



Peregrine | Connect-It
コネクタ



© 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Connect-It®は、Peregrine Systems, Inc.の商標です。

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。

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

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

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

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

本書の内容は、ライセンス契約に基づくプログラムのバージョン3.2.0に適用されます。

Connect-It

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



目次

はじめに	11
コネクタの使用目的	11
コネクタの対象ユーザ	11
本書の使用方法	12
1. コネクタの設定	13
コネクタを設定する	14
コネクタに名前を付け、役割を説明する	14
接続タイプを選択する	15
接続を選択する	16
高度な設定	24
再接続のパラメータを設定する	27
サーバとの時間差を指定する	29
スケジュールのポイントを設定する	32
トランザクションを設定する	34
キャッシュを設定する	35
ピボットドキュメントタイプの使用	36
2. コネクタのルール (ディレクティブ)	39
生成用ルール	40
取り込み用ルール	43

整合性	44
3. Peregrine Systemsコネクタ	53
Action Request Systemコネクタ	53
Asset Managementコネクタ	60
InfraTools Desktop Discoveryコネクタ	67
InfraTools Managementコネクタ	78
InfraTools Network Discoveryコネクタ	82
Peregrine Desktop Inventory (PDI) コネクタ	88
ゲートウェイ3.xコネクタ	92
ServiceCenterコネクタ	98
4. アプリケーションコネクタ	115
Lotus Notesコネクタ	115
MQSeriesコネクタ	123
NTセキュリティコネクタ	147
5. プロトコルコネクタ	151
データベースコネクタ	151
Eメールコネクタ	158
LDAPコネクタ	171
コマンドラインコネクタ	184
テキストコネクタ	187
XMLコネクタ	208
6. インベントリコネクタ	231
Intel LANDeskコネクタ	231
Tivoli Inventoryコネクタ (バージョン3.1と3.6)	235
Tivoli Inventoryコネクタ (バージョン4.0)	241
SMS 1.xコネクタとSMS 2.xコネクタ	245
Unicenter AMOコネクタ	250
Tivoli CM Inventory 4.2 コネクタ	253
7. Javaコネクタ	261
SCAutoリスニングコネクタ	261
XMLリスニングコネクタ	266
Tivoli Enterprise Consoleコネクタ (送信)	269
Tivoli Enterprise Consoleコネクタ (受信)	273
Webサービスコネクタ	279
JDBCコネクタ	284

8. ERPコネクタ	287
SAPコネクタ	287
SAP BAPIコネクタ	289
SAP IDocコネクタ	294
9. 付属シナリオ	301
Unicenter AMOシナリオ	302
InfraTools Desktop Discoveryシナリオ	302
Intel LANDesk シナリオ	304
InfraTools Managementシナリオ	305
InfraTools Network Discoveryシナリオ	305
Action Request Systemシナリオ	307
LDAPシナリオ	308
Eメールシナリオ	309
MQSeriesシナリオ	311
NTセキュリティシナリオ	313
ServiceCenterシナリオ	314
SMSシナリオ	324
Tivoli Inventory Managementシナリオ	326
ゲートウェイ3.xシナリオ	328
Webサービスシナリオ	328
Peregrine Desktop Inventoryシナリオ	328
A. AQLクエリ	331
はじめに	331
AQLクエリの記述に関する推奨事項	334
並べ替えとインデックス	343
クエリエディタ	345
AQLのシンタックス	350
AQLの関数	363
クエリの例	367
索引	373

図の一覧表

2.1. コネクタ - 生成用ルールと取り込み用ルール	40
2.2. 取り込み用ルール - [整合性] タブ	44
2.3. 取り込み用ルール - [高度な整合性チェック] タブ	50
2.4. 整合性チェック - コレクションのフィルタ	52
2.5. 整合性チェック - ターゲットコレクションの構成要素の削除	52
3.1. PDIコネクタ - 生成用ドキュメントタイプ	92
4.1. MQSeriesコネクタ - 相関IDによるMQSeriesメッセージの識別	144
4.2. NTセキュリティコネクタ - 使用可能なドキュメントタイプ	149
5.1. Eメールコネクタ - 拡張されていないInMailMessageドキュメントタイプ	168
5.2. Eメールコネクタ - DTDにより拡張されたInMailMessageドキュメントタイプ	168
5.3. テキストコネクタ - テキストファイル内のデータの記録法	188
5.4. テキストコネクタ - データ処理	202
5.5. テキストコネクタ - 取り込み用ルール	208
5.6. XMLコネクタ - の取り込み用ルール	227
6.1. Intel LANDeskコネクタ - COMPUTER_SYSTEMドキュメントタイプ	235
6.2. Tivoli Inventory Managementコネクタ - COMPUTER_SYSTEMドキュメントタイプ	239
6.3. Tivoli Inventory Managementコネクタ - 生成用ルール	240
6.4. SMSコネクタ - 生成用ルールの例	249

6.5. Tivoli CM - Inventory 4.2コネクタ - COMPUTER_SYSTEMドキュメントタイプ	257
6.6. Tivoli CM Inventory 4.2コネクタ - 生成用ルール	258
7.1. Webサービスコネクタ - 機能の概要	280
8.1. SAPコネクタ - 取り込み - 生成用ドキュメントタイプ	289
A.1. クエリエディタの活用法	346

表の一覧表

1.1. インベントリコネクタとデータベースコネクタ - 設定の詳細オプション	25
3.1. Action Request Systemコネクタの設定	55
3.2. Asset Managementコネクタの設定	62
3.3. InfraTools Desktop Discoveryコネクタの設定	69
3.4. InfraTools Managementコネクタの設定	78
3.5. InfraTools Network Discoveryコネクタの設定	83
3.6. PDIコネクタの設定	88
3.7. ゲートウェイコネクタの設定	96
3.8. ServiceCenterコネクタの設定	100
3.9. Asset Managementコネクタの処理レポートと、ServiceCenterコネクタのConnectItDelドキュメントタイプ間のマッピング	105
4.1. Lotus Notesコネクタの設定	116
4.2. MQSeriesコネクタ（読み取り）の設定	124
4.3. MQSeriesコネクタ（書き込み）の設定	134
4.4. NTセキュリティコネクタの設定	147
4.5. NTセキュリティコネクタ - 生成用ルールの例	150
5.1. データベースコネクタの設定	154
5.2. Eメール（受信）コネクタの設定	160
5.3. Eメール（送信）コネクタの設定	163
5.4. Eメール（受信）コネクタの生成用ドキュメントタイプ	166
5.5. Eメール（送信）コネクタの取り込み用ドキュメントタイプ	168

5.6. LDAPコネクタの設定	172
5.7. LDAPフィルタ	179
5.8. マップテーブルファイル (ASCII値の文字)	180
5.9. コマンドラインコネクタの設定	184
5.10. テキストコネクタの設定 (読み取り)	189
5.11. テキストコネクタの設定 (書き込み)	195
5.12. XMLコネクタの設定 (読み取り)	209
5.13. XMLコネクタの設定 (書き込み)	219
6.1. Intel LANDeskコネクタの設定	232
6.2. Tivoli Inventory Managementコネクタの設定	237
6.3. Tivoli Inventory Management 4.0コネクタの設定	243
6.4. SMS 1.xコネクタとSMS 2.xコネクタの設定	247
6.5. Unicenter AMOコネクタの設定	251
6.6. Tivoli Inventory Management 4.0コネクタの設定	255
7.1. SCAutoリスニングコネクタの設定	263
7.2. XMLリスニングコネクタの設定	268
7.3. TEC adapterコネクタの設定	270
7.4. TECコネクタの設定	274
7.5. サポートされるプロトコル - Webサービスコネクタ	281
7.6. Webサービスコネクタの設定	281
7.7. 拡張型処理レポートの名前 - Webサービスコネクタ	283
7.8. JDBCコネクタの設定	284
8.1. ABAPデータ型とConnect-Itデータ型の対応関係	290
8.2. SAP BAPIコネクタの設定	291
8.3. SAP IDocコネクタの設定	295
A.1. 表記法	350
A.2. 論理演算子	354
A.3. 比較演算子	355
A.4. 集合型のAQL関数	363
A.5. 文字列型のAQL関数	364
A.6. 日付型のAQL関数	364
A.7. 日付型のAQL関数	365
A.8. 数値型のAQL関数	366
A.9. テスト型のAQL関数	367

はじめに

コネクタの使用目的

コネクタは、外部アプリケーションと通信してデータの交換を実行します。

コネクタは3種類に分類されます。

- ベースコネクタ
これらのコネクタはシナリオビルダの全ユーザが使用できます。
- オプションコネクタ
これらのコネクタはシナリオビルダに付属していますが、ライセンスに許可されている場合のみ使用可能になります。
- 追加コネクタ
これらのコネクタはシナリオビルダには付属していません。ペレグリンシステムズにお問い合わせの上、入手してください。

コネクタの対象ユーザ

コネクタの対象ユーザは、企業内で使用する多種アプリケーションの統合を担当する従業員です。Java開発キットを使用すると、独自のコネクタも開発できます。

Java開発キットの説明については、マニュアル『ユーザガイド』の「Connect-ItのJava開発キット」の章を参照してください。

本書の使用方法

コネクタの章

以下の章

- 「Peregrine Systemsコネクタ [p. 53]」
- 「アプリケーションコネクタ [p. 115]」
- 「プロトコルコネクタ [p. 151]」
- 「インベントリコネクタ [p. 231]」
- 「Javaコネクタ [p. 261]」

では、Connect-Itで使用可能なコネクタが説明されています。各コネクタごとに以下の情報が提供されています。

- 互換性
- 制約点
- 設定
- ルール（ディレクティブ）
- 追加情報

「付属シナリオ」の章

この章では、シナリオビルダの付属シナリオが説明されています。

付属シナリオ（SCNファイル）は、Connect-Itインストール先フォルダの「scenario」サブフォルダ内にあります。例：「fc/fcac41/fcac.scn」ファイルは、コンピュータの「[Connect-Itのインストール先フォルダ]scenario/fc/fcac41/fcac.scn」に位置します。

1 | コネクタの設定

コネクタのインスタンスをシナリオ内で設定すると、以下の操作を実行できます。

- コネクタに名前を付け、役割を説明する
- コネクタが外部アプリケーション（データベース、メッセージシステム、キューマネージャなど）と通信するための接続パラメータを指定する
- 使用可能なドキュメントタイプや、生成用または取り込み用のドキュメントタイプを処理するための複数のオプションを指定する


本章では、コネクタの設定ウィザードへのアクセス方法を説明した後、複数のコネクタに共通な設定ウィザードのページについて説明します。「[Peregrine Systemsコネクタ \[p. 53\]](#)」、「[アプリケーションコネクタ \[p. 115\]](#)」、「[プロトコルコネクタ \[p. 151\]](#)」、「[インベントリコネクタ \[p. 231\]](#)」と「[Javaコネクタ \[p. 261\]](#)」の章では、コネクタの設定に関する節は、本章の説明を参照しています（例: 「[Peregrine Systemsコネクタ \[p. 53\]](#)」章の「[Asset Managementコネクタ \[p. 60\]](#)」節の「[Asset Managementコネクタの設定 \[p. 62\]](#)」）。

一部のコネクタの設定では、特定のオプションが共通な設定ウィザードページに含まれることがあります。これらのオプションは、各コネクタの節で説明されています。

コネクタを設定する

Connect-Itコネクタの設定はウィザードで実行できます。

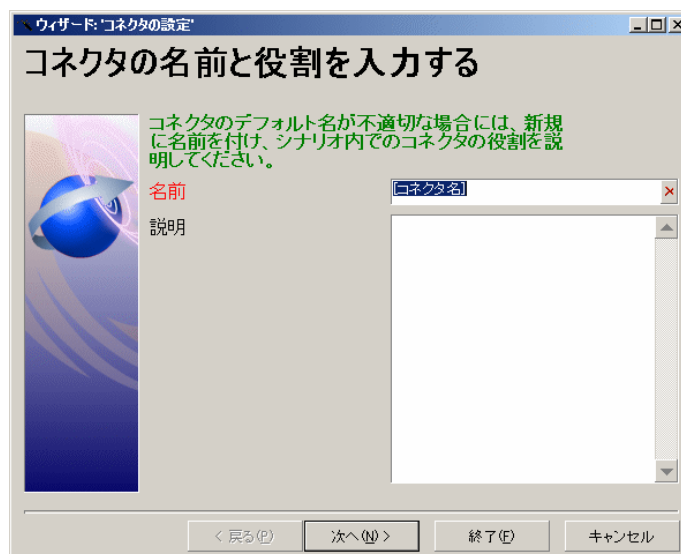
コネクタの設定ウィザードを起動するには、以下の方法があります。

- **コネクタがシナリオ図内に位置しない場合**
 - 1 ツールボックス内のコネクタをダブルクリックします。
 - 2 ツールボックスからシナリオ図へコネクタをドラッグします。
- **コネクタがシナリオ図内に位置する場合**
 - 1 コネクタを選択してから [ツール / 設定] を選択します。
 - 2 コネクタを選択し [F2] キーを押します。
 - 3 コネクタを選択し、右クリックしてショートカットメニューから [コネクタを設定する] を選択します。
 - 4  をクリックします。

コネクタに名前を付け、役割を説明する

このページでは以下の内容を入力します。

- コネクタの名前
- コネクタの役割の説明



[名前]

このフィールドではコネクタに名前を付けます。フィールドのデフォルト値は、ツールボックス内のコネクタ名です。同種のコネクタがシナリオ図内に既にある場合は、コネクタのデフォルト名は「コネクタ名」、「コネクタ名2」、「コネクタ名3」の様になります。

この名前はシナリオ内でコネクタを識別するために使用され、シナリオ図内のコネクタ上に表示されます。

警告:

ある同一コネクタの、異なった複数のインスタンスがシナリオ内にある場合、それぞれに別の名前を付ける必要があります。例：2つのAssetCenterデータベース間のマイグレーションシナリオ内での、Asset Managementコネクタの2つのインスタンス。

[説明]

コネクタを説明するテキストを入力します（例：データソースAの読み取り、データソースAへの書き込み、など）。

これは必須フィールドではありません。

接続タイプを選択する

このページ目では、データベース系アプリケーションへの接続タイプを選択します。以下の選択肢があります。

- ODBCタイプの接続
この場合ODBCレイヤが使用されます。
- Oracleネイティブ接続
- Sybaseネイティブ接続
- MySQLネイティブ接続
- DB2ネイティブ接続



 **重要項目:**

ODBC接続よりもネイティブ接続の使用をお勧めます。コネクタがデータベースを自動記述する際に、ネイティブ接続の方が高性能を発揮します。

 **注意:**

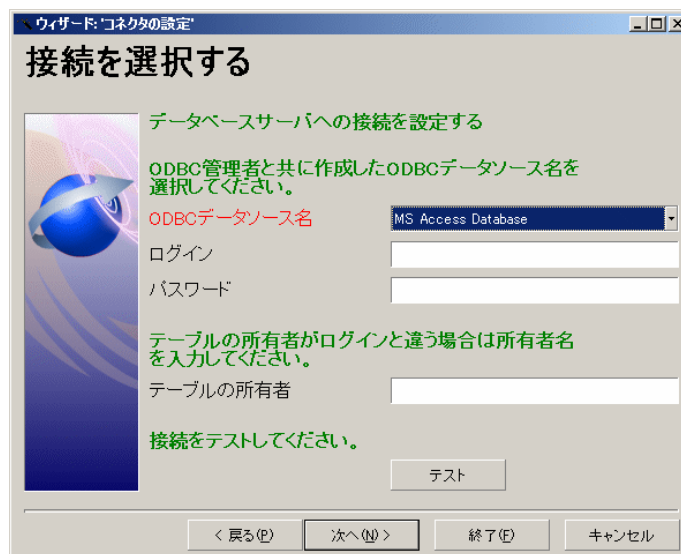
MySQLネイティブ接続を使用するには、MySQLのクライアントのダイナミックライブラリ (DLL) を、Connect-Itインストール先フォルダの「bin」フォルダ内 (またはオペレーティングシステムの「system32」フォルダ内) にコピーする必要があります。

接続を選択する

このページでは接続を選択します。

前のページ [[接続タイプを選択する](#) [p. 15]] で選択された接続タイプに応じて、このページの内容は変化します。

接続を設定する（ODBC接続）



このページでは以下の内容を指定します。

- ODBCデータソース名
- ログイン
- パスワード
- テーブルの所有者

[ODBCデータソース名]

ODBC接続名を指定します。ドロップダウンリストには、自分のコンピュータで使用できるODBC接続が列挙されています。このリストは編集不可能なため、使用するODBCデータソースが既にこのリストに含まれている必要があります。

[ログイン]

ODBCデータソースへのアクセスを許可するログインを入力します。

[パスワード]

ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

【テーブルの所有者】

ログインがテーブルの所有者と違う場合は、テーブルの所有者名を入力します。

【テスト】

このボタンを押すと接続をテストできます。

接続をテストするには、

- 接続用パラメータを入力します。
- 【テスト】をクリックします。

【接続のテスト】ウィンドウが開き、接続の成功または失敗を表示します。接続に失敗すると、失敗の原因が表示されます。

- 【閉じる】をクリックして設定ウィザードへ戻ります。

接続を選択する（Oracleネイティブ接続）

ウィザード: コネクタの設定

接続を選択する

データベースサーバへの接続を設定する

データベースサーバ

ログイン

パスワード

テーブルの所有者がログインと違う場合は所有者名を入力してください。

テーブルの所有者

接続をテストしてください。

< 戻る(B) 次へ(N) > 終了(F) キャンセル

このページでは以下の内容を指定します。

- データベースサーバ
- ログイン
- パスワード
- テーブルの所有者

[データベースサーバ]

Oracleサーバの名前を指定します。

[ログイン]

データベースへのアクセスを許可するOracleのユーザ名を入力します。

[パスワード]

ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

[テーブルの所有者]

ログインがテーブルの所有者と違う場合は、テーブルの所有者名を入力します。

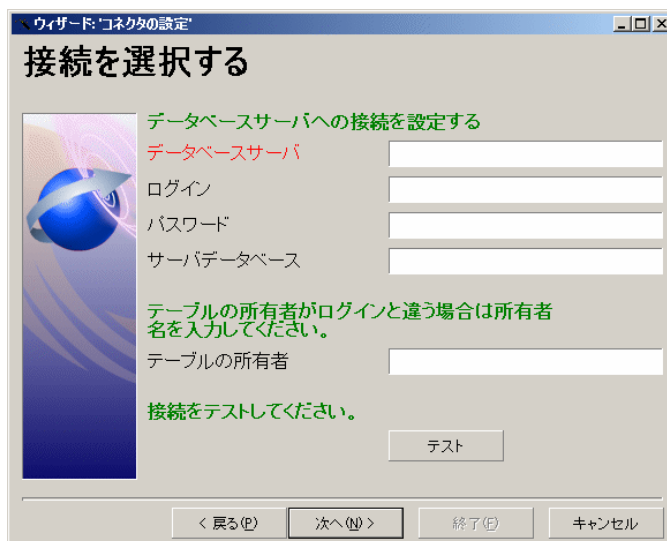
[テスト]

このボタンを押すと接続をテストできます。

接続をテストするには、

- 接続用パラメータを入力します。
- [テスト] をクリックします。
[接続のテスト] ウィンドウが開き、接続の成功または失敗を表示します。
接続に失敗すると、失敗の原因が表示されます。
- [閉じる] をクリックして設定ウィザードへ戻ります。

接続を選択する - Sybaseネイティブ接続



ウィザード: 'コネクタの設定'

接続を選択する

データベースサーバへの接続を設定する

データベースサーバ

ログイン

パスワード

サーバデータベース

テーブルの所有者がログインと違う場合は所有者名を入力してください。

テーブルの所有者

接続をテストしてください。

テスト

< 戻る(B) 次へ(N) > 終了(F) キャンセル

このページでは以下の内容を指定します。

- データベースサーバ
- ログイン
- パスワード
- サーバデータベース
- テーブルの所有者

[データベースサーバ]

Sybaseデータベースのサーバの名前を指定します。

[ログイン]

データベースへのアクセスを許可するSybaseのユーザ名を入力します。

[パスワード]

ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

[サーバデータベース]

接続先のデータベースの名前を入力します。

[テーブルの所有者]

ログインがテーブルの所有者と違う場合は、テーブルの所有者名を入力します。

[テスト]

このボタンを押すと接続をテストできます。

接続をテストするには、

- 接続用パラメータを入力します。
- [テスト]をクリックします。

[接続のテスト]ウィンドウが開き、接続の成功または失敗を表示します。
接続に失敗すると、失敗の原因が表示されます。

- [閉じる]をクリックして設定ウィザードへ戻ります。

接続を選択する - ネイティブMySQL

このページでは以下の内容を指定します。

- データベースサーバ

- ログイン
- パスワード
- サーバデータベース

【データベースサーバ】

MySQLデータベースのサーバの名前を指定します。

【ログイン】

データベースへのアクセスを許可するユーザ名を入力します。

【パスワード】

ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

【サーバデータベース】

接続先のデータベースの名前を入力します。

【テスト】

このボタンを押すと接続をテストできます。

接続をテストするには、

- 接続用パラメータを入力します。
- **【テスト】**をクリックします。

[接続のテスト] ウィンドウが開き、接続の成功または失敗を表示します。接続に失敗すると、失敗の原因が表示されます。

- **【閉じる】**をクリックして設定ウィザードへ戻ります。

接続を選択する - DB2接続

ウィザード: コネクタの設定

接続を選択する

データベースサーバへの接続を設定する

データベースサーバ

ログイン

パスワード

テーブルの所有者がログインと違う場合は所有者名を入力してください。

テーブルの所有者

接続のテスト

テスト

< 戻る(B) 次へ(N) > 終了(F) キャンセル

このページでは以下の内容を指定します。

- データベースサーバ
- ログイン
- パスワード
- サーバデータベース
- テーブルの所有者

[データベースサーバ]

DB2データベースのサーバの名前を指定します。

[ログイン]

データベースへのアクセスを許可するDB2のユーザ名を入力します。

[パスワード]

ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

[テーブルの所有者]

ログインがテーブルの所有者と違う場合は、テーブルの所有者名を入力します。

[テスト]

