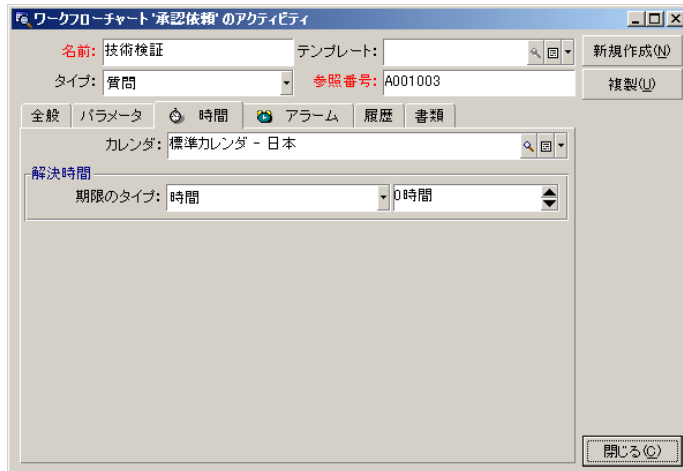
- 1 【パラメータ】タブページに次のように入力します。

- 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翌日までに質問に答えることにします。



## 「技術検証」アクティビティのアラームを定義する

**[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なります。

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 [ワークフローの役割] の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。

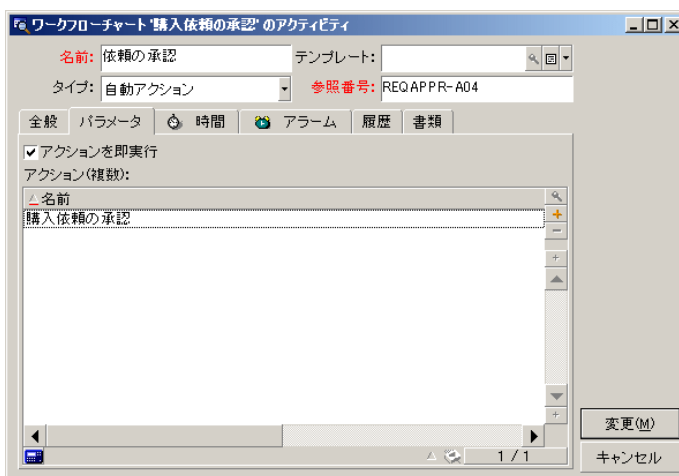
## 「依頼の承認」アクティビティを作成する

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。

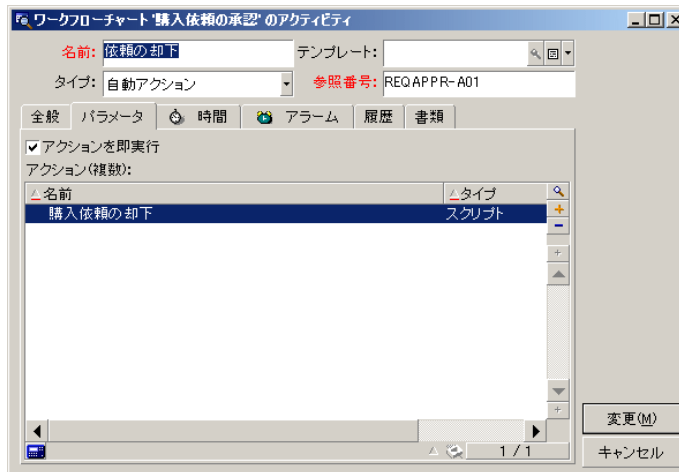
このアクティビティは [自動アクション] タイプのアクティビティです。実行されるアクションは、前の手順で作成した「購入依頼の承認」アクションです。



## 「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せん。

- 「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。

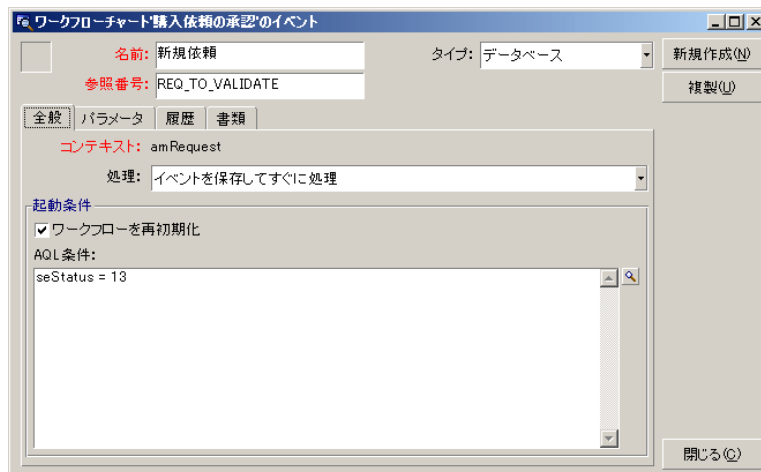
- 1 イベントの名前を右クリックします。
- 2 **[ イベントの詳細 ]**メニューを選択します。
- 3 **[ 全般 ]**タブページを選択します。
- 4 **[ 処理 ]** (SQL名 : seProcessingMode) フィールドで **[ イベントを保存してすぐに処理 ]**を選択します。
- 5 **[ 変更 ]**をクリックします。

## 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。

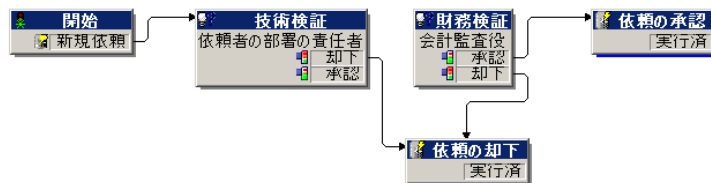
- 1 トランジションの開始イベントをクリックします。
- 2 マウスボタンを押したまま、ターゲットアクティビティまでイベントをドラッグします。

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 8.6. 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 部署と従業員のテーブルに入力する
- 2 承認を受ける購入依頼を作成する
- 3 ワークフローを制御する

## 部署と従業員の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、部署と従業員の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ァイルを指定します（対応するレコードの詳細画面の【プロファイル】タブページで指定）。デモ用データベース内のプロファイル「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）のテーブルにイベントが記録されます。