このボタンを押すと接続をテストできます。

接続をテストするには、

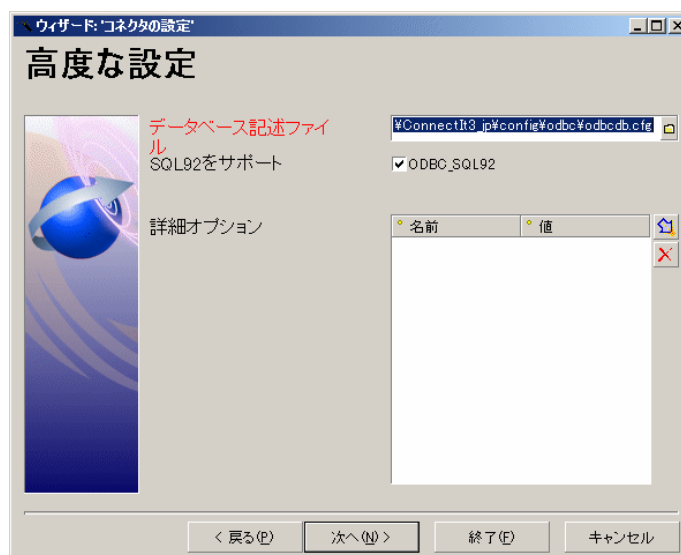
- 接続用パラメータを入力します。
- [テスト]をクリックします。

[接続のテスト]ウィンドウが開き、接続の成功または失敗を表示します。
接続に失敗すると、失敗の原因が表示されます。

- [閉じる]をクリックして設定ウィザードへ戻ります。

高度な設定

このページでは、高度な設定のパラメータを定義できます。



データベース記述ファイル

このファイルには以下の内容が含まれています。

- リレーショナル（関係型）データベース内のテーブル間の結合の定義
- スケジュールのポインタとして使用されるフィールド

このファイルは各インベントリコネクタ用にカスタマイズされます。データベースコネクタでは、このファイルは空のシェルに相当し、ここでは結合もポインタ用のフィールドも定義されていません。

ベレグリンシステムズの顧客サポートから別のファイルを受け取っていない限り、ファイルを変更することはお控えください。


SQL92をサポート

このオプションはデフォルトで選択されています。コネクタがデータベースに送信するSQLクエリが、SQL92シンタックスに従っていることを意味します。データベースがこのシンタックスをサポートしない場合は（例：Informix）、このオプションをオフにする必要があります。


詳細オプション

この編集用ゾーンには、コネクタ専用の高度なオプションを入力できます。

詳細オプションを入力するには、

- 1  をクリックします。
- 2 **【名前】**の列にオプションの名前を入力します。
- 3 **【値】**の列にオプションの値を入力します。

オプションの名前や値を変更するには、名前や値を直接ダブルクリックしてから新規の値を入力します。

オプションを削除するには、オプションに対応する行を選択してから  をクリックします。

インベントリコネクタとデータベースコネクタの詳細オプション

インベントリコネクタとデータベースコネクタで入力できる詳細オプションのリストは、次の表の通りです。

警告:

ベレグリンシステムズの顧客サポートから変更するように指示されていない場合や、マニュアルで変更方法が説明されていない場合は、詳細オプションのデフォルト値を変更することはお控えください。

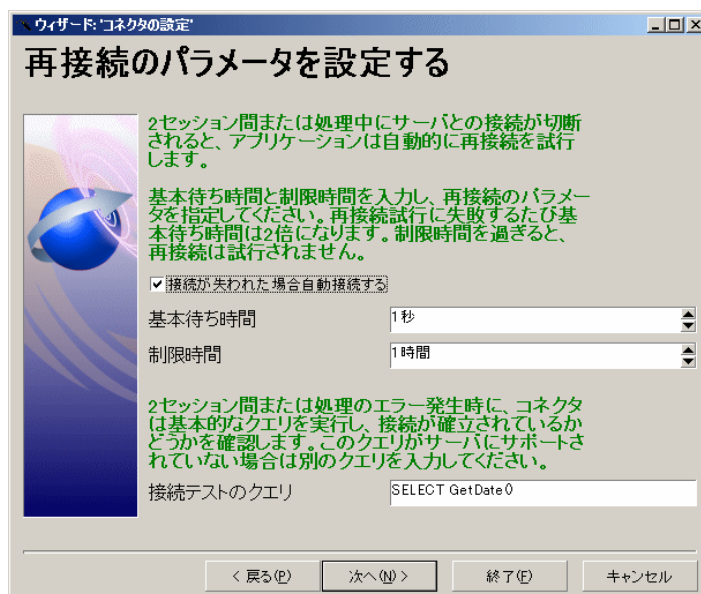
表 1.1. インベントリコネクタとデータベースコネクタ - 設定の詳細オプション

名前	デフォルト値	説明	データベース	例
StmtCache	30	キャッシュのサイズ	<ul style="list-style-type: none"> long整数 インスタンス数 	以下の場合0を入力する <ul style="list-style-type: none"> DB2 Microsoft Visual Studio
ConstAsText	0	テキストフォーマットを使用する定数のフォーマット	ブール <ul style="list-style-type: none"> 0 = False 1 = True 	以下の場合1を入力する <ul style="list-style-type: none"> Pilote ODBC Novell
NoPrepare	0	ダイナミックSQLを規制する	ブール <ul style="list-style-type: none"> 0 = False 1 = True 	以下の場合1を入力する <ul style="list-style-type: none"> Pilote ODBC Novell
AutoCommit	1	コマンドcommitの自動モード(1)または手動モード(0)を指定する	ブール <ul style="list-style-type: none"> 0 = False 1 = True 	以下の場合0を入力する <ul style="list-style-type: none"> Pilote ODBC Novell
RowSetSize	1	返す行の数を指定する	ブール <ul style="list-style-type: none"> 0 = False 1 = True 	以下の場合0を入力する <ul style="list-style-type: none"> Pilote ODBC Novell
ExtendedFetch	1	拡張取り出しの使用	ブール <ul style="list-style-type: none"> 0 = False 1 = True 	以下の場合0を入力する <ul style="list-style-type: none"> Pilote ODBC Novell
AddIndexDesc	1	インデックスの自動記述を許可する	ブール <ul style="list-style-type: none"> 0 = False 1 = True 	以下の場合0を入力する <ul style="list-style-type: none"> Pilote ODBC Novell
FetchingArraySize	データベースエンジンに応じる	リンクする行の数を指定できるようにする	long整数	以下の場合1を入力する <ul style="list-style-type: none"> Pilote ODBC Novell

名前	デフォルト値	説明	データベース	例
OdbcSelectCursorType	DB2とOracleでは0 その他のエンジンでは1	カーソルのタイプを指定できるようにする	long整数 <ul style="list-style-type: none"> 0 FORWARD_ONLY 1 KEYSET_DRIVEN 2 KEYSET_DRIVEN 3 STATIC 	以下の場合0を入力する <ul style="list-style-type: none"> DB2 NT

再接続のパラメータを設定する

このページでは、最初の接続が切断された場合に再接続するためのオプションを設定できます。



接続が失われた場合自動接続する

このオプションを選択すると、サーバへの接続に失敗した場合または接続が切断された場合に、Connect-Itが再接続する方法を指定できます。データベースの場合、データベースが位置するサーバのことを指します。

注意:

このオプションはデフォルトで選択されています。データベースサーバが再接続をサポートしない場合は、このオプションをオフにする必要があります。

再接続用パラメータは以下の通りです。

- **【基本待ち時間】**

【基本待ち時間】では、Connect-Itが何秒後にサーバへの再接続を実行するかを指定します。再接続に失敗すると、Connect-Itは基本待ち時間を2倍にして接続を再試行します。例えば、基本待ち時間が2秒の場合、2回目の再接続は4秒後に実行され、3回目の再接続は8秒後に実行されます。

- **【制限時間】**

この【制限時間】で、Connect-Itがサーバへの再接続を何分（時間）後に中止するかを指定できます。

重要項目:

再接続に成功すると、

- データの書き込みでは、処理の中断されたドキュメントは再び処理され、セッションは通常通り続行します。
- データの読み取りでは、中断されたセッションは最初から再始動されます。

接続テストのクエリ

新規セッション時や処理エラーが発生した場合は、クエリがサーバに送信され、データベースへの接続が切断されたかどうかを確認します。クエリはサーバにサポートされていないかもしれません。また、ドキュメント処理が最適になるようなクエリでなければなりません。

デフォルトの要求はSELECT GetDate()です。この要求がデータベースにサポートされていない場合は、有効な要求を入力します。

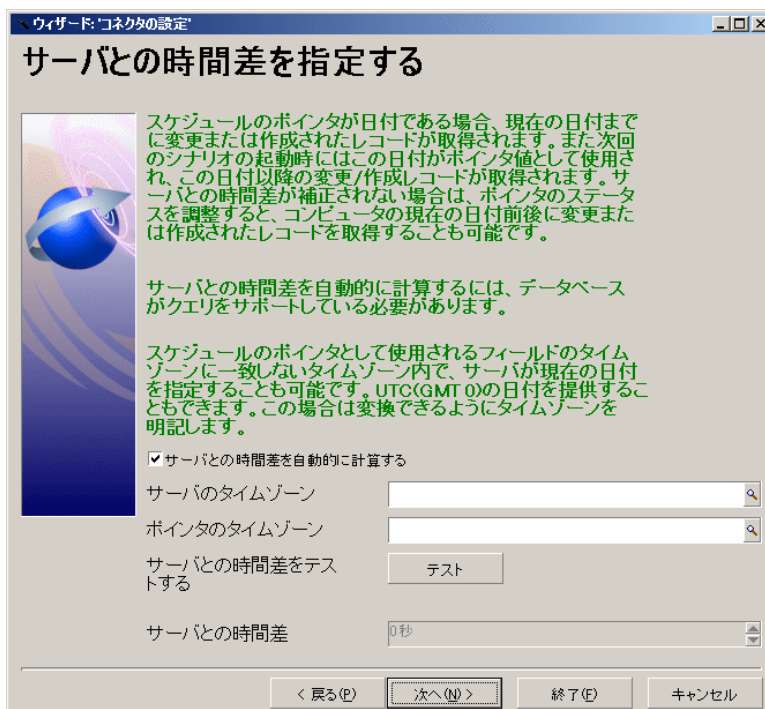
重要項目:

コネクタは、開く度に接続テストの要求を発信します。要求がサーバにサポートされていないと警告メッセージが表示されます。この場合自動的に再接続することは不可能です。

接続が切断されない場合は、このオプションを無効にして、警告メッセージがコネクタの開始時に表示されないようにします。例えば、ExcelファイルへのODBC接続の場合などです。

サーバとの時間差を指定する

このページでは、コネクタがデータを読み取るまたは受信するために接続するサーバとの時間差を、指定することができます。



ウィザード: 'コネクタの設定'

サーバとの時間差を指定する

スケジュールのポイントが日付である場合、現在の日付までに変更または作成されたレコードが取得されます。また次のシナリオの起動時にはこの日付がポイント値として使用され、この日付以降の変更/作成レコードが取得されます。サーバとの時間差が補正されない場合は、ポイントのステータスを調整すると、コンピュータの現在の日付前後に変更または作成されたレコードを取得することも可能です。

サーバとの時間差を自動的に計算するには、データベースがクエリをサポートしている必要があります。

スケジュールのポイントとして使用されるフィールドのタイムゾーンに一致しないタイムゾーン内で、サーバが現在の日付を指定することも可能です。UTC(GMT 0)の日付を提供することもできます。この場合は変換できるようにタイムゾーンを明記します。

サーバとの時間差を自動的に計算する

サーバのタイムゾーン

ポイントのタイムゾーン

サーバとの時間差をテストする

サーバとの時間差

< 戻る(B) Connect-It 3.2.0 - コネクタ ◀ 29

[サーバとの時間差を自動的に計算する] オプションが選択されている

このオプションはデフォルトで選択されています。

この場合Connect-Itはセッションの開始ごとにサーバと通信し、Connect-Itの時計とサーバ間の時間差を自動的に補正します。

[サーバとの時間差を自動的に計算する] オプションが選択されていない

[サーバとの時間差を自動的に計算する] オプションをオフにすると、時間差を手動で入力する必要があります。

コネクタがサーバとの時間差を自動的に決定できない場合は、手動で値を入力する必要があります。


タイムゾーンの使用

サーバが返す日付が、フィルタを実行するために外部アプリケーションが予期しているタイムゾーンに一致しない場合は、[サーバのタイムゾーン] フィールドと (スケジュールの) [ポイントのタイムゾーン] フィールドを使用する必要があります。

例

- サーバは現在のUTC (Universal Time Coordinated) の日付と時刻を返しますが、外部アプリケーションはこの日付と時刻を別のタイムゾーンで保存します。
- 外部アプリケーションは、あるタイムゾーンの現在の日付と時刻を返しますが、この日付にフィルタが適用される場合、Connect-Itサーバの日付と時刻のパラメータで定義されたタイムゾーンが使用されます。

タイムゾーンの指定方法

- 1 [サーバとの時間差を自動的に計算する] オプションが選択されていることを確認します。
- 2  をクリックします。

- 表示されるウィンドウでタイムゾーンを選択します。



アプリケーションが夏時間を使用するかどうかに応じて、**[夏時間を適用する]** オプションを選択します。


- [OK]** をクリックします。

外部アプリケーションの現在の日付と時刻を表示する

[サーバとの時間差をテストする] の右にある **[テスト]** をクリックします。このアクションは以下の操作をトリガします。操作は状況依存ウィンドウに表示されます。

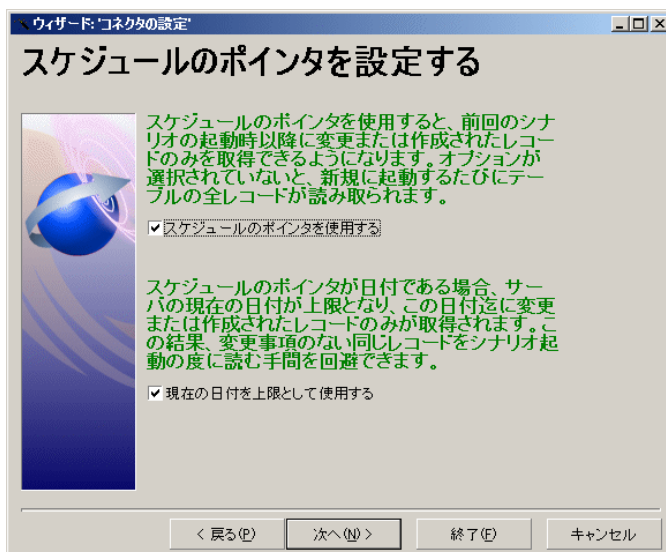
- 外部アプリケーションとの接続
- 外部アプリケーションの現在の日付と時刻が取得され、表示されます。
- タイムゾーンが指定されている場合は、タイムゾーンと時間調整をした後の現在の日付と時刻が表示されます。

タイムゾーンのパラメータをテストする

- 外部アプリケーション内のレコードを変更します。
- このレコードに対応するドキュメントタイプを作成します。
- レコードの最終変更日を示すフィールドにフィルタを適用します。フィルタのシンタックス内のフィールド値は、**[テスト]** ボタンを使って取得した外部アプリケーションの日付と時刻に一致しなければなりません。
- コネクタの生成用ドキュメントタイプの編集ウィンドウで、 をクリックし、変更されたレコードが取得されたかどうかを確認します。

スケジュールのポイントを設定する

このページでは、コネクタが使用するスケジュールのポイントを設定できます。



シナリオがスケジュールモードで起動していると、生成用ドキュメントタイプのポイントのステータスにより、コネクタは前回のセッションで処理されなかったデータのみを処理できます。

例：Asset Managementコネクタは、最後のセッションで9時00分00秒に資産のテーブルのレコードを読み取ったとします。次のセッションでは、コネクタは9時00分01秒から作成されたレコード全てを読み取ります。

スケジュールのポイントを使用する

このオプションはデフォルトで選択されています。

このオプションが選択されていると、コネクタは前セッション以降に変更されたデータのみを処理します。例えばデータベースタイプのコネクタでは、最後のセッション以降に作成または更新されたレコードのみが処理されます。

このオプションが選択されていないと、コネクタはソースアプリケーション内の全データを処理します。例えば、データベースのバックアップコピーを作成するために、データベースタイプのコネクタを使用するとします。この場合、レコードの変更日は使用されません。

現在の日付を上限として使用する

このオプションはデフォルトで選択されています。

! 警告:

データベースタイプのコネクタでの最後のセッションの日付は、前回のセッションで読み取られたレコードの変更の中で、最も最近の変更の日付を指します。

このオプションを説明するために、ソースデータを読み取るコネクタを毎時間起動するスケジューラを例に取ります。最後のセッションは9時00分00秒で、現在のセッションは10時00分00秒に起動します。

[現在の日付を上限として使用する] オプションが選択されている

この場合新規セッションでは、コネクタは以下の変更日時のデータのみを処理します。

- 変更日時が最終セッションの日時と同じ、またはそれ以降
例：全データの変更日時は、9時00分00秒と同じまたはそれ以降
コネクタは、9時00分00秒丁度に変更されたデータを処理します。
- 変更日時が現在の日時より前（サーバとの時間差が自動的に計算される場合は、時間差が調整されます）
例：全データの変更日時は、10時00分00秒よりも前
コネクタは、10時00分00秒より後に変更されたデータを処理しません。

[現在の日付を上限として使用する] オプションが選択されていない

この場合新規セッションでは、コネクタは以下の変更日時のデータのみを処理します。

- 変更日時が最終セッションの日時と同じ、またはそれ以降
例：全データの変更日時は、9時00分00秒と同じまたはそれ以降
コネクタは、9時00分00秒丁度に作成されたデータを処理します。
コネクタは、10時00分00秒より後に作成されたデータを処理します。コネクタがサーバとの時間差を自動的に計算できない場合に便利です。

トランザクションを設定する

このページでは、ドキュメントの処理用にコネクタが使用するモードを定義できます。



コネクタは以下の操作を実行できます。

- [各ドキュメントの後にコミットする]
各ドキュメントの処理後に「コミット」コマンドが実行され、ドキュメントの処理時に発信された挿入クエリ、更新クエリまたは削除クエリを有効にします。
- [ドキュメントグループごとにコミットする]
指定された数のドキュメントが処理された後、「コミット」コマンドが実行され、ドキュメントの処理時に発信された挿入クエリ、更新クエリまたは削除クエリを有効にします。ドキュメント数のデフォルト値は「100」です。

 **注意:**

このモードの使用中にグループ内の1ドキュメントにエラーが発生すると、Connect-Itは、エラーの発生したドキュメントまで、ドキュメントごとの挿入を実行します。ドキュメントの最適な処理には、グループごとのコミットを使用することをお勧めします。

キャッシュを設定する

このページでは、コネクタの使用可能なドキュメントタイプの説明を含むキャッシュを選択できます。キャッシュファイルの拡張子は「.cch」です。



ローカルネットワークにないアプリケーションを使用する場合、キャッシュを使用するとドキュメントタイプの記述を取得する必要がなくなるため、コネクタを速く使用できます。

重要項目:

データへのアクセスを高速化するため、またオフラインで作業するためには、キャッシュの使用が推奨されます。

キャッシュファイルの詳細については、『ユーザガイド』の「統合シナリオのテストとデバッグ」章の「キャッシュファイルを使用する」の節を参照してください。

以下の2つの選択肢があります。

- **【キャッシュファイルを使用する】**

このオプションを使用すると、Connect-Itインストール先フォルダ内の「cache」フォルダ内にあるシナリオと同名のサブフォルダに、キャッシュファイルが作成されます。各キャッシュファイル（「.cch」）の名前は固有

です。このため同一コネクタの他のインスタンスのキャッシュには、別の名前が付きます。一般的にファイルにはサーバ名が使用されます。

- **【特定のキャッシュファイルを使用する】**

このオプションを使用する場合は、作成または再利用するキャッシュファイルの名前とパスを指定します。

このキャッシュファイルが存在しない場合、Connect-Itは、コネクタを最初に関く時にこのファイルを作成します。既に存在する場合は、既存のファイルに上書きするか、またはキャッシュファイルに新規の名前を付けるかを選択します。

 **注意:**

シナリオの前回の作成時に作られたキャッシュファイルを使用する場合は、このオプションを使用します。

重要: 外部アプリケーションの構造が変更されるたびに（データベース型のアプリケーションで新規テーブルが作成される場合や、ServiceCenterで出力イベントが作成される場合など）、キャッシュの同期をとる必要があります。

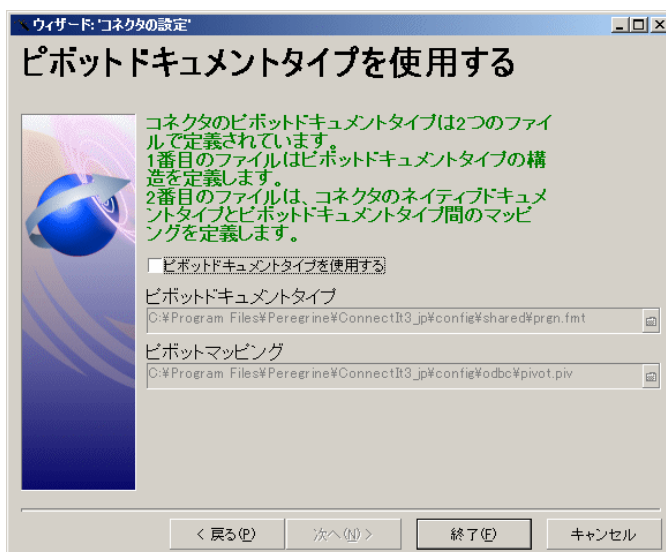
コネクタのキャッシュの同期をとるには、

- 1 シナリオ図内でコネクタを選択します。
- 2 右クリックします。
- 3 [キャッシュ/同期をとる]ショートカットメニューを選択します。
([ツール/キャッシュ/キャッシュの同期をとる])

ピボットドキュメントタイプの使用

このページでは、コネクタ用のピボットドキュメントタイプの使用を設定します。

- FMTファイルは、ピボットドキュメントタイプの説明を含んでいます。
- PIVファイルは、コネクタの使用可能なドキュメントタイプと、ピボットドキュメントタイプ間の既定マッピングを含んでいます。



ピボットドキュメントタイプの詳細については、『ユーザガイド』の「ピボットドキュメントタイプ」の章を参照してください。

 **注意:**

ペレグリンシステムズの顧客サポートから新規のファイルを受け取っていない場合、指定されているファイルを変更することはお控えください。

2 | コネクタのルール（ディレクティブ）

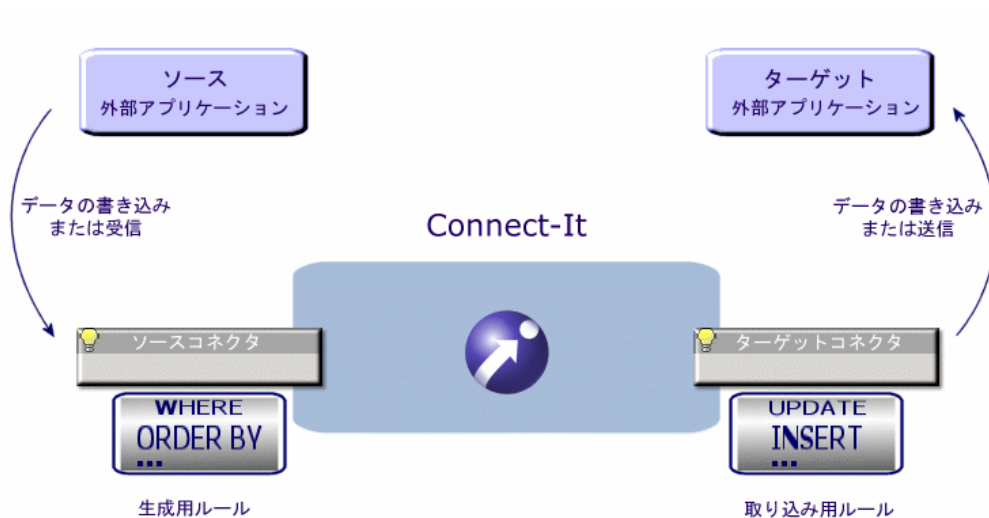
ルール（ディレクティブ）とはユーザが定義するコネクタ用の指示内容のことです。コネクタは、ドキュメントの生成または取り込み時にこの指示に従います。