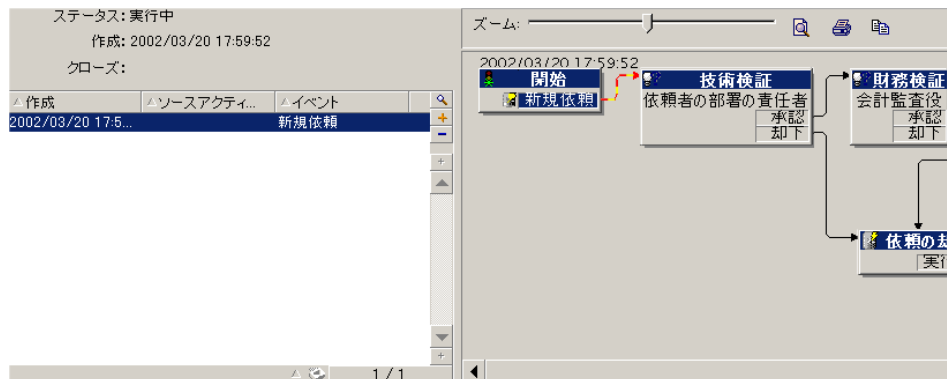
## ワークフローを制御する

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

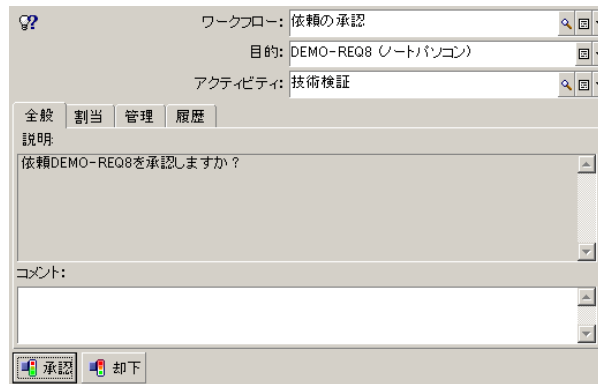
作成した購入依頼の詳細画面を表示します。[ワークフロー]タブページには、依頼の現在のワークフローが一覧表示されます。各ワークフローの詳細は、個々のサブタブページに表示されます。

- 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が。



## 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。

- ワークフローチャートの詳細画面上部にある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。この依頼に従って購入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購入発注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)。

## 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つの物件につき1つのワークフローを起動]オプションをオンにして、対象となる購入依頼を単一の承認プロセスで処理する。
- 開始イベントで[進行中のワークフローがある場合にワークフローを再度初期化]オプションをチェックして、購入依頼の構成が変更された場合にワークフローを再起動する。

## 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 (SQL名: amWfOrgRole) のテーブルで選択します。ワークフローの役割の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

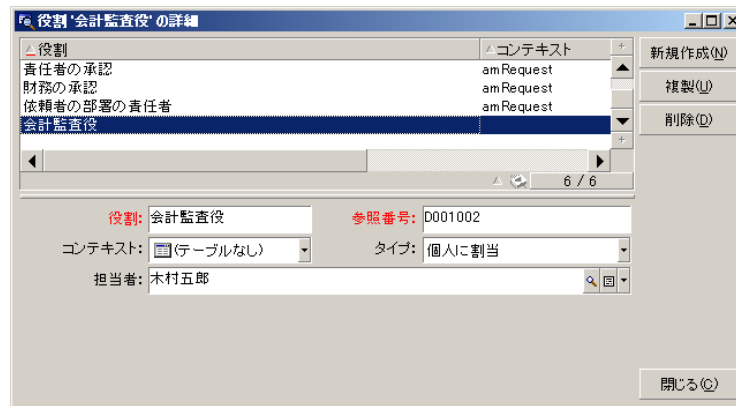
ワークフローの役割には次のタイプがあります（ワークフローの役割の詳細画面 / **[タイプ]**（SQL名：seType）フィールドで指定）。

- **[個人に割当]**
- **[特定の個人に割当]**
- **[グループに割当]**

### [個人に割当]

この場合担当者は、部署と従業員の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、部署と従業員の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（SQL名：amEmplGroup）から選択されず。

## 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

「ユーザのアクション」タイプのアクティビティには、「担当者」と呼ばれるユーザの操作が必要です。担当者は、[ 担当者 ] (SQL名: Assignee) フィールドに表示されます。

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の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、納品待ちの購入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

「テスト/スクリプト」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。

- 【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カイベントによってワークフローがトリガされません。

## 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) の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) フィールドで指定) を参照しているアクティビティ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 : seInCond) フィールドに入力する必要があります。これは、アクティビティを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）の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

実行する必要の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のテーブル (SQL名: amWfWorkItem) 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す。

## 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または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、 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。
- アクティビティが [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。

## ユーザにタスクを割り当てる

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
- 【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】**（SQL名：bInsert）、**【更新】**（SQL名：bUpdate）、および**【削除】**（SQL名：bDelet）チェックボックスを必要に応じてオンにします。  
**【挿入】**チェックボックスをオンにすると、作成されたレコードをモニタします。
    - **【更新】**チェックボックスをオンにすると、変更されたフィールドをモニタします。モニタするフィールドは、**【モニタしたフィールド】**（SQL名：MonitFieldsbox）フィールドで指定できます。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はモニタされません。

**[削除]** チェックボックスをオンにすると、破棄されたレコードがモニタされます。

**警告:**

コンテキストが参照するオブジェクトの破棄をイベントの起動条件にすることはできません。

- **[スクリプト]** (SQL名: memScript) 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す。

例: 既存の依頼の総額が変更されたときにイベントをトリガする場合は、**[パラメータ]** タブに次のように入力します。

図 8.7. **[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、「イベントの処理」の節を参照してください。

## [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。

下の表に、オプションの選択に応じた結果を示します。



表 8.1. 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
「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。

## イベントの処理

一般的な起動条件が満たされた場合、イベントは、次の指定に従って処理されます。

- イベントの「タイプ」(イベントの詳細画面上部の [タイプ] (SQL名: seMonitoringType) フィールド)
- イベントの詳細画面の [全般] タブページにある [処理] (SQL名: seProcessingMode) フィールドで選択されるオプション

下の表に、イベントの様々な処理方法を示します。

表 8.2. 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、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】	【イベントを保存しすぐに処理】	【イベントを保存しないですぐに処理】
<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>	<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。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>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せん。</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せん。</p>
<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>	<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。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>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。</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>

表 8.3. イベントの様々な処理方法

	<b>[ イベントを保存してすぐに処理 ]</b>	<b>[ イベントを保存しないですぐに処理 ]</b>
<b>[ 定期的 ] タイプのイベント</b>	<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<b>[ パラメータ ]</b>タブページで定義します。</p> <p>イベントが発生するとすぐに、AssetCenter Serverにより、<b>[ amWfOccurEvent ]</b>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<b>[ パラメータ ]</b>タブページで定義します。</p> <p>イベントが発生しても、<b>[ amWfOccurEvent ]</b>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 Serverによって即座に起動されます。</p>
<b>[ 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)</b>	<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、<b>[ amWfOccurEvent ]</b>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、<b>[ amWfOccurEvent ]</b>というSQL名のテーブルに保存されませんが、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>
<b>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</b>	<p>イベントが発生するとすぐに、AssetCenter Serverにより、<b>[ amWfOccurEvent ]</b>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、<b>[ amWfOccurEvent ]</b>というSQL名のテーブルに保存されませんが、トランジションは、AssetCenter Serverによって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

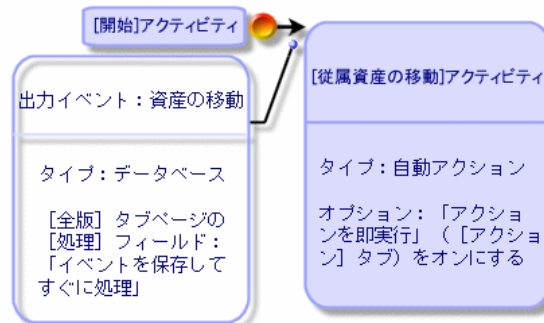
同期ワークフローチャートを作成するには、次の内容を定義する必要があります。

- **【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：bExeclmmediately）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 8.8. 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(M)** をクリックすると、次の処理が行われます。