ルールを作成するのは以下の場合です。

- 生成用または取り込み用のドキュメントタイプを作成する際
または
- ドキュメントタイプ間のマッピングを定義する際

ルールはコネクタに応じて変化します。例えばデータベース型のコネクタでは、生成用ルールはWHERE句やORDER BY句に当たります。これらの句は、ソースデータベース内で取得されるフィールド値にフィルタを適用します。反対に、取り込み用ルールは整合性チェックに相当し、ターゲットデータベースへのデータの書き込みを可能にします。

図 2.1. コネクタ - 生成用ルールと取り込み用ルール




本章では、複数のコネクタに共通の生成用ルールと取り込み用ルールについて説明します。

生成用ルール

生成用ルールは、シナリオ内のソースコネクタが読み取るまたは受け取るデータの処理方法に関連します。例えばMQSeriesコネクタのルールは、WHERE句と、メッセージ取得のオプションから成り立っています。

生成用ルールを定義する

コネクタの生成用ルールを定義するには、

- 1 このコネクタを含むシナリオを読み込むか、または作成します。
- 2 コネクタが正しく設定されているかどうか確認します。
- 3 コネクタを開きます（ [ツール / 開く] または [F4] キー）。
- 4 [ドキュメントタイプ] タブを選択します。
- 5  をクリックするか、または生成用ドキュメントタイプの枠内でドキュメントタイプを編集します。

作業枠が表示され、生成用ドキュメントタイプを編集作成できるようになります。

- 6 作業枠内に生成用ルールを入力します。



WHERE句とORDER BY句

WHERE句では、ソースデータベースのレコードにフィルタをかけることができます。例えば、2002年1月1日以降にデータベースに作成されたレコードのみをフィルタで抽出するとします。これには、[レコードの作成日を含むフィールド] >= '1/01/2002' という句を作成します。

ORDER BY句では、並べ替えの定義に使用するフィールドを指定します。これにより、データベースコネクタはこの並べ替え順でドキュメントを生成します。複数のフィールドを指定する場合は、コンマで区切って入力します。

AQL言語をサポートするコネクタで使用される表記法

Connect-Itでは、AQLを使用するコネクタ用に以下の規則があります。

- フィールド名は角括弧で囲む

例

[フィールド]

- 文字列はシングルクォーテーションマークで囲む

例

'コンピュータ'

- 数値にはクォーテーションマークをつけない
- 日付は、コンピュータの地域の設定オプションで指定したフォーマット従って入力し、シングルクォーテーションマークで囲む

AQLをサポートするコネクタ

AQLをサポートするコネクタは以下の通りです。

Asset Managementコネクタ [p. 60]

InfraTools Managementコネクタ [p. 78]

ゲートウェイ3.xコネクタ [p. 92]

データベースコネクタ [p. 151]

Intel LANDeskコネクタ [p. 231]

Tivoli Inventoryコネクタ (バージョン3.1と3.6) [p. 235]

Tivoli Inventoryコネクタ (バージョン4.0) [p. 241]

SMS 1.xコネクタとSMS 2.xコネクタ [p. 245]

Unicenter AMOコネクタ [p. 250]

Tivoli CM Inventory 4.2 コネクタ [p. 253]

sysdateフィールド

AQLで作成されるWHERE句では、sysdateフィールドは、getdate()関数に置換されなければなりません。

例：SQLクエリSELECT * from AmAsset where dinstall>sysdate は、AQLクエリSELECT * from AmAsset where dinstall>getdate()に置き換えられなければなりません。

配列の取り出しを使用しない

Blobまたはメモタイプのフィールドなどを取得する際、テーブルに主フィールドがない場合は、このオプションを選択する必要があります。配列の取り出しを使用すると、Blobタイプのフィールドは適切に処理されず、切り取られます。このオプションは特に処理の性能を低下させます。

取り込み用ルール

取り込み用ルールは、外部ターゲットアプリケーションに、コネクタが書き込むまたは送信するデータの処理方法に関連します。例えばデータベース型のコネクタのルールでは、ターゲットデータベースにコネクタがレコードを書き込む方法、またはターゲットデータベースのレコードをコネクタが更新する方法を指定します。


コネクタの取り込み用ルールを定義する

コネクタの取り込み用ルールを定義するには、以下の2つの方法があります。

- 取り込み用ドキュメントタイプ編集用ウィンドウ内
- マッピングの編集用ウィンドウ内

ドキュメントタイプ編集用ウィンドウ内で作成する取り込み用ルール


コネクタの取り込み用ルールを、取り込み用ドキュメントタイプ編集用ウィンドウで定義するには、

- 1 このコネクタを含むシナリオを開くか、または作成します。
- 2 コネクタが正しく設定されているかどうか確認します。
- 3 コネクタを開きます（[ツール/開く]または[F4]キー）。
- 4 [ドキュメントタイプ]タブを選択します。
- 5 をクリックするか、または取り込み用ドキュメントタイプの枠内で取り込み用ドキュメントタイプを編集します。
ドキュメントタイプの編集用ウィンドウが表示されます。ここで取り込み用ドキュメントタイプを作成または編集します。
- 6 編集用ウィンドウの作業枠内に取り込み用ルールを入力します。

マッピングの編集用ウィンドウ内で作成する取り込み用ルール

マッピングの編集用ウィンドウでコネクタの取り込み用ルールを定義するには、

- 1 このコネクタを含むシナリオを開くか、または作成します。
マッピングは、シナリオ内でこのコネクタと、別のコネクタの間に作成されていなければなりません。
- 2 シナリオのコネクタを開きます。
- 3 [マッピング]タブを選択します。

- 4 をクリックするか、または既存のマッピングを編集します。
マッピングの編集用ウィンドウが開きます。
- 5 マッピングウィンドウの作業枠内に取り込み用ルールを入力します。

整合性

データベース型のコネクタでは、整合性チェック用のパラメータを入力して取り込み用ルールを作成します。

パラメータには以下の内容を入力します。

- コネクタが取り込むデータと、ターゲットデータベースのレコード間の整合性を確認するために使用するフィールド
- コネクタが、ターゲットデータベースのレコードに実行する操作のタイプ（挿入、更新または削除）

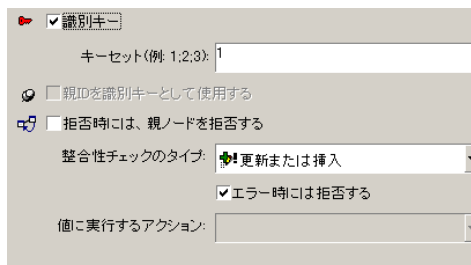
取り込み用ルールは、[整合性]タブと[高度な整合性チェック]タブ内で入力します。

[整合性]タブの使用方法

[整合性]タブでは以下の操作を実行できます。

- 1 複雑な要素（ルートノード、コレクション、構造体）ごとに、整合性チェックのタイプを指定する
- 2 識別キーとして使用されるサブノードを指定する

図 2.2. 取り込み用ルール - [整合性]タブ



識別キー
 キーセット(例: 1;2;3):
 親IDを識別キーとして使用する
 拒否時には、親ノードを拒否する
 整合性チェックのタイプ:
 エラー時には拒否する
 値に実行するアクション:

整合性チェックのタイプを選択する

[整合性チェックのタイプ]では、実行するアクションを選択できます。

- [更新または挿入]

- [更新のみ]
- [挿入のみ]
- [削除]

整合性チェックのタイプを指定する方法

- 1 ルートノード、構造体、またはコレクションをマッピング用ウィンドウの作業枠内で選択します。
- 2 [整合性チェックのタイプ] のドロップダウンリストからオプションを選択します。

識別キーを選択する

複雑な要素用に整合性チェックのタイプを選択した後、ソースデータとターゲットデータ間の整合性チェックで使用するフィールドを選択します。これらのフィールドには識別キーが付きます。

識別キーとして選択されるフィールドは、以下の基準を満たさなければなりません。

- 重複する値の禁止
例：データベースのテーブル内の [ID] フィールドでは、複数のレコードに同じ識別子を使用することが禁止されています。これに反して、[従業員] テーブルの [名前] フィールドは識別キーとしては不適切です。これは、複数の従業員が同名である可能性があるからです。
- NULL値の禁止
例：IT業者のデータベースの [コンピュータ] テーブルの [シリアル番号] フィールドは、識別キーとして適切です。シリアル番号のないコンピュータは存在しないためです。
- インデックス化
識別キーとしてインデックス化されたフィールドを選択すると、整合性チェックがより迅速に実行されます。

識別キーを選択する方法

- 1 要素を作業枠内で選択します。
- 2 [識別キー] オプションを選択するか、または取り込み用ドキュメントタイプが表示される枠内の透明なキーを、直接クリックします。
例：idd /ddac36 /ddac.scnシナリオのamAssetドキュメントタイプでは、[AssetTag] フィールドと、Product構造体の [Brand] と [Model] フィールドが識別キーとして選択されています。

構造体が識別キーとして選択されている場合

Asset Managementコネクタの使用可能なドキュメントタイプの第1レベルは、Asset Managementアプリケーションの1つのテーブルに当たります。このテーブルに1対1または1対N型のリンクで関連付けられているテーブルは、構造体になります。

1つの構造体内の複数フィールドを識別キーとして選択すると、レコードを識別できるようになります。これらのフィールドがデータの転送時に含まれるようにするには、フィールドの所属先である構造体も識別キーとして選択しなければなりません。

実際には、各構造体は1つの**識別子**フィールドを含んでおり（**画面上では非表示**）、この識別子が他のテーブルへのリンクを可能にします。構造体を識別キーとして指定するという事は、この**識別子**フィールドがレコードの識別に使用されるということです。



資産のドキュメントタイプが作成するレコードは、以下のフィールドにより識別されています。

- [資産] ルートノードの [名前] フィールド
- [製品] 構造体の [識別子] フィールド
- [製品] 構造体の [モデル] フィールド

親のIDを識別キーとして使用する方法

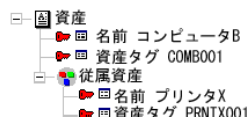
- 1 構造体またはコレクションを選択します。
- 2 [親IDを識別キーとして使用する] オプションを選択します。

例：Asset Managementアプリケーションのデータベースでは、子テーブルは外部キーによって親テーブルに関連付けられています。この外部キーは親テーブルの識別子です。AssetCenterでは、ポートフォリオ品目のテーブル（amPortfolio）は、従属資産のテーブル（実際はポートフォリオ品目のテーブルと同一）に [parentId] フィールドによりリンクされています。

[親IDを識別子として使用する] チェックボックスをオンにすると、他のレコードに親/子の関係でリンクされているレコードを作成する際、その親子関係がデータベース内に既存する親子関係と違う場合は、レコードは拒否されます。これは、2つのテーブル間のリンクが1対1または1対Nの場合のみ有効です。子

テーブル内の1つのレコードは、親テーブルの1つのレコードにのみリンクされます。

コンピュータAはプリンタXに親子関係でリンクされています。Asset Management コネクタに取り込まれるドキュメントは、コンピュータBを同じプリンタXにリンクします。この場合、資産の名前と資産タグ、また従属資産の名前と資産タグが識別キーとして選択されています。



[親IDを識別キーとして使用する] オプション - 例

この例では、[親IDを識別キーとして使用する] オプションを選択するまたはしない場合に、Connect-Itが実行する操作について説明します。

1 [親IDを識別キーとして使用する] を選択した場合、

[親IDを識別キーとして使用する] フィールドは、従属資産であるプリンタXが既に資産のテーブル内の資産にリンクされていることを意味するため、親子関係のリンクは更新されません。従属資産「プリンタX」を挿入しようとする、親IDが識別キーとして使用されます。しかしこの資産は既に存在するため、多くの場合はデータベース保全の理由から資産は挿入されません。

資産のテーブル		従属資産のテーブル			
名前	資産タグ	識別子	親識別子	名前	資産タグ
コンピュータA	COMPA001	CPAAA1	-0-	CPAAA1	プリンタX PRNTX001
コンピュータB	COMPB001	CPBBB1			

2 [親IDを識別キーとして使用する] が選択されていない場合、

親子関係のリンクは更新され、従属資産には新しい親資産がリンクされます。[親ID(識別子)] フィールドは更新され、従属資産「プリンタX」は資産「コンピュータB」へリンクされます。

資産のテーブル		従属資産のテーブル	
---------	--	-----------	--

名前	資産タグ	識別子	親識別子	名前	資産タグ
コンピュー タB	COMPB001	CPBBB1	- 0 -	0CPBBB1	プリンタX PRNTX001
コンピュー タA	COMPA001	CPAAA1			

子ノードが拒否された時に親ノードも拒否するよう指定する方法

子ノードが拒否された時には、コネクタが自動的に親ノードを拒否するように指定する場合があります。

- 1 作業枠からドキュメントタイプの1要素を選択します。
- 2 **【拒否時には、親ノードを拒否する】**オプションを選択します。

例：Asset Managementアプリケーションでは、あるレコードが[従業員] (amEmplDept) のテーブルで拒否された場合、そのレコード(ユーザ)にリンクされているレコード(ポートフォリオ品目)が[ポートフォリオ品目]のテーブルに作成されないよう設定できます。これにより、Asset Managementアプリケーションのデータベースに、ユーザの関連付けられていないポートフォリオ品目が作成されることを回避できます。

複数の識別キーセットを使用する

Connect-Itでは、複数の識別キーセットを使用できます。

これは、選択した識別キーではレコードの整合性チェックを実行できない場合に便利です。**例：**識別キーとして選択されたフィールドに、ソースドキュメントでは値が入力されていない一方で、ターゲットアプリケーションでは、このフィールドが入力必須である場合。

ある識別キーが2番目のセットに所属すると指定すると、1番目の識別キーセットで整合性チェックに失敗した場合に、Connect-Itはこのキーを使用します。

複数の識別キーセットの機能

識別キーの数には制限はありません。キーセットには1からNまでの番号が付けられています。この番号は、**【識別キー】**オプションの下にある**【キーセット】**フィールドで指定します。

Connect-Itは、識別キーセットを番号順にテストします(第1キーセット、第2セット、第3セットなど)。

ドキュメントタイプは複数の複雑要素(フィールド、構造体、コレクション)から構成されています。1つのキーセットとそのアクション(更新、挿入、削除)は、ドキュメントタイプの各複雑要素に関連付けられます。このため、複

数の要素が整合性チェックされる場合、同じドキュメント内にも、数種の整合性チェックのタイプが共存することがあります。

識別キーが、あるセットに所属することを指定する方法

- 1 識別キーとして指定するターゲット要素を選択します。
- 2 **[識別キー]** オプションを選択します。
- 3 **[キーセット]** フィールドに、識別キーの所属先である1つまたは複数のキーセットを、セミコロンで区切って入力します。

全ての識別キーは、デフォルトでキーセット1に所属します。

識別キーセットの数は、ドキュメントログ内に表示されます。

例：**[キーセット]** フィールドに「1;4」と指定すると、選択された識別キーは、**キーセット1**と**キーセット4**に所属します。

大文字小文字の区別

このオプションにより、Connect-Itの整合性チェックは以下のようになります。

- **[大文字小文字の区別]** オプションを選択する
このオプションが選択されていると、識別キー値の大文字と小文字は区別して取り扱われます。例：従業員のEメールアドレスを含むフィールドが識別キーとして使用されている場合、'jmartin@company.com'、'JMARTIN@company.com' と 'jMartin@company.com' は、ターゲットアプリケーション内で3つの別々のレコードになります。
- **[大文字小文字の区別]** オプションを選択しない
このオプションを選択しないと、識別キー値の大文字と小文字は区別されません。例：従業員のEメールアドレスを含むフィールドが識別キーとして使用されている場合、'jmartin@company.com'、'JMARTIN@company.com' と 'jMartin@company.com' は、ターゲットアプリケーション内で1つの同じレコードになります。

注意:

このオプションは、使用するデータベースエンジンが大文字小文字を区別できる場合にのみ表示されます。

整合性チェックが実行するアクションを選択する

各ターゲットフィールドごとに、以下の2オプションから選択します。

- 置換
- 追加

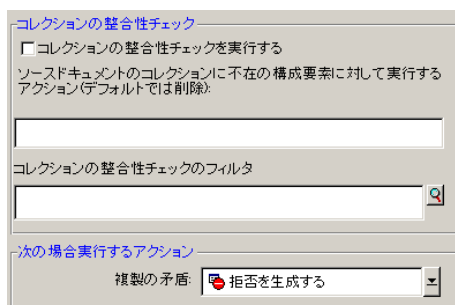
デフォルトでは、**[置換]**オプションが選択されています。メモフィールドでは**[追加]**オプションが便利です。例えば、データベースのテーブル内に既存するコメントにコメントを追加することができます。

[高度な整合性チェック] タブの使用方法

[高度な整合性チェック] タブでは、以下の操作を実行できます。

- 取り込まれるドキュメントの、コレクション用の整合性チェックのパラメータを設定する。
- 複製の矛盾が起きた場合にコネクタが実行するアクションを指定する。

図 2.3. 取り込み用ルール - [高度な整合性チェック] タブ



コレクションの整合性チェックを実行する

[コレクションの整合性チェックを実行する] オプションでは、ソースコレクションの構成要素に応じて、ターゲットコレクションの構成要素での更新または挿入を管理するためのパラメータを指定できます。

ソースコレクションに不在の構成要素を管理する

[ソースドキュメントのコレクションに不在の構成要素に対して実行するアクション(デフォルトでは削除)] フィールドでは、以下の内容を設定できます。

- ソースコレクションに不在の構成要素は、ターゲットアプリケーションから削除されない。
- ソースコレクションに不在の構成要素に対応するターゲットアプリケーションのフィールドが、特定の値を受け取る。

例えば、ターゲットアプリケーション内のコレクションの**[ステータス]**フィールドに、「**削除予定**」という値を割り当てます。これにより、ター

ゲットアプリケーションの管理者は [ステータス] フィールドが「削除予定」であるレコードを検索できるようになります。

フィールドに使用するシンタックスは次の通りです。

[フィールド名] = "[選択したフィールド値]"

例

ステータス = "削除予定"

コレクションの一部の構成要素にフィルタを適用する

コレクションを更新する場合は、[コレクションの整合性チェックのフィルタ] フィールドで以下の内容を指定できます。

- ソースドキュメントに不在のコレクションの構成要素を、ターゲットアプリケーションのデータベースから削除する。
- ターゲットアプリケーションのデータベース内のコレクションの構成要素の一部にフィルタをかけて、データの削除を制限する。

例：

Desktop DiscoveryコネクタとAsset Managementコネクタ間のAsset informationマッピングで、ソースコレクションSoftwareがターゲットコレクションSoftInstallにマップされているとします。これらの2つのコレクションは資産（この場合はコンピュータ）にインストールされているソフトウェアを表しています。

時にはいくつかのソフトウェアがアンインストールされる場合もあります。[コレクションの整合性チェックを実行する] オプションを選択すると、SoftInstallコレクションに不在のソフトウェアは、Asset Managementアプリケーションのデータベースから削除されます。

ある一定のソフトウェア（Desktop Discoveryのスキャンに認識されないソフトウェアなど、）がデータベースから削除されてしまうのを防ぐには、[コレクションの整合性チェックのフィルタ] フィールドにフィルタ句を作成します。

図 2.4. 整合性チェック - コレクションのフィルタ

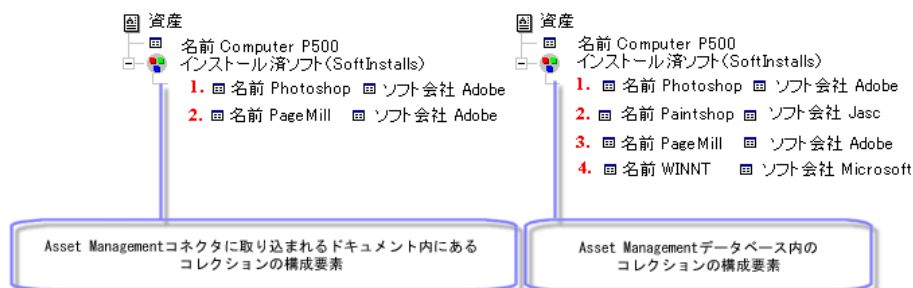
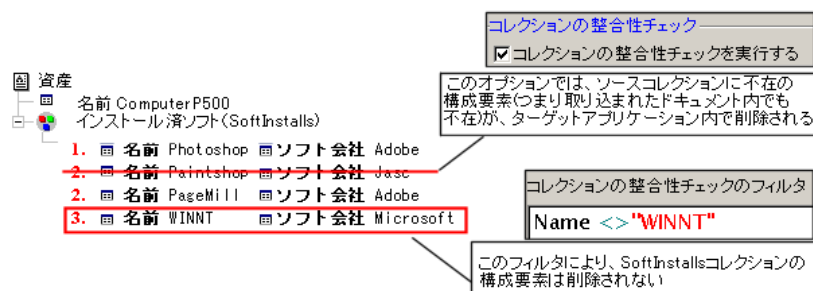


図 2.5. 整合性チェック - ターゲットコレクションの構成要素の削除



複製 (レプリケーション) の矛盾を管理する

複製の矛盾が起きた場合、以下の3つのオプションがあります。

- [拒否を生成する]
- [ログに警告を記録する]
- [上書きする]

ServiceCenterとAsset Managementアプリケーション間のシナリオでの複製の矛盾に関する詳細は、「付属シナリオ [p.301]」の章の「ServiceCenterシナリオ [p.314]」の以下の節を参照してください。

- データの整合性チェックとデータの複製の相違点 [p. 318]
- Connect-It内でのデータの複製の管理 [p. 319]

3 | Peregrine Systemsコネクタ

Peregrine Systemsコネクタは、Peregrine Systemsアプリケーションのデータを処理するコネクタです。

Action Request Systemコネクタ

ベースコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

Action Request Systemコネクタは、Action Request Systemデータベースから来るデータを処理します。

Action Request Systemコネクタの互換性

Action Request Systemコネクタは、Action Request Systemの以下のバージョンと互換性があります。

- 3.xx
- 4.xxから4.5
- 5.0

Action Request Systemコネクタ：ダイナミックライブラリ

コネクタを使用するために必要なダイナミックライブラリ（DLL）は、「[Connect-Itインストール先フォルダ]/bin」にあります。

最新版のコネクタは、このダイナミックライブラリを使用しないと機能しません。このDLLは、バージョン5.0とそれ以前のAction Request Systemにアクセスできるようにします。

- arapi45.dll
- arrpc45.dll
- aruti45.dll
- arcatalog_eng.dll

この最後のライブラリには、Action Request Systemエラーコードの完全なテキストが含まれています。

Action Request Systemコネクタの既知の制約点

読み取り（生成）と書き込み（取り込み）で、Action Request Systemは、以下のデータ型のAction Request Systemフィールドを処理します。

- テキスト（text）
- ダイアリ（diary）

読み取り（生成）モードで、このフィールドはこのフィールドの履歴全体を取得します。

書き込み（取り込み）モードで、このフィールドは、データを書き込むユーザ名、およびデータ書き込みの日時を含む文字列を挿入します。

- 整数（integer）
- 実数（Real number）
- ラジオボタン（Radio button）

このフィールドを使用するには、初期化する要素の番号を指定する必要があります。値を直接指定することはできません。

- ドロップダウンリスト（Drop down list）
- 日時（DateTime）
- 添付ファイル

コネクタは各添付ファイルごとに、1構造体を使用可能なドキュメントタイプとして発行します。この構造体は以下の3つのフィールドを表します。

- FileName
添付ファイルの完全パスを含みます。
- AttachMemoVal

添付テキストファイルの場合。このフィールドはファイルの内容を含みません。

- AttachBlobVal

このファイルは常に、添付ファイルのバイナリ内容を含みます。

書き込み（取り込み）モードでは、添付ファイルの完全パスをARS attachment FieldNameフィールドへマップする必要があります。

メモフィールドまたはBlobフィールドのコピーを共有フォルダ内に作成する場合は、共有ファイルのパスをARS attachment FieldNameにマップし、ファイルの内容をAttachMemoValまたはAttachBlobValにマップします。

注意:

Action Request Systemのバージョン5.0以降では、MemoファイルとBlobファイルはデータベース内に直接格納されます。共有ディレクトリ内にファイルのコピーをおく必要はなくなりました。

読み取り（生成）モードでは、以下のマッピングを実行します。

- AttachMemoVal フィールドをmemoタイプのフィールドへマップする。
- AttachBlobValフィールドをblobタイプのフィールドへマップする。
- FieldNameフィールドをblobタイプのフィールドへマップする。

マッピングの編集用ウィンドウで、**[ファイルからBLOBを読み込む]** オプションが選択されているかどうか確認します。

このオプションの詳細については、『ユーザガイド』の「ドキュメントタイプのマッピング」の章の「マッピングの種類」の節、「Blobタイプのフィールドのマッピング」を参照してください。

Action Request Systemコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 3.1. Action Request Systemコネクタの設定

 コネクタに名前を付け、役割を説明する [p. 14]

 接続用パラメータを定義する [p. 56]

 高度な設定 [p. 24]

 キャッシュを設定する [p. 35]

 再接続のパラメータを設定する [p. 27]

☰ _____ サーバとの時間差を指定する [p. 29]

☰ _____ スケジュールのポインタを設定する [p. 32]

☰ _____ ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続用パラメータを定義する

このページでは、Action Request Systemサーバへのコネクタ接続パラメータを設定できます。

[サーバ名]

このフィールドにはサーバ名を指定します。ネットワーク上のサーバ名、またはIPアドレスを入力します。

[ログイン]

Action Request Systemのユーザ名を入力します。このログインのプロファイルは、シナリオ内でのアクション（データの読み取りまたは書き込み）を実行できるようなプロファイルでなければなりません。

[パスワード]

ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

[使用するダイナミックライブラリ]

Action Request Systemサーバへの接続を可能にするダイナミックライブラリの完全パスを指定します。バージョン5.0またはそれ以降のライブラリを使用しなければなりません。

[スキーマの最終変更フィールドのリスト] (高度な設定)

Action Request Systemスキーマには、スキーマのレコードの「最終変更日」を示すフィールドがあります。多くの場合 [Modified-date] フィールドが使用されません。しかし、スキーマとAction Request Systemのバージョンに応じて、このフィールド名を変更することもできます。この場合このページの編集ゾーンでは、以下の内容を指定できます。

- 最終変更日用のフィールドをAction Request Systemスキーマ用に指定する際、このAction Request Systemスキーマに対応する使用可能なドキュメントタイプ
- Action Request Systemスキーマ内に表示される、最終変更日用のフィールドの名前


最終変更日用フィールドのデフォルト値は「Modified_date」です。編集用ゾーンに何も入力しない場合は、この値が全ドキュメントタイプに使用されます。

Action Request Systemの使用可能な各ドキュメントタイプごとに、「最終変更日」用の異なるフィールド名を指定できます。


警告:

データベースで指定されている最終変更日用のフィールド名を使用してはなりません。Action Request Systemスキーマで使用されている名前を使います。

スキーマと最終変更日用のフィールドを新規に関連付けるには、

- 1  をクリックします。
- 2 **[スキーマ]** 列内のテキストゾーンをクリックし、Action Request Systemコネクタの使用可能なドキュメントタイプ(スキーマ)の名前を入力します。
- 3 **[フィールド]** 列をクリックし、スキーマで使用される最終変更用フィールドの名前を入力します。

既存のフィールドを削除するには、

- 1 フィールドに対応する行を選択します。
- 2  をクリックします。

高度な設定

このページの使用方法については、「コネクタの設定 [p. 13]」章の「高度な設定 [p. 24]」の節を参照してください。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「再接続のパラメータを設定する [p. 27]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p.13]」章の「スケジュールのポイントを設定する [p.32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p.13]」章の「キャッシュを設定する [p.35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p.13]」章の「ピボットドキュメントタイプの使用 [p.36]」の節を参照してください。

生成用ルール - Action Request Systemコネクタ

コネクタの生成用ルールの入力方法については、「コネクタのルール（ディレクティブ） [p.39]」章の「生成用ルール [p.40]」の節を参照してください。

このコネクタでは、これらのルールはWHERE句やORDERBY句で作成されており、ソースデータベースのレコードにフィルタをかけることができます。これらの句の詳細については「コネクタのルール（ディレクティブ） [p.39]」章の「生成用ルール [p.40]」節、「WHERE句とORDER BY句 [p.41]」を参照してください。

以下の表はWHERE句やORDERBY句で使用可能な演算子と、オペランドのリストです。

演算子

=
<>
<
>
AND
OR
NOT
LIKE (特殊文字：%)

オペランドのタイプ

オペランドのタイプ	例
数値	1 (シングルクォーテーションマークなし)
文字	'a' (シングルクォーテーションマーク)
日付	'01/12/24' (国際標準)

オペランドのタイプ

文字列

例

'文字列' (シングルクォーテーションマーク)

Action Request Systemコネクタの取り込み用ルール

コネクタの取り込み用ルールの入力方法については、「コネクタのルール (ディレクティブ) [p. 39]」章の「取り込み用ルール [p. 43]」の節を参照してください。

取り込み用ルールを作成するには、**[整合性]**と**[高度な整合性チェック]**タブ内で整合性用のパラメータを入力します。

これらの整合性用タブの詳細については、「コネクタのルール (ディレクティブ) [p. 39]」章の「取り込み用ルール [p. 43]」節内「整合性 [p. 44]」を参照してください。

追加情報 - Action Request Systemコネクタ

本節ではコネクタに関する追加情報を説明します。

Action Request Systemコネクタの使用可能なドキュメントタイプ内のフィールド名

Action Request Systemコネクタの使用可能なドキュメントタイプでは、文字「#」は「_」に置き換えられます。

例：使用可能なドキュメントタイプ内で、**[フィールド#1]**は**[フィールド_1]**と表示されます。

[フィールド#1]または**[フィールド_1]**を呼出すクエリの結果は同じになります。これはAction Request Systemが、表示される名前と内部識別子 (ID) を区別しないためです。

付属シナリオ - Action Request Systemコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- `rsm /sm4ac36 /cmpo.scn` [p. 307]
- `rsm /sm4ac36 /sharedat.scn` [p. 308]
- `idd /ddrsm4 /ddrsm.scn` [p. 303]
- `ind /ndrsm4 /ndrsm.scn` [p. 306]

Asset Managementコネクタ

ベースコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

本節ではAsset Managementコネクタについて説明します。

Asset Managementコネクタの互換性

Asset Managementコネクタは、AssetCenter、AssetCenter Cable and CircuitとInfraCenter for Workgroupsの3つの外部アプリケーション用のコネクタです。これらのアプリケーションの以下のバージョンが使用可能です。

- AssetCenter 3.xxと4.xx
- AssetCenter Cable and Circuit 3.10
- InfraCenter for Workgroups 2.0x

Connect-Itは、これらのアプリケーション付属のDLLとAPIを使用します。

Asset Managementアプリケーションクライアントは、Connect-Itがインストールされているコンピュータにインストールされなければなりません。

Asset Managementアプリケーションの設定に関する注意事項

「amdb.ini」ファイル（4.2.1より前のバージョンのAssetCenter）

このファイルは、Asset Managementアプリケーションがインストールされているコンピュータ上に作成されているデータベース接続のリストを含んでいます。このファイルは、Asset Managementアプリケーションのインストール時に自動的に作成され、Windowsのシステムフォルダに含まれます。

例

C:\WINNT\amdb.ini

Connect-ItのAsset Managementコネクタは、このファイルを読み込み、コネクタの設定時に使用可能な接続のリストを表示します。使用中のコンピュータが、Asset Managementアプリケーションへアクセスできない場合、Asset Managementコネクタは使用不可能です。

「amdb.ini」ファイル（4.2.1バージョン以降のAssetCenter）

バージョン4.2.1以降のAssetCenterでは、コンピュータで使用可能な接続は、複数の「amdb.ini」ファイル内に宣言されています。これらのファイルは以下の場所にあります。

- システム接続の場合はWindowsのシステムフォルダ
- ユーザ接続の場合は、接続を作成したWindowsユーザの名前の「C:\Documents and Settings」のサブフォルダ

例：C:\Documents and Settings\Tanaka\amdb.ini

任意管理項目のテーブル

Asset Managementアプリケーションのデータベースでは、任意管理項目を用いてデータモデルを拡張できます。

Connect-Itにおいて、任意管理項目はAsset Managementコネクタの使用可能なドキュメントタイプ内で、フィールドまたはコレクションの形で表示されます。

例：

例：amAssetドキュメントタイプでは、資産（この場合はコンピュータ）に使用されているDOSのバージョンはfv_DOSVersionフィールドに表示されます。

Desktop DiscoveryとNetwork Discoveryからのデータを完全に統合するためには、データ転送される任意管理項目が、Asset Managementアプリケーションのデータベース内で適切に作成されていなければなりません。適切に作成されていないと、ドキュメント内の任意管理項目の値はインポートされません。

Connect-Itのデータキットには、Asset Managementアプリケーションのデータベース内にインポートしなければならないテキストファイル（「stdfeat.txt」）があります。このファイルをインポートすると、データベースの[任意管理項目]テーブル内にレコードが作成されます。

AssetCenterへのインポートについては、AssetCenterの『データベース管理』のマニュアルを参照してください。

Asset Managementデータベースの必須フィールド

Asset Managementアプリケーションでは、テーブルのフィールドやリンクが必須である場合があります。これはデフォルトで必須に設定されているか、またはAsset Managementアプリケーションの管理者により必須に設定されています。

整合性チェック時に、Asset Managementコネクタの使用可能なドキュメントタイプ内の各構造体は、1つのレコードに当たります。この構造体内の1要素が必須

フィールドであるにも関わらず入力されていない場合は、要素の所属する構造体は拒否されます。

例：

資産のテーブル内では [AssetTag] は必須フィールドです。資産のテーブルに対応するドキュメントタイプ内で、このフィールド用の要素が入力されていないと、要素の所属先の構造体は拒否されます。

API : DLL

Asset Managementコネクタを適切に使用するには、AssetCenter、AssetCenter Cable and Circuit、InfraCenterの完全インストールを実行するか、またはカスタムインストールで、コンポーネント「API」を選択する必要があります。

Asset Managementコネクタの制約点

Asset Managementアプリケーションは、1970年より前と2038年より後の日付と時間を処理しません。










Asset Managementコネクタと他のコネクタ間のシナリオが、1970年と2038年より前または後の日付を使用すると（例えばマッピングスクリプト内で使用される日付など）、シナリオの実行時にエラーメッセージが発生します。

Asset Managementコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 3.2. Asset Managementコネクタの設定

	コネクタに名前を付け、役割を説明する [p. 14]
	接続を選択する [p. 16]
	再接続のパラメータを設定する [p. 27]
	サーバとの時間差を指定する [p. 29]
	スケジュールのポイントを設定する [p. 32]
	高度な設定 [p. 63]
	トランザクションを設定する [p. 34]
	キャッシュを設定する [p. 35]
	ピポットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

接続を選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続を選択する [p. 16]」の節を参照してください。

高度な設定



警告:

このページは、オプション [高度なコネクタ設定ウィザードを表示する] が選択されていないと表示されません ([編集 / オプション] メニュー)。

[使用するダイナミックライブラリ]

このオプションで、Asset Managementアプリケーションに関連するAPIを含むDLLの完全なパスを指定できます。例えばAssetCenter 3.51では、C:\Program Files\Peregrine\AssetCenter\bin32\ampi35.dllになります。デフォルトで、コネクタは自動的に使用するDLLの場所を見つけます。

[テーブルの主キーを表示する]

このオプションを選択すると、Asset Managementコネクタの使用可能なドキュメントタイプに、データベースのテーブルの主キーに対応するフィールドが含まれます。

例: このオプションを選択すると、フィールド「IAstId」が、コネクタの使用可能なドキュメントタイプamAsset内に現れます。資産テーブルの各レコード用のこの識別子は、資産テーブルの主キーです。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「再接続のパラメータを設定する [p. 27]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

Connect-Itのセッションの終了時に接続を切断する

Connect-Itヘルプデスクの技術者に推奨されていない場合は、このオプションを使用しないでください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

トランザクションを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

Asset Managementコネクタの生成用ルール

コネクタの生成用ルールの入力方法については、「コネクタのルール（ディレクティブ） [p. 39]」章の「生成用ルール [p. 40]」の節を参照してください。

このコネクタでは、これらのルールはWHERE句やORDERBY句で作成されており、ソースデータベースのレコードにフィルタをかけることができます。これらの句の詳細については「コネクタのルール（ディレクティブ） [p. 39]」章の「生成用ルール [p. 40]」節、「WHERE句とORDER BY句 [p. 41]」を参照してください。

AQLシンタックスの詳細については、「AQLクエリ [p. 331]」の章を参照してください。

以下の表はWHERE句やORDERBY句で使用可能な演算子と、オペランドのリストです。

演算子

=

<>

演算子

<
>
AND
OR
NOT
LIKE (特殊文字: %)

オペランドのタイプ

オペランドのタイプ	例
数値	1 (シングルクォーテーションマークなし)
文字	'a' (シングルクォーテーションマーク)
日付	'01/12/24' (国際標準)
文字列	'文字列' (シングルクォーテーションマーク)

sysdateフィールド

AQLで作成されるWHERE句では、sysdateフィールドは、getdate()関数に置換されなければなりません。

例 : SQLクエリSELECT * from AmAsset where dinstall>sysdate は、AQLクエリSELECT * from AmAsset where dinstall>getdate()に置き換えられなければなりません。

Asset Managementコネクタの取り込み用ルール

コネクタの取り込み用ルールの入力方法については、「コネクタのルール(ディレクティブ) [p. 39]」章の「取り込み用ルール [p. 43]」の節を参照してください。

取り込み用ルールを作成するには、**[整合性]**と**[高度な整合性チェック]**タブ内で整合性用のパラメータを入力します。

これらの整合性用タブの詳細については、「コネクタのルール(ディレクティブ) [p. 39]」章の「取り込み用ルール [p. 43]」節内「整合性 [p. 44]」を参照してください。

付属シナリオ - Asset Managementコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- 1 rsm/sm4ac36/compn.scn [p. 307]
- 2 rsm/sm4ac36/sharedat.scn [p. 308]

- 3 scac/sc3ac36/scac.scn [p. 314]
- 4 scac/sc3ac40/scac.scn [p. 320]
- 5 scac/sc3ac41/scac.scn [p. 321]
- 6 scac/sc4ac40/scac.scn [p. 321]
- 7 scac/sc4ac41/scac.scn [p. 321]
- 8 ind/indac36/indac.scn [p. 305]
- 9 ind/indac40/indac.scn [p. 306]
- 10 ind/indac41/indac.scn [p. 306]
- 11 idd/iddac36/iddac.scn [p. 302]
- 12 idd/iddac40/iddac.scn [p. 302]
- 13 idd/iddac41/iddac.scn [p. 302]
- 14 amo/amoac36/amoac.scn [p. 302]
- 15 ild/ild64ac36/ildac.scn [p. 304]
- 16 ild/ild65ac36/ildac.scn [p. 304]
- 17 ldap/ldapac36/complete.scn [p. 308]
- 18 ldap/ldapac36/simple.scn [p. 308]
- 19 mail/mailac36/finreque.scn [p. 309]
- 20 mail/mailac36/finconfi.scn [p. 310]
- 21 mail/mailac36/hewemplo.scn [p. 310]
- 22 mail/mailac40/finconfi.scn [p. 310]
- 23 mail/mailac40/finreque.scn [p. 311]
- 24 mail/mailac40/hewemplo.scn [p. 311]
- 25 mqseries/acmq/mqtoprgn.scn [p. 311]
- 26 mqseries/acmq/prgntomq.scn [p. 312]
- 27 ntsec/htac36/addcpu.scn [p. 313]
- 28 ntsec/htac36/adduser.scn [p. 313]
- 29 ntsec/htac36/adduser.scn [p. 313]
- 30 ntsec/htac40/addcpu.scn [p. 313]
- 31 ntsec/htac40/adduser.scn [p. 313]
- 32 ntsec/htac41/addcpu.scn [p. 314]
- 33 ntsec/htac41/adduser.scn [p. 314]
- 34 scac/sc3ac40/scac.scn [p. 320]
- 35 scac/sc3ac41/scac.scn [p. 321]
- 36 scac/sc4ac40/scac.scn [p. 321]
- 37 scac/sc4ac41/scac.scn [p. 321]
- 38 acscups/dataprop/01-SCcompany2ACamCompany.scn [p. 322]
- 39 acscups/dataprop/02-SClocation2ACamLocation.scn [p. 322]

- 40 acscups /dataprop /03-SCdept2ACamEmplDept.scn [p. 322]
- 41 acscups /dataprop /04-SCcontacts2ACamEmplDept.scn [p. 322]
- 42 acscups /dataprop /05-SCdevtype2ACamNature.scn [p. 323]
- 43 acscups /dataprop /06-SCocmlcat2ACamNature.scn [p. 323]
- 44 acscups /dataprop /07-SCmodel2ACamModel.scn [p. 323]
- 45 acscups /dataprop /08-SCdevtype2ACamModel.scn [p. 323]
- 46 acscups /dataprop /09-SCdevice2AACamPortfolioAndamComputer.scn [p. 323]
- 47 acscups /dataprop /10-SClaptop2ACamPortfolioAndamComputer.scn [p. 323]
- 48 acscups /dataprop /11-SCworkstation2ACamPortfolioAndamComputer.scn [p. 324]
- 49 acscups /dataprop /12-SCpc2ACamAndamComputer.scn [p. 324]
- 50 acscups /dataprop /13-SCserver2ACamAndamComputer.scn [p. 324]
- 51 acscups /dataprop /14-SCphone2ACamPortfolioAndamPhone.scn [p. 324]
- 52 sms /sms1ac35 /smsac.scn [p. 324]
- 53 sms /sms1ac35 /smsact.scn [p. 325]
- 54 sms /sms2ac35 /smsac.scn [p. 325]
- 55 sms /sms2ac35 /smsact.scn [p. 325]
- 56 sms /sms2ac41 /smsac.scn [p. 326]
- 57 tim /tim3ac35 /timac.scn [p. 326]
- 58 uagw /gw3ac36 /gwac.scn [p. 328]

InfraTools Desktop Discoveryコネクタ

ベースコネクタ

• 生成 (ソース)

本節ではInfraTools Desktop Discoveryコネクタについて説明します。

InfraTools Desktop Discoveryコネクタの互換性

コネクタは、InfraTools Desktop Discoveryの5.1xと6.xxバージョンと互換性があります。

InfraTools Desktop Discoveryのクライアント部分 (「mifgen.dll」ファイル) は、Connect-Itと一緒に自動的にインストールされます。

InfraTools Desktop Discovery設定に関する注意事項

FSFファイル

FSFファイルを別の言語（ポーランド語、スペイン語、など）で読み込むためには、Connect-Itがインストールされているコンピュータに、使用する言語のコードページをインストールしなければなりません。

SAIファイル

Saiは、Software Applications Index（ソフトウェアアプリケーションのインデックス）の略です。このインデックスは、Desktop Discoveryスキャナがコンピュータ上で検索し識別するアプリケーションを、リストにして記述しています。

Connect-It付属のデータキットには、「master.sai」「french.sai」と「german.sai」ファイルが含まれています（インストールするConnect-Itの言語によります）。SAIファイルは、[Connect-Itインストール先ファイル]/datakit/iddにあります。

Desktop DiscoveryコネクタのCDTファイル

Desktop Discoveryはコンピュータにインベントリ用スキャンを実行します。スキャンにはコンピュータ自体（Hardware）、コンピュータにインストールされたソフトウェア（Software）とコンピュータに関連する資産とユーザ（UserAndAssets）についての情報が含まれます。

これらの情報の一部（SMBIOSとDMIの項目）はスキャンされるコンピュータに左右されます。例えばCompaqコンピュータでは、「Compaq Monitor Information」というDMI項目があります。

その他の情報はユーザの設定により変化します。ユーザは、Desktop Discoveryのスキャンをカスタマイズし、スキャナ生成プログラム（Scanner Generator）のアンケートAsset Questionnaireに入力される情報を指定することができます。カスタマイズ情報は【UserAndAssets】構造体のフィールドに関連しています。スキャン結果はFSFファイル内に保存されます。

いずれの場合でも、FSFファイルを処理し、これらのファイルから生成されるドキュメントタイプを決定する必要があります。このドキュメントタイプはCDT（Connect-It document typeの略）ファイル内に保存されます。Desktop Discoveryコネクタの設定時に、ユーザはこのファイルの名前を入力します。