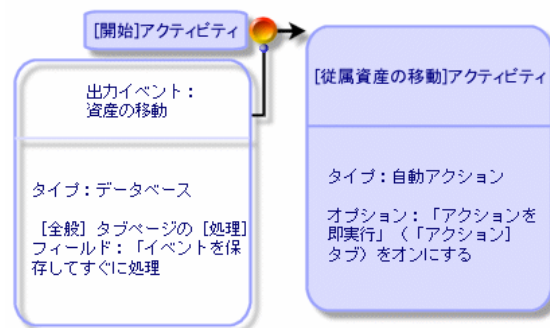
- 1 データベーストランザクションが開始します。
- 2 資産の設置場所が変更されます。
- 3 ワークフローが開始します。
- 4 ワークフローのトランジションが起動します。
- 5 従属資産の設置場所が変更されます。
- 6 トランザクション全体が検証されます。

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 8.9. 非同期ワークフローチャートの例



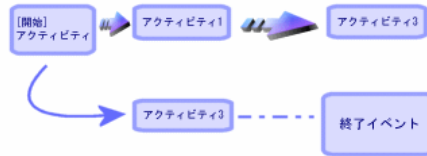
## 終了イベント

### 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 8.10. 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、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の実行グループは、【**ツール/ワークフロー/ワークフローチャート**】メニューを選択し、【**全般**】タブページ/【**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】**という新しいモジュールを作成します。

このメカニズムは、次の点で役に立ちます。

- 各ワークフローの実行グループに対して検証タイムテーブルを定義できる。
- 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。

## 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。

# 9 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。
- **[リストの出力]** 状況依存メニューを使う。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ず、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AssetCenterで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 ポップアップメニュー**【リストの出力】**を選択します。
- 3 表示されるウィンドウに入力し、 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(N)** 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(D)** 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(N)** をクリックします。
- 2 クエリ名を定義します。この名前は、出力スクリプトの詳細の **【メッセージ】**タブページの実行ログで使われます。
- 3 コメントを入力することは可能ですが、出力されません。
- 4 **【クエリ】**フィールドで、出力するデータを定義します。
- 5 ビューの作成、変更または削除ではなく、データを出力する場合は、**【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】**タブページでは、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(IASId) FROM amAsset 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:**

SQL Anywhereを使う場合以外は、**【インデックスの強制使用】**オプションをオンにすると、クエリが参照するインデックスを強制的に使用することができます。詳しくは、『AssetCenterの高度な使い方』、「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 amAsset
```

**【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 amAsset
```

## 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] オプションを選択した場合は、[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 回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。

- ① 一般情報
- エラー
- ① 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、事前に行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のインストール先ディレクトリのbinサブディレクトリにあるamexpl.exeプログラムを使って、手動でまたはバッチファイルを使って自動的に出力コマンドを実行できます。

### シンタックス

```
amexpl32 [-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:GeneraleDatabse -login:Gerald -password:PAssword
```



# 10 | 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、『**データベース管理**』、『**インポート**』の章、およびマニュアル『**AssetCenterの高度な使い方**』の『**データの出力およびSQLビューの作成**』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 ApplicationsTMと互換性があります。この言語、構造、およびシンタックスの詳細については、Basicの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。
- 特にUNIXでは、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.ILocalId]
```

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。


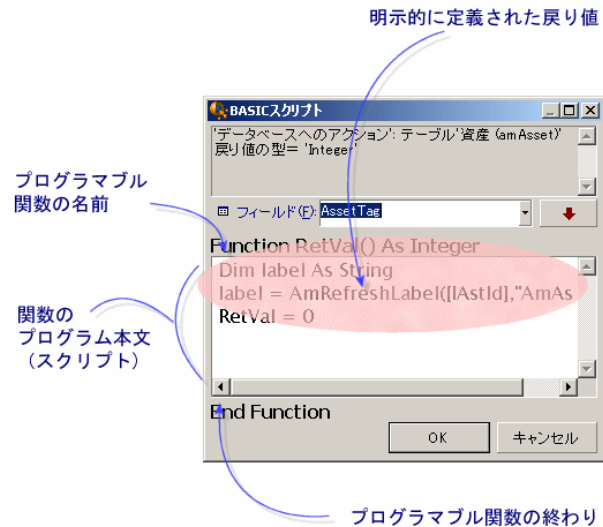
プログラマブル関数には、スクリプトビルダからアクセスできます（スクリプトを作成できるフィールドの  ボタンをクリックすると表示されます）。スクリプトビルダは、関数に対応するソフトウェアアイテムを、ユーザが簡単に作成できるように設計されています。スクリプトビルダには、プログラマブル関数を作成するためのテンプレートがあります。

図 10.1. スクリプトビルダ



プログラマブル関数の説明は、スクリプトビルダウィンドウの上部に表示されます。関数に関連するオブジェクト（例えば、資産テーブルの【バーコード】（SQL名：BarCode）フィールドのデフォルト値）と、予測される戻りコード（前の例では「文字列」）を確認できます。

## 関数とパラメータのデータ型

### 関数のデータ型

組み込み関数のデータ型は、関数から戻される値のデータ型になります。Basicスクリプトでは、コンパイルエラーや実行エラーの原因となることがあるので、この点によく注意してください。

例えば、あるデータ型のフィールドのデフォルト値を定義しているときに、別のデータ型の値を戻す関数は使用できません。例えば、下のデフォルトスクリプトに「日付」または「日付+時刻」型のフィールドを割り当てようとしたとします。

```
RetVal=AmLoginName()
```

AmLoginName()関数は接続ユーザ名を文字列で返します（文字列型）。そのため、この場合の戻り値の形式は、「日付」フィールドの形式とは互換性がない

ので、次に同じテーブルでレコードを作成すると、エラーメッセージが表示されます。

## パラメータのデータ型

組込み関数で使われるパラメータにもデータ型があります。実行する関数に適したデータ型を使う必要があります。パラメータのデータ型にエラーがある場合は、関数の実行時にエラーメッセージが表示されます。

## 使用可能なデータ型

次の表に、関数とパラメータで使えるデータ型とその説明を示します。

表 10.1. 使用可能なデータ型

データ型	説明
Integer (整数)	-32,768から+32,767の整数
Long (倍長整数)	-2,147,483,647から+2,147,483,646の整数
Double (倍精度)	8バイト浮動小数点数
String (文字列)	任意の文字からなるテキスト
Date (日付)	日付または日付+時刻
Variant (可変)	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。

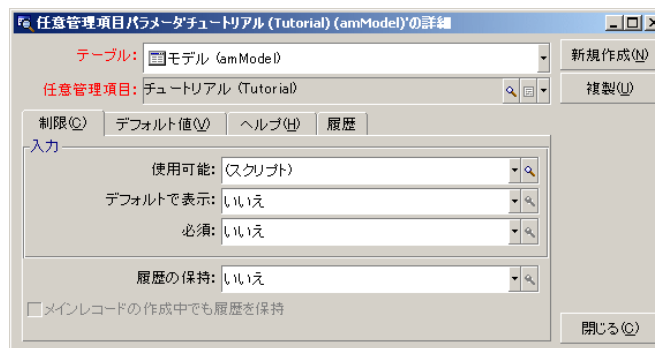
## 手順1：任意管理項目「チュートリアル」を作成する

【管理/任意管理項目】メニューを選択します。新規作成(N) 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	値
【タイトル】（SQL名：TextLabel）	チュートリアル
【SQL名】（SQL名：SQLName）	Tutorial（この例で使う名前）
【入力タイプ】（SQL名：seDataType）	【数値】

作成(C) 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。