学習モード（以下参照）では、コネクタはFSFファイルの内容を読み取り、【UserAndAssets】構造体の要素を作成します。新規要素がある場合は、コネクタは「user.cdt」ファイル内の【Hardware】構造体に新規要素を追加します。

Desktop Discoveryコネクタの設定内のチェックボックスで、学習モードの使用の有無を選択できます。

• 学習モードを使用する

学習モードを使用すると、コネクタはFSFファイル内にあるIT資産やスキャナ設定などの情報を検索します。この情報に応じて、コネクタは「user.cdt」ファイルを変更します。しかしこの時シナリオが起動されても、Desktop Discoveryコネクタに生成されるドキュメントに現れる新規要素は、シナリオ内にはすぐには組み込まれません。コネクタはこの時点で警告メッセージを発信します。新規要素を含むためには、シナリオを停止し、Desktop Discoveryコネクタを閉めてから再開し、学習モードを無効にして、更に必要ならば新規要素をマップしてから、シナリオを再起動する必要があります。

• 学習モードを使用しない

FSFファイル内に起きた変更事項は、「user.cdt」ファイル内に追加されません。その結果、新規要素を含むドキュメントはDesktop Discoveryコネクタに無視されます。新規要素を追加するには、シナリオを停止し、学習モードを使用する必要があります。

ユーザが、複数のコンピュータグループやネットワークに応じて別々のスキャンを設定する場合、必要な分だけCDTファイルを使用することができます(コネクタにつき1つのファイル)。

Connect-Itデータキットの「iddstd.cdt」ファイルは、Desktop Discoveryスキャナ生成プログラム (Scanner Generator) により生成されたデフォルトのスキャナです。

InfraTools Desktop Discoveryコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 3.3. InfraTools Desktop Discoveryコネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]
FSFファイル [p. 70]
FSFファイルから抽出するデータ [p. 71]
FSFデータをフィルタで抽出する [p. 73]
処理後のアクションを定義する [p. 74]
スケジュールのポイントを設定する [p. 32]
ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する


このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

FSFファイル


このページでは、データソースと学習モードのパラメータを選択できます。



【FSFファイルフォルダ】：「.fsf」ファイルが保存されている1つまたは複数のフォルダを指定します。

FSFファイルが保存されているフォルダを指定するには、

- 1  をクリックします。
- 2 コンピュータでのフォルダの完全パスを入力します。

フォルダのパスを変更するには、パスを直接ダブルクリックし、新規のパスを入力します。

フォルダを削除するには、フォルダに対応する行を選択してから  をクリックします。

 と  ボタンを使うと、フォルダの順番を変更できます。

【SAIファイルフォルダ】：SAIファイルが保存されているフォルダを指定します。

【ユーザ追加項目ファイル】：ユーザが追加する特定の項目が保存されている「.cdt」ファイルに名前をつけます。

警告:

FSFファイルを含むフォルダのパス

コンピュータのネットワークドライブの文字に関連付けられたフォルダ内に位置する、フォルダのパスを指定する場合は、パスにネットワークドライブの文字を入力せずに、ドライブの文字に関連付けられたフォルダを入力します。

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、『ユーザガイド』の「統合シナリオをプロダクションモードにする」の章、「Connect-Itサービスの作成」 / 「WindowsでConnect-Itサービスを作成する」 / 「シナリオのパス」を参照してください。

学習モード

学習モードを使用すると、InfraTools Desktop Discoveryコネクタはユーザが追加した以下の項目を考慮に入れます。

- IDDのスキャナの作成時にユーザが設定する【UserAndAssets】構造体の項目
- ユーザのIT資産に応じた【SMBIOS】と【DMI】構造体の項目

上記2つの構造体にある要素は、学習モードでコネクタを使用する際に識別されます。

- ユーザが保存したファイルを含む [StoredFiles] 構造体の項目

学習モードでは、コネクタはユーザが追加した特定の情報を抽出し、FSFファイル进行处理します。この情報に応じて、コネクタはCDTファイルを変更します。学習モードではドキュメントは生成されません。

学習モードは以下の場合に使用します。

- InfraTools Desktop Discoveryコネクタが一番最初に開かれ使用される時。この後、マッピングの段階でユーザが追加した新規項目を使用できるようになります。
- ユーザがスキャンの内容を変更したため、この新規項目に応じてマッピングを変更する必要がある場合。

学習モードを使用しない場合、InfraTools Desktop Discoveryコネクタに生成されるドキュメント内に新規項目や要素型の変更がある時は、Connect-Itが実行する対処方法を2つのオプションから選択できます。

- ドキュメントを拒否する
- ドキュメントログ内に警告を作成する

学習モードでコネクタを起動すると、Connect-Itログ内に次のエラーメッセージが現れることがあります：「**タイプの不一致が要素 [要素名] (パス [ドキュメントタイプ内の要素のパス] 内) にあります。パスをドキュメントタイプに追加することができません。**」

このメッセージは、学習モード時にFSFのフィールドタイプが不適切に識別される場合に表示されます（例えば、値が6であるフィールドが、**文字列**であるにも関わらずlong**整数**タイプとして識別される場合などです）。

このエラーを訂正するには、

- 1 コネクタを一旦閉じてから、再び開きます。
- 2 問題の原因となったFSF（または前回のセッションで処理された全FSF）で、コネクタを学習モードで再起動します。
フィールドタイプは適切に識別されます。
- 3 Connect-Itログ内にエラーメッセージが現れなくなるまで、この操作を繰り返します。

FSFファイルから抽出するデータ

このページでは、FSFファイルから抽出する要素を選択します。

例：

マッピング内で [Hardware] 構造体の要素が無視されている場合は、 [IT (ハードウェア)] オプションをオフにします。

[格納されたファイル]

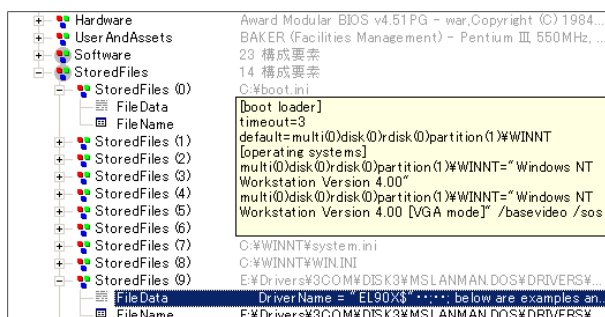
[格納されたファイル] を選択すると、InfraTools Desktop Discoveryコネクタは格納されたファイルの内容を取得します。

格納されているファイルが「.ini」フォーマットの場合、Connect-Itは、[以下の.iniファイルの内容を処理する] フィールドで指定されたファイルをツリー構造で表示します。

例 3.1. 格納されたファイルの例

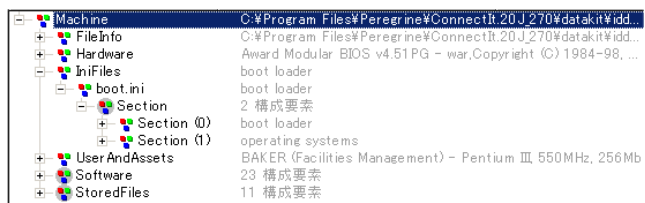
以下は、データキットの「london01.fsf」ファイルに含まれている、格納されたファイル「boot.ini」の例です。

[以下の.iniファイルの内容を処理する] フィールドにファイル名を入力しないと、ファイルは、InfraTools Desktop Discoveryコネクタに生成されるMachineドキュメントのStoredFilesコレクションの構成要素として表示されます。このファイルの内容は、[FileData] メモフィールドに表示されます。



[以下の.iniファイルの内容を処理する] フィールドに「boot.ini」ファイル名を入力すると、このファイルはIniFiles構造体の要素として表示されます。

「boot.ini」構造体のSectionコレクションの構成要素は、初期化ファイルのセクション（boot loader、operating systems）に相当します。



FSFデータをフィルタで抽出する

[ソフトウェア認識フィルタ]

！ 警告:

Connect-Itに表示されるSoftwareコレクションの要素は、InfraTools Desktop Discoveryの【Applications】タブ内のファイルに当たります。

このフィールドではFSFファイルから抽出するソフトウェアのタイプを指定できます。以下の3つのオプションがあります。

- **[何も取得しない]**
このオプションが選択されていると、InfraTools Desktop Discoveryコネクタに生成されるドキュメントのSoftwareコレクションには、ソフトウェアが含まれません。
- **[全ファイルを取得する]**
このオプションが選択されていると、InfraTools Desktop Discoveryコネクタに生成されるドキュメントのSoftwareコレクションには、FSFファイル内にある全てのソフトウェアが含まれます。
- **[次のフィルタを使用する]**
このオプションが選択されていると、InfraTools Desktop Discoveryコネクタに生成されるドキュメントのSoftwareコレクションには、下のフィールドで指定するファイルの拡張子（セミコロンで区切って記入）に一致するソフトウェアが取得されます。

スキャンされたファイルのフィルタ

！ 警告:

Connect-Itに表示されるFileコレクションの要素は、InfraTools Desktop Discoveryの【Software】タブ内のファイルに当たります。

このフィールドでは、FSFファイル内にあるスキャンされたファイル用のフィルタを設定できます。これらのファイルは、InfraTools Desktop Discoveryの【Software】タブ内のファイルに当たります。

以下の3つのオプションがあります。

- **[何も取得しない]**
このオプションが選択されていると、スキャンされたファイルはFSFファイルから抽出されません。
- **[全ファイルを取得する]**

このオプションが選択されていると、スキャンされたファイルは全てFSFファイルから抽出されます。

- **【次のフィルタを使用する】**

このオプションが選択されていると、下のフィールドで指定するファイルの拡張子（セミコロンで区切って記入）に一致するファイルが、FSFファイルから抽出されます。

警告:

コンピュータからスキャンされるファイルは多数にのぼる可能性があります。よって情報を取得するために非常に長い時間がかかることもあります。このため、データを取得する前に、これらの情報が必要不可欠であるかどうか再確認することを強くお勧めします。

処理後のアクションを定義する

このページでは、FSFファイルがプロダクションモードでシナリオに使用された後の、ファイルの管理方法を指定できます。



FSFファイルの処理に成功または失敗した場合、以下の3つのオプションがあります。

- FSFファイルをフォルダに保存する

- ファイルを削除する
- パスを指定したフォルダ内にファイルを移動させる

FSFファイルを削除または移動する場合、InfraTools Desktop Discoveryコネクタは、FSFファイルが正常に処理されたことを示す処理レポートを取り込む必要があります。この処理レポートは、InfraTools Desktop Discoveryコネクタが生成したドキュメント（FSFファイルに相当）を正常に取り込んだ別のコネクタから、InfraTools Desktop Discoveryコネクタに転送されます。

処理レポートの詳細については、『ユーザガイド』の「処理レポート」の章を参照してください。

処理に成功したファイルが移される場合、[履歴を保存する]オプションを使用するとファイルの様々なステータスを保存できます。移動先のフォルダ内では、一番新しいファイルには番号がついていませんが、それより前のファイルには1からNの番号がついています（Nは「生成されるFSFファイルの数-1」）。



警告:

処理後にファイルを削除する

これによりデータを損失する場合があります。例：2つのドキュメントタイプ「Machine」を定義するとします。1番目のドキュメントタイプを起動すると、例えば「PC.FSF」ファイルを処理した後最初のドキュメントを作成します。このドキュメントは適切に処理されたためFSFファイルは削除されます。2番目のドキュメントタイプ「Machine」は、FSFファイルを処理できません。

例：

「jdoe.fsf」、「jdoe_1.fsf」、「jdoe_2.fsf」と「jdoe_3.fsf」ファイルがフォルダ内にある場合、「jdoe.fsf」ファイルが一番新しいファイルで、「jdoe_3.FSF」ファイルが一番古いファイルに当たります。

FSFファイルの基準日とポインタ

[フォルダに保存する]オプションが選択されている場合

シナリオのプロダクションモードでは、[フォルダに保存する]オプションがInfraTools Desktop Discoveryコネクタの設定で選択されていると、ファイルの一番最近の日付が、InfraTools Desktop Discoveryコネクタの基準日になります。

例：

InfraTools Desktop Discoveryコネクタの第1回目の使用時に、次の2つのファイルがあるとします。

- 1月1日の「paris.FSF」ファイル
- 1月15日の「rome.FSF」ファイル

コネクタを次回に起動する際、1月15日という日付が基準日（ポイント）になります。このため、この日付より前に作成されたファイルは**処理されません**。1月15日より前のFSFファイルの処理をシナリオ内で強制するには、スケジュールの編集用ウィンドウで、ポイントのステータスを直接変更する必要があります。

ポイントのステータスを直接手動で変更するには、『ユーザガイド』の「統合シナリオをプロダクションモードにする」の章の「スケジュールの編集」の「ポイントのステータス」を参照してください。

[フォルダから削除する]または[次のフォルダに移す]オプションが選択されている場合

上記オプションのうち1つが選択されていると、FDFのフォルダ内にある全ファイルが日付に関係なく処理されます。この場合、ポイントの概念は存在しません。

[フォルダの相対パスを使用する]

このオプションを選択すると、シナリオのコンポーネントが処理するFSFファイルを含むフォルダの、相対パスを指定できます。例：シナリオのコンポーネントが処理に成功したファイルと、コンポーネントが拒否したファイルを保存するには、「../success」と「../error」、と指定します。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

InfraTools Desktop Discoveryコネクタのルール

ルール（ディレクティブ）はInfraTools Desktop Discoveryコネクタでは使用できません。コネクタは、取り込みまたは生成を行うデータ上にフィルタやその他のアクションを実行せずに、ドキュメントを生成します。

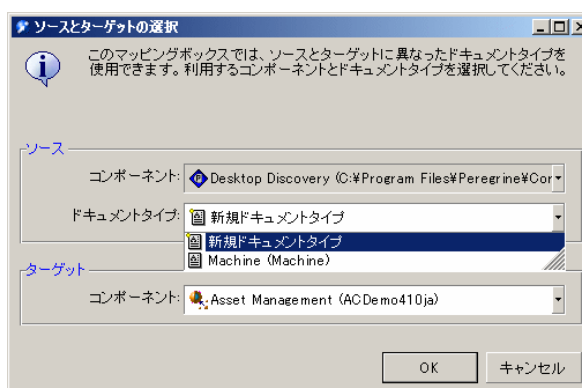
追加情報

本節ではコネクタに関する追加情報を説明します。

複数のドキュメントの生成

ドキュメントタイプ「Machine」に基づいたドキュメントが生成されるたびに、FSFファイルが処理されます。2つの「Machine」ドキュメントタイプが定義されていると、FSFファイルは2度処理されてしまいます。このため、データ処理の性能を向上させるためには、複数のマッピング内で使用する1つのドキュメントタイプを作成する必要があります。

マッピングの作成時に、ウィンドウでは既存のソースドキュメントタイプを選択できるようになっています。コネクタの同一の生成用ドキュメントタイプ用に2つ目のマッピングを作成する場合は、この機能を使用してください。



付属シナリオ - Infratools Desktop Discoveryコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- `idd /ddac36 /ddac.scn` [p. 302]
- `idd /ddac40 /ddac.scn` [p. 302]
- `idd /ddac41 /ddac.scn` [p. 302]
- `idd /ddrsm4 /ddam.scn` [p. 303]
- `idd /ddrsm4 /ddhd.scn` [p. 303]
- `idd /ddsc4 /ddsc.scn` [p. 303]
- `idd /ddrsm4 /ddrsm.scn` [p. 303]

InfraTools Managementコネクタ

ベースコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

本節ではInfraTools Managementコネクタについて説明します。

InfraTools Managementコネクタの互換性

InfraTools Managementコネクタは、InfraToolsアプリケーションのデータベース内のデータを取得できます。また、データベース内にデータを挿入することもできます。（例：InfraTools Remote Controlで使用されるデータベース）

以下のバージョンが使用可能です。

- InfraTools Remote Controlバージョン5.xx










InfraToolsアプリケーションクライアントは、Connect-Itがインストールされているコンピュータにインストールされなければなりません。

InfraTools Managementコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 3.4. InfraTools Managementコネクタの設定

	コネクタに名前を付け、役割を説明する [p. 14]
	接続を選択する [p. 16]
	高度な設定 [p. 79]
	再接続のパラメータを設定する [p. 27]
	サーバとの時間差を指定する [p. 29]
	スケジュールのポイントを設定する [p. 32]
	トランザクションを設定する [p. 34]
	キャッシュを設定する [p. 35]
	ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続を選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続を選択する [p. 16]」の節を参照してください。

高度な設定



警告:

このページは、オプション [高度なコネクタ設定ウィザードを表示する] が選択されていないと表示されません ([編集 / オプション] メニュー)。

[Connect-Itのセッションの終了時に接続を切断する]

InfraTools Managementアプリケーションサーバへの接続は、InfraTools Managementコネクタが開いた時点で自動的に実行されます。コネクタが閉じると接続も切断されます。このオプションを選択すると、Connect-Itのセッションとセッションの間はサーバ接続が自動的に切断されます。これにより、InfraTools Managementコネクタが使用されていない時には、サーバに自由にアクセスできるようになります。

[サーバとの時間差]

このオプションにより、Connect-ItがインストールされたコンピュータとInfraTools Managementアプリケーションサーバ間の時間差を指定できます。InfraTools Managementアプリケーションから読み取られたデータ、またはアプリケーションに書き込まれたデータに関連する日付は、この時間差に応じて調整されます。

[使用するダイナミックライブラリ]

このオプションで、InfraTools Managementアプリケーションに関連するAPIを含むDLLの完全なパスを指定できます。例えばInfraTools Remote Control 5.5では、`C:\Program Files\Peregrine\InfraTools Remote Control\bin32\ftdb55.dll`になります。デフォルトで、コネクタは自動的に使用するDLLの場所を見つけます。

サーバへの再接続用オプション

サーバへの再接続用オプションでは、サーバへの接続に以下の状況で失敗した場合に、Connect-Itが再接続する方法を指定できます。

- InfraTools Managementコネクタを開く時にサーバへの接続に失敗する。
- データの読み取りまたは書き込み中に接続が中断する。

再接続オプション用のフィールドは次の2つです。

- **【基本待ち時間】**

[基本待ち時間]では、Connect-Itが何秒後にInfraTools Managementアプリケーションサーバへの再接続を実行するかを指定します。再接続に失敗すると、Connect-Itは基本待ち時間を2倍にして接続を再試行します。例えば、基本待ち時間が2秒の場合、2回目の再接続は4秒後に実行され、3回目の再接続は8秒後に実行されます。

- **【制限時間】**

この[制限時間]で、Connect-Itがサーバへの再接続を何分(時間)後に中止するかを指定できます。

 **注意:**

再接続に成功すると、

- データの書き込みでは、処理の中断されたドキュメントは再び処理され、セッションは通常通り続行します。
- データの読み取りでは、中断されたセッションは最初から再始動されます。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「再接続のパラメータを設定する [p. 27]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

トランザクションを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

InfraTools Managementコネクタの生成用ルール

コネクタの生成用ルールの入力方法については、「コネクタのルール (ディレクティブ) [p. 39]」章の「生成用ルール [p. 40]」の節を参照してください。

このコネクタのルールとしてWHERE句とORDERBY句を作成すると、ソースデータベースのレコードにフィルタを適用できるようになります。これらの句の詳細については、「コネクタのルール (ディレクティブ) [p. 39]」章の「生成用ルール [p. 40]」節、「WHERE句とORDER BY句 [p. 41]」を参照してください。

AQLシンタックスの詳細については、「AQLクエリ [p. 331]」の章を参照してください。

以下の表はWHERE句やORDERBY句で使用可能な演算子と、オペランドのリストです。

演算子

=
<>
>
<
AND
OR
NOT
LIKE (特殊文字: %)

オペランドのタイプ

オペランドのタイプ	例
数値	1 (シングルクォーテーションマークなし)
文字	'a' (シングルクォーテーションマーク)
日付	'01/12/24' (国際標準)
文字列	'文字列' (シングルクォーテーションマーク)

sysdateフィールド

AQLで作成されるWHERE句では、sysdateフィールドは、getdate()関数に置換されなければなりません。

例：SQLクエリSELECT * from AmAsset where dinstall>sysdate は、AQLクエリSELECT * from AmAsset where dinstall>getdate()に置き換えられなければなりません。

InfraTools Managementコネクタの取り込み用ルール

取り込み用ルールを作成するには、InfraTools Managementコネクタの [整合性] タブ内のオプションを選択し、フィールドに入力します。

この取り込み用ルールは、Asset Managementコネクタのルールと同一です。フィールドの入力方法の詳細については、「Asset Managementコネクタ」の節を参照してください。

付属シナリオ - InfraTools Managementコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- ntsec/htim5/htim.scn [p. 305]

InfraTools Network Discoveryコネクタ

ベースコネクタ

- 生成 (ソース)

本節ではInfraTools Network Discoveryコネクタについて説明します。

InfraTools Network Discoveryコネクタの互換性

InfraTools Network Discoveryコネクタは以下のバージョンと互換性があります。

- Network Discovery 4.0X、4.10と4.20

注意事項






Connect-Itがインストールされているコンピュータは、Network Discoveryサーバにアクセスできなければなりません。

InfraTools Network Discoveryコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 3.5. InfraTools Network Discoveryコネクタの設定

	コネクタに名前を付け、役割を説明する [p. 14]
	接続用パラメータを定義する [p. 83]
	スケジュールのポイントを設定する [p. 32]
	キャッシュを設定する [p. 35]
	ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

接続用パラメータを定義する

[サーバ名]

Network Discoveryサイトのアドレスを入力します。

イベントのURL (高度な設定オプション)

「events」のリストを取得するためにNetwork Discoveryサーバに発信するクエリ用の文字列を入力します。

 **注意:**

多くの場合、[周辺機器のURL] フィールドと [イベントのURL] フィールドのデフォルトのクエリは、InfraTools Network Discoveryのデフォルトのインストールに適しています。不適切な場合は、Network Discovery管理者に連絡してください。

周辺機器のURL (高度な設定オプション)

「devices」のリストを取得するためにNetwork Discoveryサーバに発信するクエリ用の文字列を入力します。

[ログイン]

Network Discoveryアプリケーションとの対話に必要なログインを入力します。
このログインのプロファイルは、シナリオ内でのアクションを実行できるようなプロファイルでなければなりません。

[パスワード]

ログイン用のパスワードを入力します。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

InfraTools Network Discoveryコネクタの生成用ルール

ルールは、WHERE句で作成します。WHERE句でフィルタをかけることにより、生成されるドキュメントから一定の情報を取り出すことができます。

Network Discoveryコネクタには2つの使用可能なドキュメントタイプがあります。

- EVENTタイプ
ネットワーク上に起こり得るあらゆる種類のイベント（コンピュータの故障、ルータの再設定など）を識別します。
- FULL_DEVICEタイプ
ネットワーク上にある機器全体を記述します（コンピュータ、ルータ、バックボーンなど）。