```
If the model's full name starts with "/コンピュータ/マザーボード/" Then
The feature is available
Otherwise
The feature is not available
```

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から、モデルのテーブルから**【完全名】**（SQL名：FullName）フィールドを選択できます。

フィールドを選択したら、 ボタンをクリックして、編集ウィンドウに値を移します。


## 手順4：Basicスクリプトを作成する

Basicスクリプト作成用のウィンドウで、スクリプトを作成します。このスクリプトは、手順3で作成したBasicアルゴリズムを変換したものです。

```
If Left([FullName], Len("/コンピュータ/マザーボード/"))="/コンピュータ/マザーボード/" Then
  RetVal=1
Else
  RetVal=0
End If
```

 **注意:**



スクリプトでは、大文字と小文字は区別されません。

 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となります。

## 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「従業員」のテーブル (SQL名: amEmplDept) 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の様々な機能で使用します。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およびUnixでの日付形式

BasicとUnixでは、日付形式が異なります。

- 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。
- UNIXでは、Long（倍長）型の整数で記述します。現地時間に関係なく、UTC（協定世界時）で1870年1月1日の午前0時から現在までに経過した秒数を示します。

## 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
【除却済み】	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「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、ま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、ユーザのID番号を返します。
- 「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]

## 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。

- Basicスクリプト内にコメントだけの行を加える。特に注意事項はありません。

```
'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
```

- スクリプト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。この場合は、コロン(:)文字を使ってコメントとスクリプトを分離する必要があります。コメントの前に一重垂直引用符(')を付けます。

```
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 (<Error number>, <Error message>)
```

**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

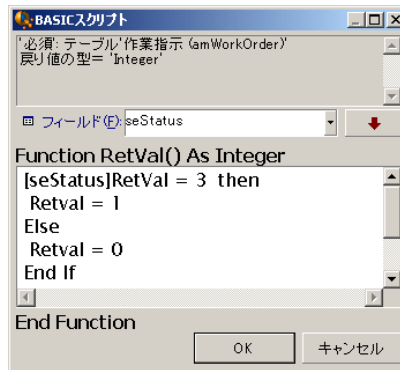
ここでは以下のリストデータの値を使用します。

- 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。



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かどうかを確認します。

- 1 【ポートフォリオ/作業指示】メニューを使って作業指示のテーブルを表示し、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となります。

## 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"
```

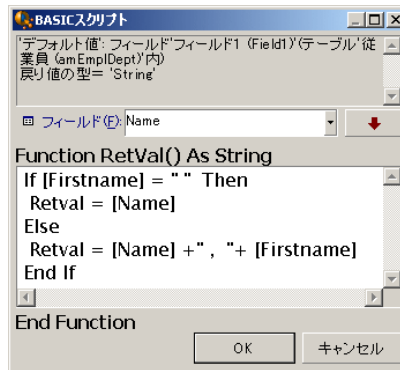
従って、従業員の詳細画面の【フィールド1】（SQL名：Field1）のデフォルト値の条件となるのは、従業員のテーブルの【部署名/姓】（SQL名：Name）フィールドと【名前】（SQL名：FirstName）フィールドの値です。アルゴリズムでは、これら2つのフィールドしか使いません。

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。



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かどうかを確認します。

- 1 **【ポートフォリオ/部署と従業員】**メニューを選択して従業員のテーブルを開き、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、 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。



# 11 | 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せん。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 11.1. カレンダーの詳細画面の**【予定表】**タブページ

	0	3	6	9	12	15	18	21	24
月曜日 (M):				■	■	■	■		
火曜日 (T):				■	■	■	■		
水曜日 (W):				■	■	■	■		
木曜日 (R):				■	■	■	■		
金曜日 (F):				■	■	■	■		
土曜日 (S):									
日曜日 (D):									

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[午前午後]>
```

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す。

#### 注意:

グラフィカルコントロールを使う場合は、30分単位で指定できます。直接時間を数値で入力する場合は、1分単位で指定できます。

## [ 例外 ] タブページに入力する

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

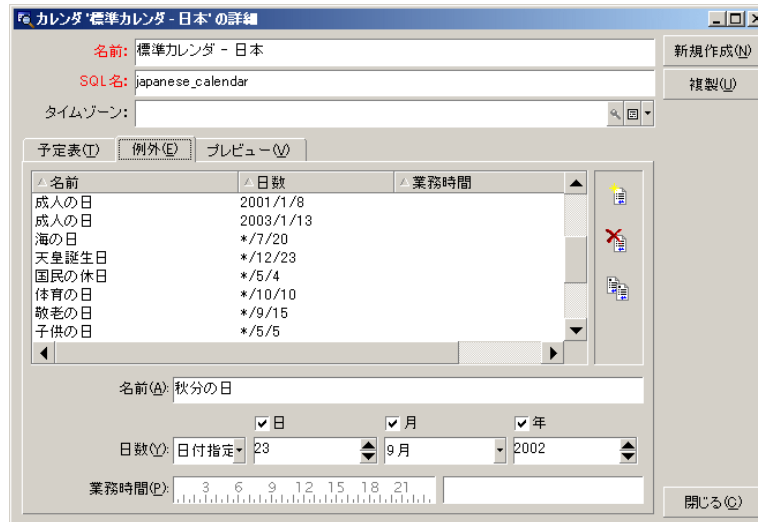
### 作成方法

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。


[日数]フィールドの値	例外期間
毎日	例外なくその年のすべての日が例外期間になります。
日付指定	[日]、[月]、および[年]チェックボックスを使って定義する日が例外期間になります。
第1	[月]および[年]チェックボックスをオンにした月および年の、[日]チェックボックスで定義した曜日が例外期間になります。 <b>例</b> 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。 <b>例</b> 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。 <b>例</b> 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。 <b>例</b> 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  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せん。

- 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。

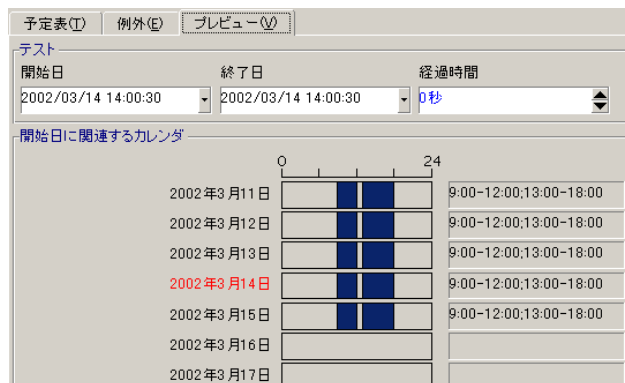
## 規則2：8月中は、午前8時30分から午後12時30分までの午前中だけ勤務する

- 1 **新規作成(N)** 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 11.2. カレンダーの詳細画面の【プレビュー】タブページ



- 【開始日に関連するカレンダー】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。

---

# 12 | 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。
- サーバには、「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参照した値が格納されます。
- 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として表示されず。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1998年5月15日の17時30分に設定されているとします。この場合、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のタイムゾーンの管理はWindowsと異なり、年間を通じて時間の変更規則を考慮します。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。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。 をクリックして実行するスクリプトを選択します。選択するファイルは、AssetCenterのインストール先フォルダの /datakit/standard サブフォルダにある「tz.scr」です。
- 3  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。

	<DaylightInfo>引数に値なし	<DaylightInfo>引数に値がある
<Year>引数に値がある	適用できません。	<Year>引数で指定した年から次の<Year>引数までの各年について、指定しているタイムゾーンの夏時間情報が有効になります。

## <Year>引数の値

<DaylightInfo>で定義された夏時間変更情報を適用する年を指定する<Year>引数には、4桁の年値を指定します（1990、1997、1998、2012など）。

## <DaylightInfo>引数の値

<DaylightInfo>引数の完全な形式は次の通りです。

```
<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	"月曜日" "火曜日" "水曜日" "木曜日" "金曜日" "土曜日" "日曜日" 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	"1月" "2月" "3月" "4月" ... "11月" "12月"
<SDayPos>	夏時間から標準時間に変更する月内の日の位置	"第1" "第2" "第3" "第4" "最後" "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	"月曜日" "火曜日" "水曜日" "木曜日" "金曜日" "土曜日" "日曜日" 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	"1月" "2月" "3月" "4月" ... "11月" "12月"
<DDayPos>	夏時間から標準時間に変更する月内の日の位置	"第1" "第2" "第3" "第4" "最後" "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。

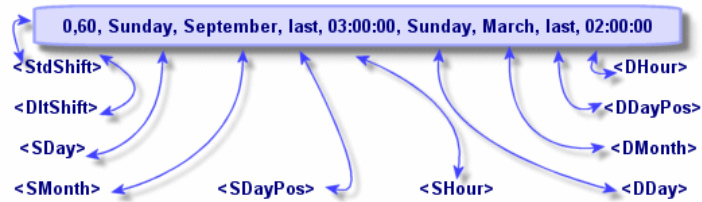
```
2000=0,60,Sunday,October,last,03:00:00,Sunday,March,last,02:00:00
0,60,Sunday,September,last,03:00:00,Sunday,March,last,02:00:00
```

## 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分以内にすると、再接続できるようになります。

## [ データベースサーバのタイムゾーンの検証 ] オプションの特徴

### 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

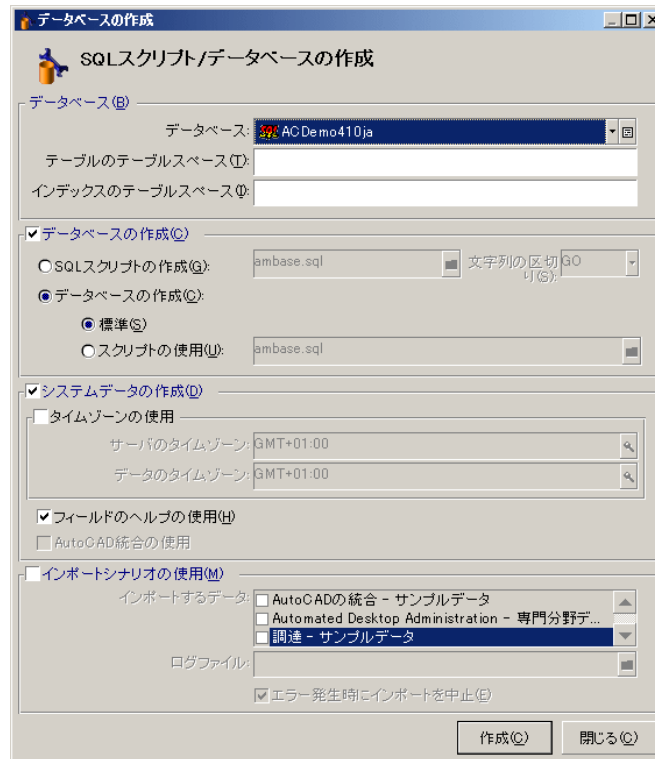
タイムゾーンは、次のような様々な操作に影響を与えます。

- 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AssetCenterでAssetCenter Database Administratorでタイムゾーンオプションを定義できます。AssetCenter Database Administratorで [ アク

シヨン/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。サーバとデータが参照する時間を定義できます。これらの値を変更しても、データベース内の既存の「日付+時間」値には影響しません。

## 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（Windowsのルートディレクトリにある）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 NTまたはWindows 95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。

下の表に、データベースへの接続時に「Am.ini」に加えられる変更を示します。

「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
ある	ある	データベースのタイムゾーン
	ない	変更なし
ない	ある	データベースのタイムゾーン
	ない	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）。

クロックの差 = 母数 (43/30) = 13分

この差が5分（固定値）を超えると、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」（Windowsのルートフォルダにある）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。

```
[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。



# 13 | 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。

次の表に、3つのタイプの主な相違点を示します。

表 13.1. 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
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、『AssetCenterの高度な使い方』、「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: cf\_AveYearlyTco  
 説明: 年間TCO  
 テーブル: 資産 (am Asset)  
 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

Dim IRC As Long
```

## 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があります

 **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。

- Basic言語の場合は「AssetCenterの高度な使い方」、「スクリプト」の章。使用する関数はRetVal()です。
- AQL言語の場合は、「AssetCenterの高度な使い方」、「AQLクエリ」の章。




- 特殊文字列の場合は、『データベース管理』、「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。

テーブルのフィールドおよびリンクのリストで、「**特殊フィールド**」のマークがついたブランチを開きます。このブランチには、使用可能な特殊フィールドのリストが表示されます。**【編集/オプション】**メニュー/**【表示】**項目で指定したフォーマットで、特殊フィールドが表示されます。

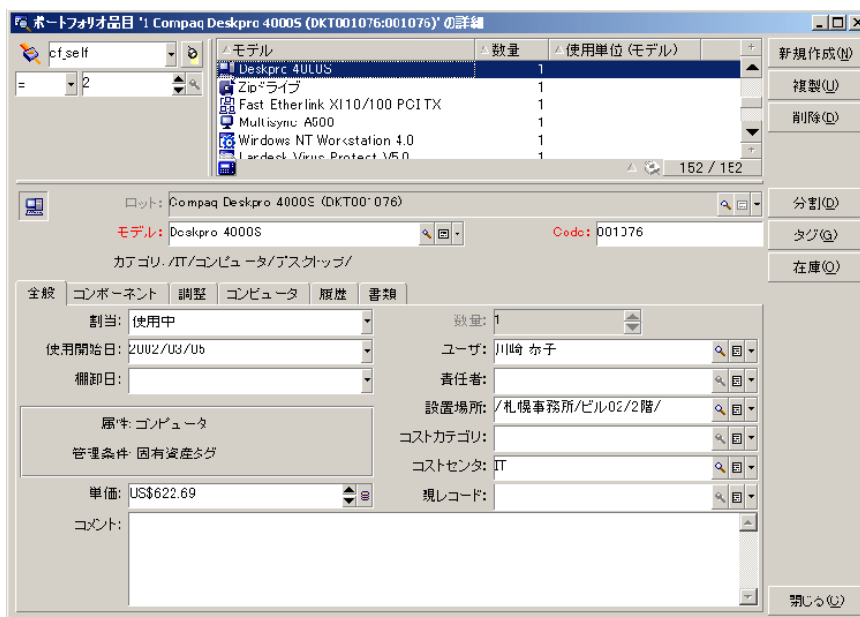
他のフィールドと同じように、リストにこれらの特殊フィールドを追加できます。

## 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」プランチに移動します。AQLタイプのフィールドだけが表示されます。

## 特殊フィールドを参照する

「cf\_」という接頭語を持つSQL名を使って、特殊フィールドを参照できます。下の画面は、特殊フィールドのSQL名をフィルタに使用して、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。