ルール（ディレクティブ）を作成すると、例えば必要な種類のイベントのみを取り出すようにフィルタをかけることができます。

例：

例：Network Discoveryコネクタが、ネットワーク上の不調なコンピュータのみを取り扱う「EVENT」ドキュメントを生成するように、設定するとします。

【category】フィールドには、ネットワーク上の機器に起こるイベントの種類（故障、不調など）が記入されています。

以下のWHERE句を作成します。

```
category = "error"
```

Network DiscoveryコネクタはXMLファイルで既定の数のデータを供給します。クエリがサーバへ送信されるたびに、デフォルトでは20台のコンピュータ（Full-device）または20のイベントが取得されます。使用可能なデータ全てが取得されるまで、新規のクエリがサーバに送信されます。

サーバのドキュメントの作成には時間がかかりますが、かかる時間は1回のクエリで取得するデータ数には必ずしも関係しません。場合によっては、デフォルト値20よりも大きな値を指定することで、コネクタの性能を上げることも可能です。最大値は1000です。

例：

イベントの数を1クエリにつき50に指定する場合（最大値100）、

- 1 EVENTドキュメントタイプのルートノードに移動します。
- 2 以下のクエリを作成します。

```
nrows=50
```

コンピュータ（devices）の数を1クエリにつき50に指定する場合（最大100）、

- 1 FULL_DEVICEドキュメントタイプのルートノードへ移動します。
- 2 以下のクエリを作成します。

```
Limit = 50
```

 **注意:**

フィルタの詳細については、InfraTools Network Discoveryのマニュアルを参照してください。

InfraTools Network Discoveryコネクタの使用可能なドキュメントタイプ

使用可能なドキュメントタイプは、InfraTools Network Discoveryコネクタが取得する情報に相当します。

非スケジュールモード

生成（ドキュメント生成のテスト）モードは、シナリオのテスト用です。デフォルトでは、「EVENT」または「DEVICE」型の20のドキュメントのみがコネクタにより生成されます。この値を変更するには、**[編集/オプション]**メニューの**[ドキュメントの生成テスト中に生成するドキュメントの数]**オプション（表示項目）を使用します。

スケジュールモード

EVENT

イベントのポインタはイベントの識別子です。スケジュールモードでは、ポインタにより、コネクタの最終起動時より後に発生したイベントのみを取得できます。

DEVICE

コネクタの最終起動時より後に変更された「DEVICES」のみを取得することはできません。InfraTools Network Discoveryに識別される全DEVICES（周辺機器）は、コネクタが起動するたびに生成されます。

追加情報 - InfraTools Network Discoveryコネクタ

本節ではコネクタに関する追加情報を説明します。

マッピングスクリプト内で使用されるフィールド

InfraTools Network Discoveryコネクタの生成用ドキュメントタイプと別のコネクタとのマッピングで、シナリオの一部のプログラムやスクリプトは、以下のフィールドを必要とする場合があります。

- EventID
- NMID
- MACAddress


例

- NMIDフィールドは、HTTPの要求をInfraTools Network Discoveryコネクタへ送信し、コンピュータに関する追加情報（DEVICE）を取得することができます。
- EventIDフィールドはスケジューラのポインタを更新します。
スケジュールモードでシナリオを使用する場合、このフィールドは必須です。

InfraTools Network Discoveryコネクタの生成用ドキュメントタイプ内では、これらのフィールドのパスは以下の表の通りです。

要素	フィールドのパス
EVENT	EVENT.EventID
FULL-DEVICE	FULL_DEVICE.NMID
EVENT.FULL_DEVICE	EVENT.Device.MACAddress
FULL_DEVICE.Ports.Port.Connection.FULL_DEVICE	FULL_DEVICE.Ports.Port.Connection.Device.NMID

InfraTools Network Discoveryコネクタの生成用ドキュメントタイプにフィールドを追加するには

- 1 InfraTools Network Discoveryコネクタを含むシナリオを開きます。
- 2 InfraTools Network Discoveryコネクタのを選択します。
- 3 コネクタを開きます（【F4】キー）。
- 4 **【ドキュメントタイプ】**タブを選択します。
- 5 コネクタの生成用ドキュメントタイプを編集します（）。
- 6 必要なフィールドを追加します。

例：EVENT.EventIDフィールド

ドキュメントタイプの作成に関する詳細は、『ユーザガイド』の「統合シナリオのインプリメンテーション」の章の「生成用または取り込み用ドキュメントタイプを編集する」の節を参照してください。

付属シナリオ - InfraTools Network Discoveryコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- [ind /ndac36 /ndac.scn](#) [p. 305]
- [ind /ndac40 /ndac.scn](#) [p. 306]
- [ind /ndac41 /ndac.scn](#) [p. 306]
- [ind /ndrsm4 /ndrsm.scn](#) [p. 306]

Peregrine Desktop Inventory (PDI) コネクタ

ベースコネクタ

• 生成 (ソース)

PDIコネクタは、Peregrine Desktop Inventoryのインベントリで取得されたXMLファイル进行处理します。これらのファイルは、Connect-Itデータキット付属のGZファイル内に圧縮することもできます ([Connect-Itインストール先フォルダ]/datakit/pdi/*.xml)。

PDIコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 3.6. PDIコネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]	
接続のプロトコルを選択する [p. 89]	
FTPサーバ [p. 89]	ローカルファイル/ネットワークファイル [p. 90]
処理後のアクションを定義する [p. 91]	
DTD / XSDを選択する [p. 91]	
キャッシュを設定する [p. 35]	
ピボットドキュメントタイプの使用 [p. 36]	



警告:

ファイルまたはフォルダのパス

コネクタの設定時に、コンピュータのネットワークドライブの文字に関連付けられたフォルダ内に位置する、ファイルまたはフォルダのパスを指定する場合は、パスにネットワークドライブの文字を入力せずに、ドライブの文字に関連付けられたフォルダを入力します。

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、『ユーザガイド』の「統合シナリオをプロダクションモードにする」の章、「Connect-Itサービスの作成」 / 「WindowsでConnect-Itサービスを作成する」 / 「シナリオのパス」を参照してください。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続のプロトコルを選択する

以下のオプションがあります。

- [FTPサーバ]
- [ローカルファイル/ネットワークファイル]

FTPサーバ

FTPサイトにあるテキストファイルの読み取りを選択する場合は、以下の操作を実行します。

- 1 FTP接続パラメータを選択する
- 2 ファイル処理後のアクションを選択する

FTPサーバ接続

このページでは、FTPサーバへの接続用に以下の3つのフィールドに入力します。

- [サーバ]
FTPサーバ名を指定します。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

[ファイルを読み取る]

このオプションを選択する場合は、コンピュータやネットワーク上にあるテキストファイルのパスを指定します。

[フォルダを読み取る]

このオプションを選択する場合は、以下の2フィールドに入力します。

- [パス]
ファイルを含むフォルダのパスを入力します。
- [拡張子]
コネクタが読み取るファイルの拡張子を入力します。

このフィールドのデフォルト値は、「gz」です。GZファイルは、Peregrine Desktop Inventoryアプリケーションのインベントリに対応するXMLファイルを含む圧縮ファイルです。

[サブフォルダを読み取る]

このオプションを選択すると、コネクタは、選択したフォルダのサブフォルダ内のファイルも読み取ります。

注意:

FTPモードでの接続中にエラー「Error: 12015」が発生する可能性があります。このエラーは、同時に多数の接続を実行しているためディレクトリにアクセスできないことを示しています、

処理後のアクションを定義する

PDIコネクタがファイルを読み取った後のアクション用に、3つのオプションがあります。

- [フォルダに保存する]
- [フォルダから削除する]
- [次のフォルダに移す]

3番目のオプションを選択する場合は、処理されたファイルの移動先フォルダのパスを入力します。

コネクタがテキストファイルの処理に成功または失敗した場合用に、上記の3オプションから1つを選択します。

ローカルファイル/ネットワークファイル

ローカルファイルまたはネットワーク上のファイルを読み取る場合、以下の操作を実行します。

- 1 ファイルまたはフォルダを選択する
- 2 ファイル処理後のアクションを選択する

ファイルまたはフォルダを選択する

このページでは、コネクタが読み取るファイルを選択します。以下の2つのオプションがあります。

- [ファイルを読み取る]
- [フォルダを読み取る]

[ファイルを読み取る]

このオプションを選択する場合は、コンピュータやネットワーク上にあるテキストファイルのパスを指定します。

[フォルダを読み取る]

このオプションを選択する場合は、以下の2フィールドに入力します。

- [フォルダ名]
ファイルを含むフォルダのパスを入力します。
- [拡張子]
コネクタが読み取るファイルの拡張子を指定します。このフィールドのデフォルト値は、「gz」です。GZファイルは、Peregrine Desktop Inventoryアプリケーションのインベントリに対応するXMLファイルを含む圧縮ファイルです。

[サブフォルダを読み取る]

このオプションを選択すると、コネクタは、選択したフォルダのサブフォルダ内のファイルも読み取ります。

処理後のアクションを定義する

コネクタがファイルを読み取った後のアクション用に、3つのオプションがあります。

- [フォルダに保存する]
- [フォルダから削除する]
- [次のフォルダに移す]

3番目のオプションを選択する場合は、処理されたファイルの移動先フォルダのパスを入力します。

コネクタがテキストファイルの処理に成功または失敗した場合用に、上記の3オプションから1つを選択します。

シナリオ内の他のコネクタやマッピングボックスに取り込まれるドキュメントに、処理後のアクションが実行されるようにするには、各コネクタやマッピングボックスが生成する処理レポートを使用する必要があります。詳細については、『ユーザガイド』の「**処理レポート**」の章を参照してください。

DTD / XSDを選択する

Peregrine Inventory Desktopインベントリに対応するXMLファイルを処理するために、コネクタはDTDまたはXSDを使用しなければなりません。

Connect-It付属のデフォルトのDTDは、「[Connect-Itのインストール先フォルダ]config/pdi/pdi.dtd」にあります。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

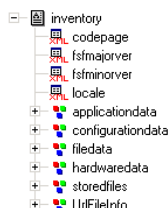
ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

生成用ドキュメントタイプ

PDIコネクタには1つの生成用ドキュメントタイプがあります。ドキュメントタイプ内の要素は以下のスクリーンショットの通りです。

図 3.1. PDIコネクタ - 生成用ドキュメントタイプ



付属シナリオ - PDIコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- pdi/pdiac35/pdiac.scn [p. 329]

ゲートウェイ3.xコネクタ

ベースコネクタ

- 生成 (ソース)

ゲートウェイ3.xコネクタでは、Peregrine Systemsゲートウェイのバージョン3.0とそれ以上(3.x)で取得されたデータを処理できます。

これらのゲートウェイは、ソースアプリケーションのデータをAsset Managementアプリケーション（例：AssetCenter）のデータベース内にインポートするためにMS Accessで開発されたものです。

ゲートウェイコネクタを使用するには、ゲートウェイ3.xのコードを変更またはカスタマイズしていないことが前提となっています。ユーザまたはPS（Professional Services）が実行した変更事項は、標準のゲートウェイコネクタには統合されない可能性があります。このため全データを処理するには、ゲートウェイ3.xコネクタを変更する必要も出てきます。

これらのゲートウェイでは以下のアプリケーションからデータを取得できます。

- D2M NetBird
- LAN Auditor
- Novell Manage Wise
- Novell ZENWorks
- Tally NetCensus
- Tivoli Inventory
- Veritas Winland

汎用ゲートウェイ3.xコネクタは、Peregrine Systems 3.xゲートウェイ全てに接続できます。

ゲートウェイコネクタのODBC互換性

ゲートウェイがサポートするODBCドライバのリストについては、ゲートウェイのマニュアルを参照してください。

重要項目:

ゲートウェイ3.xコネクタの使用は、MS Accessで書かれたゲートウェイ3.xを事前に実行することを前提としています。

ゲートウェイ3.xコネクタは、ゲートウェイ3.xが生成したMS Accessデータベースに接続します。

全ゲートウェイコネクタは、以下のドライバとODBC Administratorバージョン3.520.4403.2、Accessバージョン4.00.4403.02とのテストに成功しています。

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739
- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301

- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500

ゲートウェイコネクタの既知の制約点

ゲートウェイ3.xコネクタを適切に使用するためには、ODBCドライバとODBCアドミニストレータのDLLバージョンが、同一である必要があります。

Peregrine Systems3.xゲートウェイにサポートされているアプリケーションのバージョン

コネクタがサポートするゲートウェイで使用可能なソースアプリケーションのバージョンは、以下の表の通りです。

Peregrine Systems ゲートウェイ	ソースアプリケーション	バージョン	制約点
<ul style="list-style-type: none"> • NBD 3.0.20-4 	D2M NetBird	<ul style="list-style-type: none"> • NetBird 1.2, 1.21, 1.3.1.5, 2.0 	<ul style="list-style-type: none"> • AssetCenter 3.xとのみ使用可能 • 英語、フランス語、日本語版のみ使用可能
<ul style="list-style-type: none"> • LAN 2.7.1 	LANAuditor	<ul style="list-style-type: none"> • LanAuditor 3.1、3.11、3.12、3.13、3.15(CIM) 97Q3 (dBase ISAMドライバが必要) • CIM 98Q2 (ladumpのアップグレードと共に) 	<ul style="list-style-type: none"> • LAN 2.7.1をAssetCenter 3.0xと一緒に使用しないでください。アダプタゲートウェイを使用するか、または新バージョンをお待ちください。
<ul style="list-style-type: none"> • MWS 2.7.1 	Novell ManageWise	<ul style="list-style-type: none"> • バージョン2.7x • Manage wise 2.5 (Btrieve ODBCドライバが必要。未出荷。) 	<ul style="list-style-type: none"> • MWS 2.7.1をAssetCenter 3.0xと一緒に使用しないでください。アダプタゲートウェイを使用してください。

<ul style="list-style-type: none"> ZEN 3.0.20-1 ZEN 3.0.21-2 (日本語) 	Novell ZENWorks	<ul style="list-style-type: none"> Novell ZENworks 2 	<ul style="list-style-type: none"> バージョン3.x AssetCenter 3.xとのみ使用可能 英語、フランス語、日本語版のみ使用可能 Access2000使用時の問題。一部の数値フィールドは、Sybase SQLAnywhere V6 ODBCドライバと共にバイナリとしてインポートされます。 Access 97使用時の問題。一部の数値フィールドは、Sybase SQLAnywhere V6 ODBCドライバと共にテキストとしてインポートされます。
<ul style="list-style-type: none"> PCC 3.0.20-2 	Tally NetCensus	<ul style="list-style-type: none"> NetCensus 2.50、2.80、2.90 	<ul style="list-style-type: none"> AssetCenter 3.xとのみ使用可能 英語、フランス語、日本語版のみ使用可能
<ul style="list-style-type: none"> TME 3.0.20-9 TME 3.0.20-10 (日本語) 	Tivoli Inventory	<ul style="list-style-type: none"> Tivoli inventory 3.1と3.6 	<ul style="list-style-type: none"> AssetCenter 3.xとのみ使用可能 英語、フランス語、日本語版のみ使用可能
<ul style="list-style-type: none"> WLD 2.8.03-1 	Veritas Winland	<ul style="list-style-type: none"> WinLand 4.01.003 	

ゲートウェイコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 3.7. ゲートウェイコネクタの設定

☐	コネクタに名前を付け、役割を説明する [p. 14]
☐	接続タイプを選択する [p. 15]
☐	接続を選択する [p. 16]
☐	高度な設定 [p. 24]
☐	再接続のパラメータを設定する [p. 27]
☐	サーバとの時間差を指定する [p. 29]
☐	スケジュールのポイントを設定する [p. 32]
☐	キャッシュを設定する [p. 35]
☐	ピボットドキュメントタイプの使用 [p. 36]

前提条件

コネクタの設定の前に、データベースへのODBC接続が、ODBCアドミニストレータ内で宣言されていなければなりません。

有効な宣言は、ODBC接続用の適切なログインとパスワードを使用しなければなりません。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続タイプを選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続タイプを選択する [p. 15]」の節を参照してください。

接続を選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続を選択する [p. 16]」の節を参照してください。

高度な設定

このページの使用方法については、「コネクタの設定 [p. 13]」章の「高度な設定 [p. 24]」の節を参照してください。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

ゲートウェイ3.xコネクタの生成用ルール

コネクタの生成用ルールの入力方法については、「コネクタのルール (ディレクティブ) [p. 39]」章の「生成用ルール [p. 40]」の節を参照してください。

このコネクタのルールとしてWHERE句とORDERBY句を作成すると、ソースデータベースのレコードにフィルタを適用できるようになります。これらの句の詳細については、「コネクタのルール (ディレクティブ) [p. 39]」章の「生成用ルール [p. 40]」節、「WHERE句とORDER BY句 [p. 41]」を参照してください。

ゲートウェイ3.xコネクタの使用可能なドキュメントタイプ

ゲートウェイ3.xコネクタには、以下の使用可能なドキュメントタイプがあります。

- outAsset
- outAssetFeatVal
- outProductFeatVal
- outCategoryFeatVal
- outSoftware
- outSoftware
- outLocation
- outDeletedAsset
- outRelConnection
- outSupplier
- outHistorics

これらのドキュメントタイプでは、AssetCenterの主要テーブルへマッピングを作成できます。

付属シナリオ - ゲートウェイ3.xコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- `uagw/gw3ac36/gwac.scn` [p. 328]

ServiceCenterコネクタ

ベースコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

本節ではServiceCenterコネクタについて説明します。

ServiceCenterコネクタの互換性

ServiceCenterコネクタは、ServiceCenterデータベースでのデータの取得または挿入を実行できます。

コネクタは以下のバージョンと互換性があります。

- ServiceCenter 3.0x。「bin32/sc3」ディレクトリにある、Connect-It付属のDLL（例えばWindowsバージョンでは「sccl32.dll」ファイル）を使用。
- ServiceCenter 4.0x。「bin32/sc4」ディレクトリにある、Connect-It付属のDLL（例えばWindowsバージョンでは「sccl32.dll」ファイル）を使用。

ServiceCenterの設定に関する注意事項

入力イベントの作成

ServiceCenter以外のコネクタはドキュメントを直接取り込むことができますが、ServiceCenterコネクタでは不可能です。まずServiceCenterで、対応する入力イベントが作成されなければなりません。ドキュメントを取り込む際に、このイベントはイベントキューに格納されます。次にServiceCenterの内部マッピングは、このイベントの要素をServiceCenterテーブルのフィールドにマップします。

例：

デフォルトでは、コンピュータにインストールされているソフトウェアの説明に対応する入力イベントはありません。

ServiceCenterコネクタがソフトウェアの情報を使用できるようにするには、以下の手順に従います。

- 1 ソフトウェア用の同期の入力イベントを作成します（付属のシナリオでは「pcsoftware」の名前がついています）。
 - 2 入力イベントのフィールドと【pcsoftware】テーブルのフィールドをマップします。
-

Connect-It付属のデータキットでは、上記の2つのアクションをServiceCenterで自動的に実行するファイル（evsoft.unl）をインポートできます。

ServiceCenterの各付属シナリオ用のUNLファイルは、ペレグリンシステムズの顧客サポートから入手可能です。

これらのファイルをインポートする方法については、ServiceCenterのマニュアル内の、テキストファイルのインポート方法の節を参照してください。

入力イベントの同期をとる

ServiceCenter内の入力イベントは同期式または非同期式に処理されています。非同期モードでは、ServiceCenter用スケジュールが、イベントキューに一時的に保存されているイベントを処理する時期を決定します。同期モードでは、イベントはServiceCenterイベントキューに挿入されるとすぐに処理されます。

シナリオのテスト中に、シナリオが機能するかどうかすぐに確認する場合は、同期モードを選択します。

Connect-Itは同期モードで機能するため、非同期モードで処理すると、ServiceCenterでデータが適切に処理されたかどうかを即時に確認することができなくなります。この場合、ドキュメント処理が待機中であることを知らせる警告メッセージが表示されます。

同期モードでは、ドキュメント処理の成功または失敗のメッセージが表示されます。

ServiceCenterコネクタのUNIXでの使用

ServiceCenterコネクタをUNIXで使用するには、









- 1 2つのテキストファイルを作成します。
 - sc.ini
 - sc.log
- 2 log=sc.msggという行を「sc.ini」ファイルに書きます。
- 3 2つのファイルを[Connect-Itインストール先フォルダ]/binに置きます。

ServiceCenterコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 3.8. ServiceCenterコネクタの設定

	コネクタに名前を付け、役割を説明する [p. 14]
	接続を設定する [p. 101]
	高度な設定 [p. 101]
	再接続のパラメータを設定する [p. 27]
	サーバとの時間差を指定する [p. 29]
	スケジュールのポイントを設定する [p. 32]
	キャッシュを設定する [p. 35]
	ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続を設定する

【サーバ名】：以下のように名前を入力します。

例：

[サーバ名].[ServiceCenterクライアントサーバのポート番号]

SC.peregrine.com.12670

 注意:

インストール時に、ServiceCenterのクライアントサーバのポート番号のデフォルト値は、「12670」になります。

【ログイン】

ServiceCenterサーバの接続用ログインを入力します。このログインのプロファイルは、シナリオ内でのアクション（データの読み取りや入力イベントの送信）を実行できるようなプロファイルでなければなりません。

【パスワード】

ログイン用のパスワードを入力します。

【テスト】

このボタンを押すと接続をテストできます。

接続をテストするには、

- 接続用パラメータを入力します。
- 【テスト】をクリックします。

【接続のテスト】ウィンドウが開き、接続の成功または失敗を表示します。接続に失敗すると、失敗の原因が表示されます。

- 【閉じる】をクリックして設定ウィザードへ戻ります。

高度な設定

このページでは、【ServiceCenterサーバからバージョン番号を取得する】オプションを選択できます。

このオプションを使用しない場合、【サーババージョン】フィールドにバージョン番号を入力します。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

ServiceCenterコネクタの生成用ルール

ルールはWHERE句とORDERBY句で作成します。これにより、生成されるドキュメントにフィルタをかけることができます。

以下の表はWHERE句やORDERBY句で使用可能な演算子と、オペランドのリストです。

演算子

=

≠

>

<

AND

OR

NOT

#

オペランドのタイプ	例
数値	1 (シングルクォーテーションマークなし)
文字	"a" (ダブルクォーテーションマーク)
日付	'2002-01-10' (国際標準)
文字列	"文字列" (ダブルクォーテーションマーク)

例：

ServiceCenterのコンピュータのテーブルに当たる「devicepc」ドキュメントタイプで、コネクタが、以下の条件を満たすCompaqコンピュータのみを含むドキュメントを生成するように設定するとします。

- 2001年1月1日から使用されているもの
- Network Discoveryに監視されているネットワーク上にあるもの

以下のWHERE句を作成します。

例：

```
vendor = 'Compaq' and Instal.date > '2000/01/01' and Ind.removed = 0
```

ORDERBY句では、ドキュメントのどのフィールドを昇順での並べ替え（アルファベットまたは数値）に使用するかを指定します。フィールドはカンマで区切ります。例：「Email」生成用ドキュメントタイプでは次のORDERBY句を作成すると、先ず受信者で、次に送信者で並べ替えられます。user.to, user.from

追加情報

本節ではコネクタに関する追加情報を説明します。

ServiceCenterコネクタが適切に処理したイベントを削除する

ServiceCenterコネクタが適切に処理したイベントを削除すると、ServiceCenterのキューを削除できます。

例：ServiceCenterコネクタは出力イベントを処理し、Eメールコネクタ（送信）へ転送します。

これらのイベントを削除するには、各コネクタが作成する処理レポートを使用する必要があります。

処理レポートの詳細については、『ユーザガイド』の「**処理レポート**」の章を参照してください。

ServiceCenterと共に処理レポートを使用する例

この例を模倣するには、Peregrine Systemsのサポートから以下の3つのファイルを手に入れる必要があります。

- delevreg.unl
- delevmap.unl
- delete.scn

UNLファイル

UNLファイルは入力イベントConnectItDelを作成します。

ServiceCenter内にUNLファイルを読み込むには、

- 1 管理者としてServiceCenterのセッションを開始します。
- 2 データベースマネージャを開きます。
- 3 **[オプション/インポート/読み込み]**を選択します。
- 4 以下の2ファイルを読み込みます。
 - delevreg.unl
 - delevmap.unl

2つのファイルはConnectItDel入力イベントを作成します。このイベントによりServiceCenterコネクタは、キューにあるPMOタイプの出力イベントをevsysseqフィールドを使って識別し、出力イベントを削除するようになります。

ServiceCenter内にPMOタイプの出力イベントを作成するには、

- 1 管理者としてServiceCenterのセッションを開始します。
- 2 以下のアクションを実行して、eventoutキュー内にイベントを作成します。
 - **プログラムユーティリティ**タブを選択します。
 - **イベントサービス**をクリックします。
 - **キュー**タブを選択します。
 - **出力イベント**をクリックします。
 - PMOタイプのイベントを作成します。
 - **[追加]**をクリックします。

処理レポート

「delete.scn」シナリオは以下の内容を使用します。

- PMOタイプの出力イベントを処理するServiceCenterソースコネクタ
- ServiceCenterコネクタが生成するドキュメントを変換するマッピングボックス
- ヘルプデスクチケットのテーブル (amTicket) 内にレコードを作成するAsset Managementターゲットコネクタ

- Asset Managementコネクタが生成する処理レポートを取り込むマッピングボックス

Asset Managementコネクタが取り込むドキュメントが適切に処理されると、処理レポートは2番目のマッピングボックスを経由してServiceCenterへ転送されます。これにより、キューにあるPMOタイプのイベントの削除が可能になります。

Asset Managementコネクタが生成する処理レポートとConnectItDelドキュメントタイプ間のマッピングは、イベントの削除を可能にします。

Asset Managementコネクタの処理レポートとServiceCenterコネクタ間のマッピングは、以下の表の通りです。

表 3.9. Asset Managementコネクタの処理レポートと、ServiceCenterコネクタのConnectItDelドキュメントタイプ間のマッピング

ConnectItDelドキュメント 処理レポートの要素またはスクリプト タイプの要素

evsysseq	[\$ParentDoc\$.SParentDoc\$.EventFormatFields.evsysseq]
----------	---

コメント：このスクリプトは固有の識別子（evsysseqフィールド）を取得します。この識別子は、ServiceCenterにキュー内のイベントを削除するように指示します。

\$ParentDoc\$変数の詳細については、『ユーザガイド』の「処理レポート」の章の「付属シナリオでの処理レポートの使用」の節の「\$ParentDoc\$変数の使用」を参照してください。

evtype	"pmo"
--------	-------

コメント：このマッピングは、イベントタイプを使用して整合性チェックをセキュリティで保護します。この例では、イベントタイプはpmoです。

ServiceCenterデータベース内に添付ファイルを挿入する

ServiceCenterコネクタに取り込まれたドキュメントが、ターゲットデータベースに添付ファイルを挿入できるようにするには、ドキュメントの基になるドキュメントタイプを変更する必要があります（ドキュメントタイプは入力イベントに当たります）。

このドキュメントタイプを変更するには、「[Connect-Itのインストール先フォルダ]/config/sc」サブフォルダにある「scdb.cfg」ファイルを編集する必要があります。

ドキュメントタイプが添付ファイルの処理をサポートできるようにするには、以下の操作を実行します。

- 添付ファイルに関連するコレクションをドキュメントタイプに追加する
- SYSBLOBファイルに格納されている添付ファイルをターゲットレコードにリンクするパラメータを初期化する

以下の2つの典型的な例があります。

- 1 添付ファイルはイベントに関連付けられている
この場合、ServiceCenterのイベントマネージャは、添付ファイルをターゲットレコードへ関連付けます。
- 2 添付ファイルはSYSBLOBファイルに直接挿入されている

「scdb.cfg」ファイルの変更

この変更内容は、添付ファイルの2種類の処理方法に共通しています。

次節では、pmoドキュメントタイプ（ヘルプデスクチケットの作成）が、添付ファイルをサポートするためにどのように変更されているかを、コードの抜粋と共に説明します。添付ファイルの例：ヘルプデスクチケットに添付されたスクリーンショット

手順1：イベントに対応するドキュメントタイプ（構造）の宣言

以下のコードはpmoドキュメントタイプの宣言に当たります。

```
{ STRUCT pmo
  NODETYPE = EVENT
  [...]
}
```

ドキュメントタイプが、他のドキュメントタイプと同じ動作をしないことを指定するには、以下の方法でAllTables構造内で例外として宣言します。

```
{ STRUCT AllTables
  Exception = $(LINK_TABLES), pmo
  { ATTRIBUTE AllFields
  }
}
```

別の方法では、添付ファイルをどのイベント用に処理するかを列挙したリストを含む変数を作成します。次に、例外の宣言でこの変数を参照します。

```
#define EVENT_ATTACHMENT pmo
{ STRUCT AllTables
  Exception = $(LINK_TABLES), $(EVENT_ATTACHMENT)
  { ATTRIBUTE AllFields
  }
}
```

手順2：添付ファイルに対応するコレクションの追加

attachmentsコレクションをドキュメントタイプの宣言に追加すると、ドキュメントタイプはこれらの添付ファイルを処理できるようになります。このコレクションには以下の要素が含まれていなければなりません。

- 添付ファイルの名前に当たるnameフィールド（例：sc.ini）
この必須フィールドにはファイルのパスが含まれてはなりません。
- 添付ファイルのバイナリ内容に当たるblobフィールド

```
{ STRUCT pmo
  NODETYPE = EVENT
  { ARRAY attachments
    CIRCULAR = ATTACHMENT_TEMPLATE_EVENT
  [...]
}
```

CIRCULARプロパティを使うと、nameフィールドとblobフィールドをattachmentsコレクションへ追加するテンプレートを読み込むことができます。

```
{ STRUCT ATTACHMENT_TEMPLATE_EVENT
  MODEOUT = 0
  MODEIN = 0
  { STRING name
    MANDATORY = 1
  }
  { BLOB attachment
    MANDATORY = 1
  }
}
```

手順3：添付ファイルの設定

```
{ STRUCT pmo
  { ARRAY attachments

  [...]

  // Not displayed for the output event pmo
  MODEOUT      = 0
  NODETYPE     = BLOB

  BLOBTYPE     = 5
  BLOBFORMAT   = SC
  APP          = problem
  INSERTBLOB   = 1
```

[...]

MODEOUT

値0は、添付ファイルの処理が、取り込みモードでのみ使用可能であることを意味します。値1は、添付ファイルの処理が、生成モードでのみ使用可能であることを意味します。これは生成用と取り込み用の使用可能なドキュメントタイプ用です。

NODETYPE

Attachmentsコレクションが、バイナリタイプ (Connect-ItではBLOBタイプ) のファイルを管理することを示します。この要素は必須であり、値はBLOBでなければなりません。

BLOBTYPE

このパラメータによって、ServiceCenterはBLOBタイプのファイルを分類できるようになります。値5は添付ファイルに当たります。このタイプは、ServiceCenterが挿入時にサポートするBLOBファイルの唯一のタイプです。

BLOBFORMAT

このパラメータを使って、ServiceCenterは、BLOBタイプのファイルの格納フォーマットを指定します。添付ファイルではこのパラメータはSCでなければなりません。

INSERTBLOB

このパラメータは、ターゲットレコード内で添付ファイルを挿入するのみか、または挿入と置換を実行するかを指定します。デフォルト値1は、「挿入のみ」に当たります。

APP

添付ファイルの挿入先のServiceCenterファイルを示します。ファイル名は、SYSBLOBファイルの application フィールド内に含まれます。

典型的な例1 : ServiceCenterで添付ファイルをイベントへ関連付ける

使用中のServiceCenterのバージョンで、添付ファイルをイベントへ関連付けられる場合は、イベントマネージャが、SYSBLOBファイルとターゲットレコード間の関連付けを自動的に実行します。ServiceCenterでこの機能がサポートされている場合は、マネージャでキュー (eventqueue) を閲覧する際に [attachments] タブが表示されます。この場合ターゲットレコードとSYSBLOBファイル間の整合性は、ターゲットレコードの [number] フィールドに基づいています。

以下のコードの抜粋では、ドキュメントタイプが添付ファイルのこの種の処理をサポートするように、pmdドキュメントタイプを変更しています。

```

{ STRUCT pmo
  NODETYPE = EVENT

  { ARRAY attachments

    [...]

    // Insert the attachments into the event
    BLOBRECONCTYPE = EVENT
    APP          = eventin

  }

  { ATTRIBUTE AllFields
  }
}

```

BLOBRECONCTYPE

値EVENTは、添付ファイルがイベントに関連付けられていることを意味します。

APP

値eventinは、SYSBLOBファイル内での、添付ファイルの挿入先テーブルを指定します。

イベントマネージャは次に、eventinの名前を、イベントに関連付けられたレコードのテーブル名に置き換えます。例：pmoイベントの作成時のproblemテーブル

 **注意:**

「添付ファイル - イベント」の関連付けが可能な場合は、この処理方法を必ず使用しなければなりません。

「scdb.cfg」ファイル内のpmo、pmu、pmcとsminイベントを設定すると、この処理方法を使用できます。

典型的な例2 : SYSBLOBファイルとターゲットレコード間の結合を、topicフィールドを使って定義する

ServiceCenterが添付ファイルをイベントへ関連付けられない場合、【topic】フィールドの値が取得される方法を指定しなければなりません。SYSBLOBファイルのこのフィールドは、ターゲットレコードとの結合を定義します。

以下のコンテキストが可能です。

- 1 【topic】フィールドの値は、ServiceCenterコネクタが取り込むドキュメント内に含まれます。

例：イベントICMpc（コンピュータに関するイベント）に対応する取り込み用ドキュメントタイプ内では、【logical.name】フィールドを使用することによって、SYSBLOBファイルとdevicepcファイルのターゲットレコード間のリンクを、ServiceCenterデータベース内に作成できるようになります。

このフィールド値は、イベント内でマップされたフィールドから、以下のシンタックスで直接抽出されます。

```
{ STRUCTURE ICMpc
  [...]

  { STRING logical.name
    MANDATORY = 1
  }

  { ARRAY attachments
    [...]
    BLOBRECONCTYPE = INTERNAL
    PIFLINK      = @{'..logical.name'}
    [...]
  }

  { ATTRIBUTE AllFields
    EXCEPTION = 'logical.name'
  }
}
```

BLOBRECONCTYPE

このパラメータのINTERNAL値は、【topic】フィールドの値が、コネクタに取り込まれるドキュメントのフィールド内に含まれるかどうかを指定します。

PIFLINK

値@{..nom_du_champ}によって、コネクタに取り込まれるドキュメント内のフィールドの値を取得できます。本例では、フィールドのパスは【attachments】コレクションに相対的です。

2つのピリオド(..)は、【logical.name】フィールドを見つけるには1レベル上の階層へ行かなければならないことを意味します。

ファイルのパスのように、2つのピリオド(..)は現在のレベルの上のレベルを意味します。このため、@{....nom_du_champ}では2レベル上へ行くことができます。

 注意:

フィールドに1つのピリオド (.) が含まれる場合は、一重引用符 (!) で囲みます。例: PIFLINK = @{'..logical.name'}

本例では、【logical.name】フィールドは必須になります。これは、添付ファイルの挿入にこのフィールド値が必要になるからです。

- 2 【topic】フィールドの値は、ServiceCenterデータベースへ送信される要求を使って取得されます。

この場合、【topic】フィールドではないフィールドが、レコードの整合性チェックのために使用されます。例: 【contact.name】フィールドは、【topic】フィールドに値を入力するために必要ですが、【first.name】フィールドと【last.name】フィールドのみが、取り込まれるドキュメント内に存在しません。

```
{ STRUCTURE eventcontacts
  [...]

  { STRING last.name
    MANDATORY = 1
  }

  { ARRAY attachments
    [...]
    BLOBRECONCTYPE = QUERY
    TOPICFIELD   = contact.name
    PIFLINK      = last.name = @{'..last.name'}
    [...]
  }

  { ATTRIBUTE AllFields
    EXCEPTION = last.name
  }
}
```

BLOBRECONCTYPE

値QUERYは、【topic】フィールドの値が、ServiceCenterデータベースで実行された要求を使って取得されることを表します。

TOPICFIELD

この例では、値【contact.name】は、【topic】フィールドに値を入力するためにこのフィールドが使用されたことを意味します。

PIFLINK

この例では、値@{..'last.name'}は、ServiceCenterデータベースへ送信されるWHERE句内で使用される値を指定します。

 **注意:**

この処理モードでは、イベントが同期で処理されなければなりません。

- 3 ドキュメントがServiceCenterイベントマネージャに処理された後に、【topic】フィールドの値を取得することも可能です。

重要: この処理モードは、使用中のServiceCenterバージョンがイベントと添付ファイルの関連付けをサポートしない場合は、使用不可能です。

この値は、イベントの【evid】フィールドの読み取り時に取得されます。このフィールドのタイトルはProblem IDまたはIncident IDで、使用するServiceCenterのバージョンにより変化します。

この処理モードは、作業指示の作成 (eventpmo) またはヘルプデスクのコール (event smin) に対応します。

```
{ STRUCT pmo
  [...]

  { ARRAY attachments
    [...]

    BLOBRECONCTYPE = EVID

    [...]
  }

  [...]
}
```

BLOBRECONCTYPE

値EVIDは、フィールド【topic】の値が、イベントのフィールド【evid】を使って取得されることを意味します。

パラメータPIFLINKは必要ではありません。

 **注意:**

この処理モードでは、イベントが同期で処理されなければなりません。

非同期モードのイベントへの警告

非同期モードで設定されるイベントが、ServiceCenter内にデータを挿入するために使用される場合、Connect-Itはイベントが適切に処理されたかどうかを確認できません。

この場合以下の警告メッセージが表示されます。

- セッションの開始時に、Connect-Itログに警告メッセージが表示される
- 処理されるドキュメントごとに、ドキュメントログに警告メッセージが表示される

警告メッセージが表示されないようにするには、

- 1 [編集/オプション]を選択します。
- 2 [コネクタ]項目の[ServiceCenter]ノードを開きます。
- 3 [非同期モードで使用されているイベント用に警告を表示する]オプションを「いいえ」に設定します。
- 4 [OK]をクリックします。

付属シナリオ - ServiceCenterコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- `idd/iddsc4/iddsc.scn` [p. 303]
- `scac/sc3ac36/scac.scn` [p. 314]
- `scac/sc3ac40/scac.scn` [p. 320]
- `scac/sc3ac41/scac.scn` [p. 321]
- `scac/sc4ac40/scac.scn` [p. 321]
- `scac/sc4ac41/scac.scn` [p. 321]
- `scauto/scacfg/scacfg.scn` [p. 321]
- `tim/tim4sc4/timsc.scn` [p. 327]

4 | アプリケーションコネクタ

アプリケーションコネクタは、外部アプリケーションや特定のオペレーティングシステムのデータを処理するコネクタです。

Lotus Notesコネクタ

オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

Lotus Notesコネクタでは、以下の操作が可能です。

- 生成モードで、Lotus Notesデータベースのデータを外部アプリケーション（例：ServiceCenterデータベース）に挿入する。
- 取り込みモードで、外部アプリケーションのデータをLotus Notesデータベースに挿入する。

Lotus Notesコネクタの互換性

Lotus Notesコネクタは、Lotus Notesのバージョン5.xとのテストに成功していません。

Lotus Notesコネクタは、Lotus Notesのバージョン4.6の一部とテストされていません。日本語版はサポートされていません。

 **注意:**

Lotus Notesコネクタを使用するには、Connect-Itがインストールされているコンピュータに、クライアントLotus Notesがインストールされ正しく設定されていなければなりません。

Lotus Notesコネクタの既知の制約点

Lotus Notesコネクタが処理できるのは、以下のタイプのフィールドのみです。

- テキスト
- テキストリスト
- 数値
- 日付
- Lotus Notesリッチテキスト形式





この形式では、Lotus Notes文書に特有のフォーマットオプションを保存できません。この場合Notes文書へのリンクや添付ファイルは全て失われ、ページ設定のみが保存されます。

Lotus Notesコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 4.1. Lotus Notesコネクタの設定

	コネクタに名前を付け、役割を説明する [p. 14]
	接続用パラメータを定義する [p. 117]
	スケジュールのポイントを設定する [p. 32]
	ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続用パラメータを定義する

[基本設定] ウィザードの2ページ目では、Lotus Notes接続を設定します。このページの4つのフィールドでは、次の内容を指定できます。

- Lotus Notes IDファイル（拡張子は「.id」）
- パスワード
- Lotus Notesサーバ
- Lotus Notesデータベース

[Lotus Notes IDファイル]

Lotus Notesデータベースへの接続に使用する「.id」ファイルを指定します。このファイルにはNotes IDとパスワードが含まれています（例：C:\Program Files\notes\data\user.id）。このIDファイルは、**[データベース]**フィールド内で選択されたデータベースに実行するアクション（読み取り、書き込み、削除）を許可するIDファイルでなければなりません。

警告:

このファイルは、Connect-ItがインストールされているLotus Notesクライアントコンピュータに使用されているファイルでなければなりません。

[パスワード]

Lotus Notesデータベースへの接続用パスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

「nextpwd.dll」DLLファイルはこのパスワードを記憶できるため、Lotus Notesコネクタを含むシナリオを起動する度に、パスワードを入力する必要はありません。

Lotus Notesコネクタのインストール時に、このDLLはConnect-Itのインストール先フォルダの「bin 32」フォルダ内に置かれます。「notes.ini」ファイルに次の行、EXTMGR_ADDINS=[Connect-Itインストール先フォルダ]bin32\extpwd を追加する必要があります。（例：EXTMGR_ADDINS=C:\Program Files\Peregrine\ConnectIt\Bin32\extpwd）

[サーバ名]

データベースがインストールされているサーバを選択します。原則として、このフィールドは使用中のコンピュータがアクセスできるサーバを表示しますが、希望のサーバ名が表示されない場合もあり得ます。この場合は、サーバアドレスを直接手動で入力します。自分のコンピュータにインストールされているデータベースに接続するには、**[ローカル]**を選択します。

[データベース]

Connect-Itシナリオで使用するLotus Notesデータベースを、[データベース] フィールドのドロップダウンリストから選択します。

- リスト内の内容は選択したサーバ名により変わります。
- [サーバ]フィールドで[ローカル]を選択した場合、リスト内の内容は、Lotus Notesのインストール先フォルダの「data」フォルダ内に含まれているデータベースに一致します。

使用するデータベース名がこのフィールドに表示されない場合は、フィールドに直接入力します。この場合、データベースの**完全なパス**を入力しなければなりません。

[データベースの全フィールドを表示する] オプション

このオプションを使うと、コネクタはLotus Notesデータベースの全フィールドを表示できます。このオプションを選択しないと、フォームで宣言されたフィールドのみが、コネクタの使用可能なドキュメントタイプ内に現れます。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

Lotus Notesコネクタの生成用ルール

Lotus Notesコネクタの生成用ルールでは、Lotus Notesデータベース内のデータに直接フィルタをかけることができます。Lotus Notes内のデータの並べ替えは、データを含むNotes文書の変更日に応じて自動的に実行されます。

コネクタが生成するドキュメント用のデータにフィルタをかけるには、WHERE句を作成する必要があります。

WHERE句

WHERE句では、Lotus Notesデータベースのレコードにフィルタをかけることができます。例えば、2000年1月1日からLotus Notesデータベースに作成されたレコードのみをフィルタで抽出するとします。これには、[レコードの作成日を含むフィールド] >= [1/01/2000] という句を作成すればフィルタを実行できます。

以下の表はWHERE句で使用可能な演算子と、オペランドのリストです。

演算子	アクション
=	等しい
<>	等しくない
>	より大きい
<	より小さい
&	と
	または

オペランドのタイプ	例
数値	1 (シングルクォーテーションマークなし)
文字	"a" (ダブルクォーテーションマーク)
日付	[10/10/00] (地域設定に対応する日付)
文字列	"文字列" (ダブルクォーテーションマーク)

フォーム名

Lotus Notesコネクタの使用可能な各ドキュメントタイプは、Lotus Notesデータベース内の1つのNotes文書に対応しています。また各Notes文書は1つのフォームに関連付けられています。フォームの名前はNotes文書の任意管理項目フィールドになります。

複数のNotes文書が同一のデータを含んでいても、別々のフォームに関連付けられていることもあります (例えば【サプライヤ】フォームと【連絡先】フォームなど)。希望のフォームに対応するConnect-Itドキュメントをコネクタが生成するよう設定するには、【フォーム名】フィールドのドロップダウンリストから、使用するフォームの名前を選択する必要があります。

Lotus Notesコネクタの取り込み用ルール

Lotus Notesコネクタの取り込み用ルールは、Lotus Notesデータベース内のレコードと、コネクタに取り込まれるドキュメント値の整合性を確認します。

【整合性】タブの詳細

【フォーム名】

Lotus Notesコネクタの使用可能な各ドキュメントタイプは、Lotus Notesデータベース内の1つのNotes文書に対応しています。また各Notes文書は1つのフォーム