# 14 | 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 14.1. 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립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

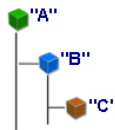
- オブジェクト「A」は、オブジェクト「B」の親オブジェクトになります。
- オブジェクト「B」は、オブジェクト「A」の「子オブジェクト」になります

**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も、これらのフルネームを使って参照できます。必要に応じて、これらの変数を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  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、 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。

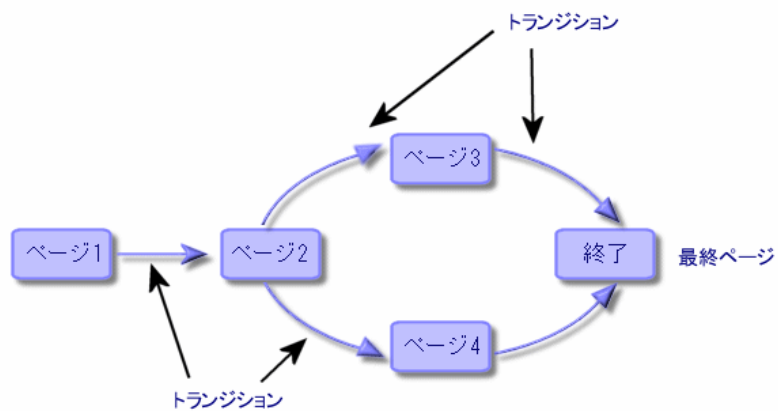
- ウィザードスクリプト（ウィザード）はノードで構成されます。



- 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。
  - TRANSITION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 14.1. 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
  - <プロパティ名>=<テキスト>
- ブール型のプロパティ
  - <プロパティ名>= TRUE
  - <プロパティ名>= FALSE
  - <プロパティ名>(<プロパティ名>= TRUEに相当)
- 数値型のプロパティ
  - <プロパティ名>= 42
- <プロパティ名>= {<Basic変数またはプロパティのフルネーム>}

 注意:

ブール値「TRUE」は、「0」以外の数値と同じです。「FALSE」は数値「0」と同じです

## プロパティを参照する

オブジェクトのプロパティを参照する (プロパティまたはオブジェクトの内容、特に値を参照する) には、次のシンタックスを使います。

```
{<プロパティのフルネーム>}
```

ページ「Page1」のプロパティ「Prop」を参照する場合は、次のように入力します。

```
{Page1.Prop}
```

このシンタックスでは、大文字と小文字は区別されません。

## スクリプトをプロパティの値として定義する

### スクリプトの概念

スクリプトは、グローバル変数RetVal内の値を返す1行または複数行のBasicプログラムです。1行スクリプトの場合、変数は暗黙的です。複数行スクリプトの場合は指定する必要があります。

すべてのBasicスクリプトについて、戻り値のデータ型に注意してください。このデータ型は、スクリプトを介して計算されるプロパティのタイプに依存します。

### 1行スクリプトのシンタックス

```
<プロパティ名>=<スクリプト>
```

例

```
Variable="The name is: " & {Name}
```

上の1行スクリプトは、次の複数行スクリプトと同じです。

```
{ Variable =  
RetVal="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

部署と従業員の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ります。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]
```

次の表に、これら2つの変数の特長を示します。



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

### 警告:

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})
```

## 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  

```

## 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 14.2. 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、この名前でも保存されます。そのため、各ウィザードに異なる名前を付けることお奨めします。
TITLE="<ウィンドウのタイトル>"	ウィザードのウィンドウのタイトルを定義します。 文字列型のプロパティ	TITLE="Move Wizard"	このプロパティにはできるだけ値を定義するようにしてください。
GLOBAL=<スクリプト>	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</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 14.3. 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。 LONG (倍長整数) 型のプロパティ	HEIGHT=5000
MINWIDTH=<MinWidth>	ウィザードのウィンドウの最小幅を定義します。 値はtwipで指定します。	
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。 間隔の値は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。 <del>DEFAULTONNEXT</del> 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 14.4. 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 14.5. 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 14.6. 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。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=<ビットマップファイルのパス>	文字列型のプロパティ		「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 14.7. 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へタイマを関連付けることができます。

## TRANSITION ノードの定義

TRANSITION ノードでは、ウィザード内の2つのページ間の移動を定義します。このノードは、プロパティのブロックだけで構成されます。

 注意:

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>
トランジション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 14.8. 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。

プロパティ名=値	プロパティの説明	例	コメント
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 14.9. 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>	<b>[終了]</b> 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 14.10. 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>	タイト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 14.11. 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 14.12. 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	タイマを実行する時間の間隔を定義します。 間隔はミリ秒単位で指定します。	
TIMER="スクリプト"	タイマの間隔 ("INTERVAL") ごとに実行される内容を定義します。 スクリプトタイプのプロパティ	
VALUE=tickcount	時間の間隔が経過した回数。間隔に依存するプロパティは全て、定期的な間隔で再評価されます。 tickcountの値は自動的に増加します。	



## 倍長整数および文字列ノードの定義

LONG（倍長整数）およびSTRING（文字列）ノードでは、変数を定義します。変数はウィザードのすべてのノードで参照できます。ノード名によって、変数名が決まります。

これらのノードには1つのプロパティしかなく、そのデータ型はノードに依存します。データ型は倍長整数ノードの場合は、倍長整数、文字列ノードの場合は、文字列です。この「VALUE」プロパティを使って、変数の値を定義できます。

表 14.13. 倍長整数および文字列ノードの論理プロパティ

プロパティ名=値	プロパティの説明	例	コメント
VALUE=<値>	ノード名を名前に持つ変数の値を定義します。 倍長整数の場合は倍長整数型、文字列ノードの場合は文字列型	VALUE=12	

### 注意:

倍長整数および文字列ノードは、ウィザードのどのノードでも定義できます。サブノードはありません。

## コントロールノードの定義

ページのコントロールを使ってユーザは情報を指定できます。1ページ内に好きな数だけコントロールを定義できます。ページ内でのコントロールの編成は、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

次の表に、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を示します。

表 14.14. 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"> <li>• コントロールが表示されていない。</li> <li>• コントロールは読取専用である。</li> <li>• コントロールが無効になっている。</li> </ul>	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。&lt;値&gt;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>	<p>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。</p> <p>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。</p> <p>ブール型のプロパティ</p>		
SERIALIZE=<TRUEFALSE>	<p>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せん。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。</p>		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"	<p>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。</p>		

表 14.15. 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	
ENABLED = <TRUE FALSE>	コントロールの値を読み取り専用にして (=TRUE)、ユーザが変更できないようにするか、変更できるようにするか (=FALSE) を定義します。	READONLY=TRUE	
LABEL=" <ラベルのテキスト> "	コントロールの上または左に表示される任意のテキストを定義します。 文字列型のプロパティ	Choice1.Label="ユーザの選択"	このラベルはHTMLをサポートします。

プロパティ名=値	プロパティの説明	例	コメント
LABELLEFT	このプロパティは、コントロールのラベルをコントロールの左側に置きます。 このプロパティを使用する場合は、XOFFSETプロパティに入力する必要があります。 ブール型のプロパティ		
XOFFSET	LABELLEFTプロパティを使用して、コントロールのラベルをコントロール左側に置く場合、ラベル用に確保される場所を定義します。 twip型のプロパティ		
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 14.16. 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 14.17. 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。	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。 <b>例</b> VALUES=A,B,Cと定義すると、VALUES=A1B2C3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 14.18. 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 14.19. 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。 <b>例</b> 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 amEmpDept WHERE Name Like 'A%', ' ', ',', '=') 「VALUES」プロパティを「VALUE」プロパティと混同しないでください。
EDITABLE="<0 1>"	列のテキストが編集可能か (=TRUE)、不可能か (=FALSE) を定義します。 文字列型のプロパティ	EDITABLE="0 1"	
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="amEmpDept"	

プロパティ名=値	プロパティの説明	例	コメント
COLNAME="<名前 名前 名前>"	リストの列のプロパティとタイトルを、フィールドのSQL名と共に定義します。TABLEプロパティには値を入力する必要があります。列のタイトル用に「名前」をSQLで置き換えます。列のタイトルにCOLTITLEプロパティで値が入力されている場合、このプロパティはCOLNAMEプロパティよりも優先されますが、SQLフィールドタイプ(テキスト、日付など)を保持します。	COLNAME=NameNameName	
MULTISEL = <TRUE FALSE>	リスト用に複数選択の使用を定義します。ブール型のプロパティ	MULTISEL=1	

表 14.20. 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。	al bc	
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.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) ID(i)	行 (i) のID値を返します。		

表 14.21. LISTBOX コントロールの必須論理プロパティ

プロパティ名=値	プロパティの説明	例
TABLE=<テーブル名>	列のタイトルの抽出に使うテーブル名。 文字列型のプロパティ	TABLE=amAsset
COLNAME=<タイトル タイトル ...>	TABLEプロパティを使って定義したテーブルのフィールドのSQL名を使って、列のタイトルを定義します。このプロパティを使って、使用する編集コントロールも定義できます。コントロールは、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 14.22. 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 14.23. 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 14.24. 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 14.25. 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 ...>"	データベースの列の幅を、DBLISTBOXコントロール全体に占めるパーセントで定義します。 文字列型のプロパティ	COLWIDTH="40 60"	
LISTHEIGHT = <パーセント>	DBLISTBOXコントロールのサイズを、ウィザード全体の他のDBLISTBOXコントロールを基準にして定義します。 LONG（倍長整数）型のプロパティ	LISTHEIGHT=50	このプロパティの値がそれぞれ「10」と「20」の2つの「DBLISTBOX」コントロールがある場合、2番目のコントロールの高さは最初のコントロールの2倍になります。
TREE=<TRUE FALSE>	データをツリーで表示（=TRUE）するか、表示しない（=FALSE）かを指定します。 ブール型のプロパティ	TREE=TRUE	このプロパティはデフォルトでは「FALSE」に設定されます。



プロパティ名=値	プロパティの説明	例	コメント
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	
MULTISEL = <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 = "<Condition>"	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' "	
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)	

プロパティ名=値	プロパティの説明	例	コメント
[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値を返しません。	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"><li>• 1 - システム名</li><li>• 2 - SQL名</li><li>• 3 - デフォルトのタイトル</li><li>• 4 - 説明 (ヘルプ用テキスト)</li></ul> 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 14.26. 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 amEmpIDept 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 = "姓  名前"	
COLWIDTH="<幅 幅 ...>"	リストの列のサイズを、コントロール全体のサイズに占めるパーセントで定義します。 文字列型のプロパティ	COLWIDTH = "50 50"	
LISTHEIGHT=<パーセント>	DBQUERYBOXコントロールのサイズを、ウィザード全体の他のDBQUERYBOXコントロールを基準にして定義します。 LONG (倍長整数) 型のプロパティ	LISTHEIGHT=50	このプロパティの値がそれぞれ「10」と「20」の2つのDBQUERYBOXコントロールがある場合、2番目のコントロールの高さは最初のコントロールの高さの2倍になります。


プロパティ名=値	プロパティの説明	例	コメント
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。 ブール型のプロパティ	DBLCLICK=FALSE	

プロパティ名=値	プロパティの説明 例	コメント
[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 14.27. 標準モードのDBEDITコントロールの物理プロパティ

プロパティ名=値	プロパティの説明	例
TABLE=<テーブルのSQL名>	開始テーブルのSQL名 文字列型のプロパティ	TABLE=amAsset
FIELD=<フィールドのSQL名>	コントロールに使用されるフィールドのAQL名。 文字列型のプロパティ	FIELD="seAcquMethod"

プロパティ名=値	プロパティの説明	例
TABLE.LABEL([iNameType])	<p>このメソッドは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"> <li>• 1 - システム名</li> <li>• 2 - SQL名</li> <li>• 3 - デフォルトのタイトル</li> <li>• 4 - 説明 ( ヘルプ用テキスト )</li> </ul> <p>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。</p>	本章の「DBQUERYBOXコントロールの物理プロパティ」の節を参照してください。
FIELD.LABEL([iNameType])	<p>このメソッドはフィールド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"> <li>• 1 - システム名</li> <li>• 2 - SQL名</li> <li>• 3 - デフォルトのタイトル</li> <li>• 4 - 説明 ( ヘルプ用テキスト )</li> </ul> <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>	

## 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に入力するためのコントロールを作成します。

## プロパティ

このコントロールに固有のプロパティはありません。

## DBPathコントロール

DBPATHは、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 14.28. 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 14.29. 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。

プロパティ名=値	プロパティの説明	例/コメント
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。	
TABLE.LABEL([iNameType])	このメソッドは、リンクのソーステーブルのタイトルを返します。 タイトルタイプ (iNameType) は以下の通りです。 <ul style="list-style-type: none"> <li>• 1- システム名</li> <li>• 2- SQL名</li> <li>• 3- デフォルトのタイトル</li> <li>• 4- 説明 (ヘルプ用テキスト)</li> </ul> TABLEプロパティには値が入力されなければなりません。	本章の「DBQUERYBOXコントロールの物理プロパティ」の節を参照してください。



プロパティ名=値	プロパティの説明	例/コメント
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"> <li>• 1 - システム名</li> <li>• 2 - SQL名</li> <li>• 3 - デフォルトのタイトル</li> <li>• 4 - 説明 ( ヘルプ用テキスト )</li> </ul> <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 14.30. 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 14.31. CHARTコントロールの論理プロパティ

プロパティ名= 値	プロパティの説明	例
SERIES	縦線（パイプ文字）で区切られた系列のリストを定義します。 このリストは空であってはなりません。このリストなしではグラフは表示されません。	SERIES="購入価格 販売価格"
VALUES	グラフの系列の数値を定義します。 2次元の図表	VALUES="1 2,1 4"
FORMAT	データ型を定義します。 <ul style="list-style-type: none"> <li>• 倍長整数（long）</li> <li>• 倍精度浮動小数点型（double）</li> <li>• 数値（number）</li> <li>• パーセント（percent）</li> </ul>	
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 14.32. 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	<p>図表のタイプを定義します。</p> <ul style="list-style-type: none"> <li>MODE=0 : 縦の棒グラフ</li> <li>MODE=1 : 横の棒グラフ</li> <li>MODE=3 : 円グラフ</li> </ul>	
LABELS	列のタイトルを定義します。	1月 2月
3D	<p>グラフを3Dにするか (=TRUE) しないか (=FALSE) を定義します。</p>	
COLORS	<p>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。</p> <p>RGB値は十進法です。</p>	255 16777215 16711680 赤、白、青色を表示します。
INTERACTIVE	<p>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。</p>	
POPUP	<p>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。</p>	
BACKGROUND	<p>グラフに背景を表示するか (=TRUE) しないか (=FALSE) を定義します。</p>	
BACKIMAGE	<p>グラフの背景になる画像のパスを定義します。</p> <p>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。</p>	
STACKED	<p>積み上げ棒グラフにするか (=TRUE) しないか (=FALSE) を定義します。</p>	
CHARTHEIGHT	<p>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。</p>	
CAPTION	<p>タイトルを表示します。</p>	

プロパティ名=値	プロパティの説明	例
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 14.33. 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"> <li>• OPENMODE=1：ファイルを開きます。</li> <li>• OPENMODE=2：ファイルを保存します。</li> <li>• OPENMODE=4：フォルダを開きます。</li> <li>• OPENMODE=8：フォルダを保存します。</li> </ul>	
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 14.34. 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	ユーザがスケジューラ内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 14.35. 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"> <li>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倍長整数 ( long )</li> <li>フォーマットされていない整数</li> <li>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倍精度浮動小数点型 ( double )</li> </ul>	

## 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 14.36. 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 14.37. 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"> <li>日付 (date)</li> <li>時刻 (time)</li> <li>日付 + 時刻</li> </ul> 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

## 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。
- 資産のテーブルのレコードを選択して、移動する資産を直接選択する。
- 設置場所から移動する資産を特定する。まず、設置場所を選択し、次にこの設置場所から移動する資産を選択します。

---

#### 注意:

ユーザが、移動する資産の選択方法をこれらの中から選択できるよう、ユーザ選択ページを作成する必要があります。

---

### 新しい設置場所を選択する

資産の新しい設置場所を選択するには、設置場所のテーブルからレコードを選択するだけです。

## 手順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

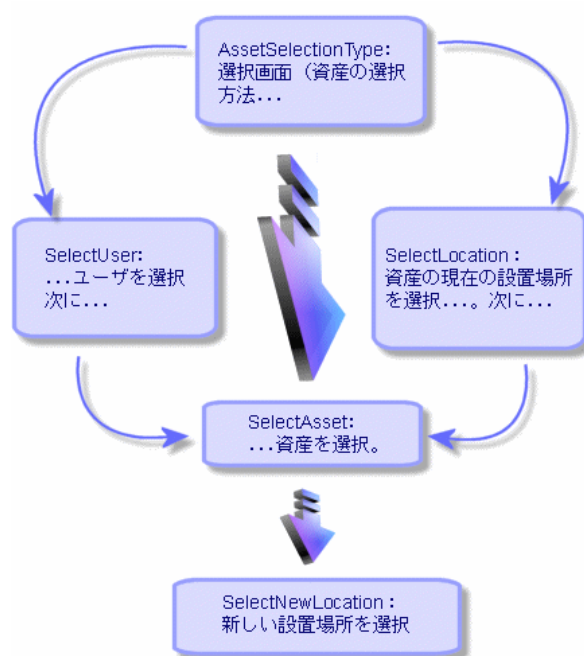
### 3 各ページの内容

#### 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 14.2. 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	ユーザが資産の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、部署と従業員の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、設置場所の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、設置場所の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、amEmplDeptから選択

```

```

{ PAGE pgUser
TITLE = "移動する従業員の選択"
{ DBLISTBOX Users
COLNAME = "Name|FirstName"
COLWIDTH = "50|50"
DBLCLICK = 1
LABEL = "移動する従業員"
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
WHERE 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 amEmpDept WHERE
  FullName LIKE LikeParam(amEmpDept_2:FullName)+'%' AND
  amEmpDept_2:EmpDeptId IN(" & {pgUser.Users} & ")","|",",", "=",)
  }
}

;最後のページ

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

    dim hRecord as Long

    dim iEmpCount as Integer
    iEmpCount = {pgRecap.Users.VALUES.Count()}
  }
}

```

```

dim iMax as Long
iMax = iEmpICount

dim ILocald as long
ILocald = {pgNewLoc.NewLocId}

IErr = amStartTransaction()

dim i as Integer
For i = 1 To iEmpICount
  IErr = AmProgress((100 * i) / iMax)
  IErr = AmLog("Deplacement de la personne " + {pgRecap.Users.VALUES(i,1)})
  hRecord = AmGetRecordFromMainId("amEmpIDept", {pgRecap.Users.VALUES(i,0)})
)
  If hRecord <> 0 then
    IErr = AmSetFieldLongValue( hRecord, "ILocald", ILocald)
    IErr = AmUpdateRecord(hRecord)
    IErr = AmReleaseHandle(hRecord)
  End If
Next i

IErr = 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
}

```

## グラフィカルエディタの使用方法

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必要があります。

- 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 14.3. 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PAGEノードを作成するには、最初にROOTノードをクリックします。親ノードを選択したら、マウスの右ボタンでクリックしてポップアップメニューを表示します。**【新規作成】**メニューを選択すると、作成できるノードがグループごとに表示されます。

ウィザードのツリーにノードが挿入されます。

## ノードのプロパティを編集する

ノードを作成したら、ノードのプロパティに値を割り当てることができます。この作業は、エディタの右側で行います。

プロパティの値は、次の2つの方法で定義できます。

- 固定値を入力する。
- スクリプトを定義する。

### 注意:

スクリプトは常に、固定値より優先されます。プロパティにスクリプトと固定値の両方を割り当てると、固定値は無視され、スクリプトが解釈されます。

## プロパティに固定値を割り当てる

対象のプロパティの**【値】**列を直接クリックします。プロパティで使用可能なデータ型（テキスト、ブール、倍精度数値など）に従って、リストから値を選択するか、テキスト編集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。

```
{IbxMyListBox.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。

```
{IbxMyListBox.Values.Count()}
```



## 質問

次のコードが機能しません。

```
{IbxMyListBox.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。

```
{IbxMyListBox.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。下に、ウィザードから資産テーブルを開く方法を示します。

```
Dim irc as Long
irc = AmActionDDE("aam", "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せん。

- 「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。

# 15 | 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/ニュースマークーの表示]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、ニュースマーカーを表示できます。

## ニュースの重要度

ニュースメッセージの重要度を定義するには、ニュースの詳細画面にある**【重要度】**（SQL名：seSeverity）フィールドで指定します。

各重要度には**【色】**（SQL名：IColor）フィールドで選択された色が関連付けられます。ニュースマーカーには、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 15.1. 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、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。