に関連付けられています。フォームの名前はNotes文書の任意管理項目フィールドになります。

複数のNotes文書が同一のデータを含んでいても、別々のフォームに関連付けられていることもあります（例えば【サブライヤ】フォームと【連絡先】フォームなど）。希望のフォームに対応するConnect-Itドキュメントをコネクタが生成するよう設定するには、【フォーム名】フィールドのドロップダウンリストから、使用するフォームの名前を選択する必要があります。

【フォント】

このフィールドでは、コネクタが書き込むデータのフォーマットで使用される文字のフォントを指定できます。

【識別キー】

 識別キー

このチェックボックスでは、現在選択されている要素が識別キーとして使用されているかどうかを指定できます。

フィールドまたは構造体を識別キーとして指定するには、

- 要素を作業枠内で選択します。
- 【識別キー】チェックボックスをオンにするか、または取り込み用ドキュメントタイプが表示される枠内の透明なキーを、直接クリックします。

【大文字小文字の区別】

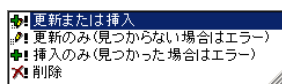
このオプションにより、Connect-Itの整合性チェックは以下のようになります。

- 【大文字小文字の区別】オプションを選択する
このオプションが選択されていると、識別キー値の大文字と小文字は区別して取り扱われます。例：従業員のEメールアドレスを含むフィールドが識別キーとして使用されている場合、'jmartin@company.com'、'JMARTIN@company.com' と 'jMartin@company.com' は、ターゲットのLotus Notesデータベース内で3つの別々のレコードになります。
- 【大文字小文字の区別】オプションを選択しない
このオプションを選択しないと、識別キー値の大文字と小文字は区別されません。例：従業員のEメールアドレスを含むフィールドが識別キーとして使用されている場合、'jmartin@company.com'、'JMARTIN@company.com' と 'jMartin@company.com' は、ターゲットのLotus Notesデータベース内で1つの同じレコードになります。

[キーセット]

→ 複数の識別キーセットを使用する [p. 48]

[整合性チェックのタイプ]



[整合性チェックのタイプ] フィールドでは、各親ノードとドキュメントタイプ間の整合性チェックのタイプを指定できます。

取り込み用ドキュメントタイプの非端末ノードでは、以下の手順に従うと整合性チェックの方法を選択できます。

- 1 取り込み用ドキュメントタイプの非端末ノード（ルートノード、構造体、コレクション）を選択します。
- 2 **[整合性チェックのタイプ]** フィールドでオプションを選択します。

以下のオプションがあります。

- **[更新または挿入]**
この場合、コネクタに取り込まれるドキュメントのデータは、ターゲットのLotus Notesデータベース内のレコードを更新できます。またはデータベース内にレコードを挿入できます。
- **[更新のみ]**
この場合、コネクタに取り込まれるドキュメントのデータは、ターゲットのLotus Notesデータベース内の既存のレコードを更新できます。
- **[挿入のみ]**
この場合、コネクタに取り込まれるドキュメントのデータは、ターゲットのLotus Notesデータベース内に新規レコードを挿入できます。
- **[削除]**
この場合、コネクタに取り込まれるドキュメントのデータは、ターゲットのLotus Notesデータベース内の既存のレコードを削除できます。

Lotus Notesコネクタに関する追加情報

本節ではコネクタに関する追加情報を説明します。

バイナリLotus Notesフィールド

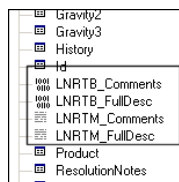
Lotus Notesデータベースには、バイナリ型のフィールドを含むドキュメントがあります。バイナリフィールドは、テキストの他、色やフォントなどの書式情報などが含まれています。

Lotus Notesコネクタの使用可能なドキュメントタイプでは、各バイナリフィールドは2つのフィールドに分かれます。

- 可変長バイナリフィールド
このフィールドは、テキストと書式情報を含みます。このフィールドには LNRTB_ (Lotus Notes Rich Text Binary) という接頭語が付いています。これらのフィールドをLotus Notesバイナリフィールド以外にマップすることはできません (Lotus Notes - Lotus Notesシナリオの場合) 。
- メモフィールド (ロングテキストフィールド)
このフィールドはテキストのみを含みます。このフィールドには、 LNRTM_ (Lotus Notes Rich Text Memo) という接頭語が付いています。マッピングでは、これらのフィールドを他のテキストフィールドにマップすることができます。

注意:

Lotus Notesコネクタの取り込み用ドキュメントタイプでは、可変長バイナリフィールドのみ使用可能です。



スケジュールモードでのデータ処理

Lotus Notesデータベースのローカルコピーを使用すると、2セッション間に変更されたNotes文書はコネクタによって処理されないことになります。

コネクタがNotes文書を処理するように設定するには、スケジュールモードでの2セッション間に、Notesサーバとローカルコピーの同期をとる必要があります。

Notes文書のプロパティには2つの変更日があります。

- 初期変更日 (initially)
- このファイルの変更日 (in this file)

コネクタは初期変更日（サーバと同期をとるたびに更新される日付）を使用するため、Notes文書の編集時に「ファイルの変更日」（in this file）が変更されているだけでは、ファイルは処理されません。

付属シナリオ - Lotus Notesコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- `ldap/ldapnote/hames.scn` [p. 309]

MQSeriesコネクタ

オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

MQSeriesは、多数のハードウェアプラットフォームとソフトウェアプラットフォームへ、固有、非同期で確実なデータ送信を実行するIBMのアプリケーションです。

MQSeriesは、同一コンピュータ上、またはネットワークに分散する複数のコンピュータ間で、異種アプリケーション同士のコミュニケーションを実現するインフラストラクチャです。

MQSeriesは、一般に広く使用されている通信プロトコルを処理できます。また、異種のプロトコルを使用するネットワーク間用にルートを提供します。Connect-Itの統合シナリオでは、MQSeriesはXMLドキュメントを交換します。

MQSeriesのブリッジとゲートウェイで、Lotus Notes、WebブラウザやJavaアプレットなど、多数のアプリケーションシステムや環境へのアクセスが容易になっています。

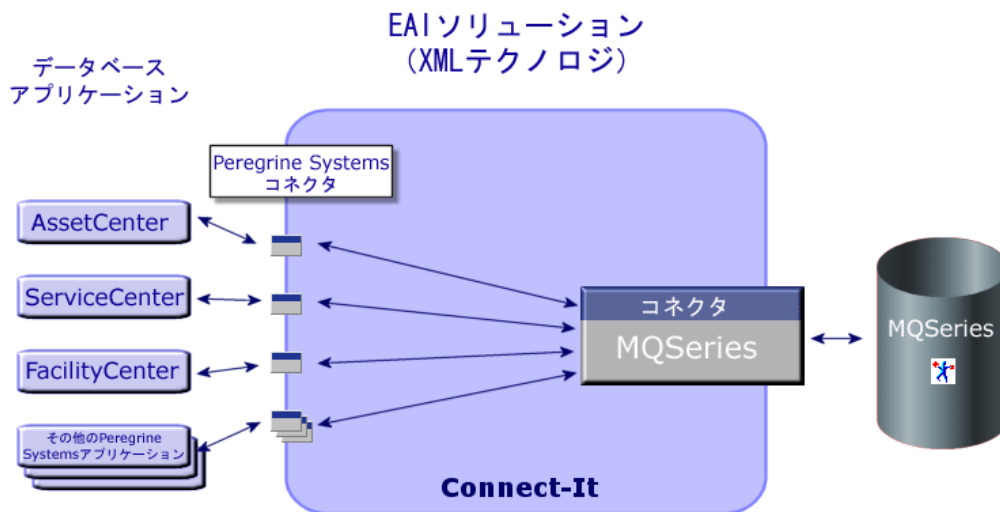
MQSeriesの新機能は、システムやネットワークインフラストラクチャに欠陥がある場合でも、データの送信を保証します。

MQSeries内のデータは、「メッセージ」の形で循環しており、このメッセージ内のデータが異種アプリケーション間で交換されます。

メッセージは、「キュー」と呼ばれるデータ構造内に格納されます。メッセージは、キュー内に置かれたり、アプリケーションの「キューマネージャ（キュー管理プログラム）」によりキューから取り出されたりします。

外部アプリケーションは、XMLドキュメントをMQSeriesキュー内に置きます。適切なキューマネージャに接続されると、MQSeriesコネクタはこのメッセージを読み取り、シナリオ内で指定されたPeregrine Systemsアプリケーション内に送

信します。データは、MQSeriesからPeregrine Systemsアプリケーションへ、またはPeregrine SystemsアプリケーションからMQSeriesへ、の2方向で交換されます。



互換性

MQSeries コネクタは、IBM MQSeriesバージョン5.1と5.2とのテストに成功しています。

MQSeriesコネクタ（読み取り）の設定

MQSeriesコネクタ（読み取り）の基本設定では、コネクタがどのMQSeriesキュー内でデータを読み取るかを設定できます。コネクタは、読み取ったデータをConnect-Itドキュメントに変換します。

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 4.2. MQSeriesコネクタ（読み取り）の設定

■	コネクタに名前を付け、役割を説明する [p. 14]
■	処理モードを選択する [p. 125]
■	キューマネージャへ接続する [p. 125]
■	再接続のパラメータを設定する [p. 27]
■	サーバとの時間差を指定する [p. 29]
■	キューの選択 [p. 127]
■	処理後のアクションを定義する [p. 128]
■	DTD / XSDを選択する [p. 128]
■	スケジュールのポインタを設定する [p. 32]
■	ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

処理モードを選択する

【基本設定】ウィザードの2ページ目では、MQSeriesコネクタの処理モードを選択します。【読み取り】モードを選択します。

キューマネージャへ接続する

このページでは以下の内容を入力します。

- MQSeriesキューマネージャへ接続するためのMQSeriesコネクタのパラメータ
- MQSeriesサーバへ再接続するためのMQSeriesコネクタのパラメータ
- Connect-ItとMQSeriesサーバ間の時間差

キューマネージャへ接続する

コネクタがキューマネージャへ接続できるように、以下のパラメータを指定します。

- 【サーバ名】
このフィールドには、ネットワーク上のMQSeriesサーバのDNS名、またはIPアドレスを入力します。
- 【接続ポート】
キューマネージャのポート番号を指定します。このポートのデフォルト値は「1414」です。
- 【チャンネル名】

キューマネージャが通信パスとして使用する接続チャンネル名を入力します。デフォルト値は「SYSTEM.DEF.SVRCONN」です。

- **【キューマネージャ名】**

キューマネージャ名を入力します。MQSeriesコネクタの付属シナリオを使用する場合、キューマネージャ名はconnect.itです。このフィールドが入力されていない場合、MQSeriesコネクタは、MQSeriesのデフォルトのキューマネージャへ接続します。

- **【MQSeriesサーバはOS/390上で稼動する】**

このオプションを使用すると、OS/390がサポートしない全機能を無効できます。

このオプションを選択すると、以下の機能がコネクタのルール内で使用不可能になります。

取り込み用ルール

自動セグメント化は不可能

GroupIDの指定は不可能

生成用ルール

【グループの全メッセージが使用可能な場合にのみメッセージを取得する】 オプションはオフ

WHERE句ではGroupIDキーワードは作動不可能

コネクタの生成用または取り込み用ドキュメントタイプでは、GroupIDノードがMessageInfo構造体から消えます。

 **注意:**

【拡張接続を使用する】 デフォルトオプションを選択しないと、**【チャンネル名】**と**【キューマネージャ名】**フィールドのみが入力可能になります。

【テスト】ボタン

このボタンを押すと接続をテストできます。

接続をテストするには、

- 1 接続用パラメータを入力します。
- 2 **【テスト】**をクリックします。

【接続のテスト】ウィンドウが開き、接続の成功または失敗を表示します。接続に失敗すると、失敗の原因が表示されます。

- 3 **【閉じる】**をクリックして設定ウィザードへ戻ります。

MQSeriesサーバへの再接続用パラメータ

再接続オプションでは、MQSeriesサーバへの接続が失敗した場合に、MQSeriesコネクタが再接続する方法を指定できます。

再接続オプション用のフィールドは以下の通りです。

- **【基本待ち時間】**

【基本待ち時間】は、コネクタが何秒後にMQSeriesサーバへの再接続を実行するかを指定します。再接続に失敗すると、Connect-Itは基本待ち時間を2倍にして接続を再試行します。例えば、基本待ち時間が2秒の場合、2回目の再接続は4秒後に実行され、3回目の再接続は8秒後に実行されます。

- **【制限時間】**

この【制限時間】で、コネクタがサーバへの再接続を何分（時間）後に中止するかを指定できます。

 **注意:**

再接続に成功すると、中断されたConnect-Itセッションは最初から再始動されません。

【サーバとの時間差】

【サーバとの時間差】フィールドでは、Connect-ItサーバとMQSeriesサーバ間の時間差を指定できます。このフィールドのデフォルト値は「0秒」です。この時間差は、MQSeriesコネクタ用のシナリオをテスト段階で試行してから入力するようにします。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

キューの選択

このページでは、MQSeriesコネクタがどのキュー内でデータを読み取るかを指定します。MQSeriesの付属シナリオを使用する場合、キューの名前は「PEREGRINE.IN」です。

[モデルキュー] オプション

[キュー名] フィールドに入力されたキューがモデルキューである場合は、このオプションを選択します。

使用するダイナミックキュー名のフォーマットを指定します。「*」は、作成されるダイナミックキュー名が固有になるように、キューマネージャにより置換されます。

[モデルキュー] オプションを選択すると、[終了時のオプション] がアクティブになります。以下の3つのオプションがあります。

- 永久ダイナミックキューは保存され、一時ダイナミックキューは破棄されません。
- 空の永久ダイナミックキューは破棄されます。一時ダイナミックキューも破棄されます。
- メッセージは永久ダイナミックキューから消去され、永久ダイナミックキューは破棄されます。一時ダイナミックキューも破棄されます。

処理後のアクションを定義する

このページでは、MQSeriesコネクタがキューのメッセージを読み取った後に、メッセージをどの様に処理するかを指定します。

メッセージの処理に成功または失敗した場合には、以下の3つのオプションがあります。

- メッセージをキュー内に残す。
- メッセージを削除する。
- メッセージをキューに移動させる。編集用テキストゾーンに移動先のキューを入力します。

MQSeriesの付属シナリオでは、処理後のアクション用に以下のオプションがあります。

- コネクタに正常に処理されたメッセージはキューから削除される。
- コネクタに処理されなかったメッセージは、MQSeriesの基本設定時に作成したPEREGRINE.ERROR.INキューへ移される。


シナリオ内の他のコネクタやマッピングボックスに取り込まれるドキュメントに、処理後のアクションが実行されるようにするには、各コネクタやマッピングボックスが生成する処理レポートを使用する必要があります。詳細については、『ユーザガイド』の「処理レポート」の章を参照してください。

DTD / XSDを選択する

XMLファイルを処理するために、MQSeriesコネクタはDTD (Document Type Definition) またはXSD (Extended Schema Definition) を使用する必要があります。このページでは、使用するDTDまたはXSDを [DTD / XSD] フィールドで指定できます。

 **注意:**

学習モード（下記参照）でMQSeriesコネクタを使用する場合、[DTD / XSD] フィールドには空のファイル名を指定します。学習モードでデータ処理が終わった時点で、このファイルの内容が充填されます。このフィールドに記入された名前が既存ファイルの名前と一致すると、Connect-ItはこのDTDファイルに自動的に番号を付けて保存します（例：request_01.dtd）。コネクタが数回学習モードで起動されると、起動の度に作成されるそれぞれのDTDには、作成の順番に従って番号が付けられます。（例：request_01.dtd、request_02.dtd、request_03.dtd。「request_03.dtd」が最も最近に保存されたDTDファイルです。）

 をクリックすると、[**ファイルの場所**] ページが表示されます。このページでは、接続プロトコルの種類に応じてDTDのパス名を指定できます。

ファイルの場所

以下の3つのオプションがあります。

- [**ローカル/ネットワーク**]
Connect-Itがインストールされているコンピュータやネットワーク上の、DTDの完全パス名を入力します。
- [**FTP**]
[**サーバ名**]、[**ログイン**]、[**パスワード**] と [**パス**] フィールドにFTPパラメータを入力します。
[**パス**] フィールドでは、DTDのサーバ上の場所を指定します。例えば、DTDがサーバ「ftp.mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、[**サーバ名**] フィールドに「ftp.mycompany.com」を入力し、[**パス**] フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。
注意：DTDを学習モードでFTPサイトに書き込むことはできません。

 **注意:**

スラッシュ（ / ）はサーバ名の最後には入れませんが、DTDまたはXSDパスの最初に入れます。

- [**HTTP**]
[**サーバ名**]、[**ログイン**]、[**パスワード**] と [**パス**] フィールドにHTTPパラメータを入力します。
[**パス**] フィールドでは、DTDまたはXSDのサーバ上の場所を指定します。例えば、DTDがサーバ「http://mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、[**サーバ名**] フィールドに「http://mycompany.com」を入力し、[**パス**] フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。

注意:

スラッシュ (/) はサーバ名の最後には入れませんが、DTDまたはXSDパスの最初に入れます。

[第1レベルの各コレクションに使用可能なドキュメントタイプを発行する]

このオプションを選択すると、MQSeriesコネクタは、選択したDTDまたはXSD内にある第1レベルのコレクションと同じ数の使用可能なドキュメントタイプを発行します。

例:

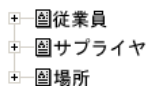
あるDTDのルート要素が「会社」で、第1レベルに3つのコレクション「従業員」、「サプライヤ」と「場所」があるとします。上記のオプションが選択されていないと、コネクタが発行する使用可能なドキュメントタイプのルート要素は「会社」になります。

注意:

DTDの第1レベルのコレクションは、コネクタの使用可能なドキュメントタイプのルート要素のすぐ下にあるコレクションになります。



[第1レベルの各コレクションに使用可能なドキュメントタイプを発行する] オプションが選択されていると、MQSeriesコネクタは、DTDの第1レベルのコレクションごとに使用可能なドキュメントタイプを発行します。この場合DTDのルート要素 (要素「会社」) は、MQSeriesコネクタの使用可能なドキュメントタイプには表示されません。



ルート要素を指定する

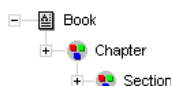
DTDまたはXSDは複数の要素から構成されており、これらの要素は更に別の要素を含んでいます。(例: 文書編集用のDTDでは、複数の「Section」要素が複数の「Chapter」要素に含まれており、更に「Chapter」要素は1つの要素「Book」に含まれています。)他の要素に含まれることがない要素は、ルート要素と呼ばれます。DTDまたはXSDは、1つまたは複数のルート要素から構成されることもあれば、全くルート要素が存在しないこともあります。

ルート要素(つまり、MQSeriesコネクタに発行される使用可能なドキュメントタイプに当たります)を指定するためには、以下の2つのオプションがあります。

[DTD / XSD内に見つかった各ルート要素に使用可能なドキュメントタイプを発行する(推奨オプション)]

このオプションを選択すると、MQSeriesコネクタはDTD / XSDの全てのルート要素を検索し、見つかった各ルート要素ごとに使用可能なドキュメントタイプを発行します。

例: 文書編集用のDTDで、ルート要素(「Book」)が1つだけ見つかったとします。この場合、MQSeriesコネクタの使用可能なドキュメントタイプは1つのみです。



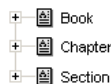
注意:

ルート要素が見つからない場合(全ての要素が他の要素と包含関係にある場合)、Connect-Itは、DTDまたはXSD内で最初に検索された要素をルート要素と見なします。ルート要素が見つからないのは、例えば、DTDがリレーショナル(関係型)データベースの構造と同様に構成されている場合などです。リレーショナルデータベースでは、全要素(つまり全テーブル)がお互いを包含し合っているため、ルート要素はありません。例: 資産のテーブルはユーザのテーブルに関連し、ユーザのテーブルはユーザの使用する資産のテーブルに関連しています。

[ユーザの選択する各ルート要素に使用可能なドキュメントタイプを発行する(コマで値を区切る)]

このオプションを選択すると、ユーザはコマで値を区切って、ルート要素を自分で指定することができます。

例：文書編集用のDTDで、ユーザが、3つのルート要素「Book」、「Chapter」と「Section」を選択するとします。この場合MQSeriesコネクタは、ユーザの選んだルート要素ごとに使用可能なドキュメントタイプを発行します。



注意:

DTDがリレーショナルデータベースの構造（全要素つまり全テーブルがお互いを包含し合う関係）を取っている場合、このオプションを使用すると、テーブルごとに使用可能なドキュメントタイプを発行するように指定できます。

学習モード

学習モードを使用すると、MQSeriesコネクタは、**[キューの選択]** ページで指定したキュー内にあるメッセージに基づいてDTDを作成します。

注意：この機能はXSDでは使用不可能です。

注意:

学習モードで取得されたDTDは、ローカルまたはネットワークファイル内に保存されなければなりません。HTTPまたはFTPサイト上に保存することはできません。

学習モードを使用するには、

- **[学習モード]** オプションを選択し、MQSeriesコネクタを設定します。
- シナリオ図部分でMQSeriesコネクタを選択します。
- **[ツール / 生成する]** メニューを選択します。
または
- 右クリックして、ショートカットメニューから **[生成する]** を選択します。
または
- **[F5]** キーを押します。


学習モードを使用しない場合は、MQSeriesコネクタを再度設定し、**[DTDを選択する]** ページでオプションをオフにします。

XMLドキュメントとDTD / XSDドキュメントの矛盾

原則として、MQSeriesコネクタに読み取られるメッセージのXML要素は、DTDまたはXSD内に定義された要素に一致するはずですが、メッセージに新

規の要素が現れることもあります。要素の少ないメッセージを学習モードで処理した時に得られたDTDを使用する際に、このようなXMLとDTDの矛盾が頻繁に起こります。

この矛盾を管理するために、Connect-Itには2つのオプションがあります。

- ドキュメントを拒否し、処理を続行する
読み取られたメッセージ内に、DTDで定義されていない要素があると、このメッセージは拒否されます。(ドキュメントが保存されるかどうかは[**処理後のアクション**]ページのオプションによります。)メッセージを適切に処理するには、MQSeriesコネクタを学習モードで再起動する必要があります。
- 警告を作成し、処理を続行する
矛盾が発生しても、メッセージは正常に処理されます。 アイコンが、読み取りモードのMQSeriesコネクタに関連するドキュメントログのトラッキング項目内に表示されます。

XMLファイルのルート要素が現在のドキュメントタイプに一致しない場合、ファイルを無視する

このオプションを選択すると、処理されるドキュメントのルート要素が、ドキュメントの基盤となるドキュメントタイプのルート要素に対応しない場合、コネクタはドキュメント、処理レポートまたはエラーを生成しません。

シナリオで複数のXMLコネクタが使用されている場合、このオプションは便利です。XMLファイルは、シナリオ内の別のXMLコネクタで定義されている別のドキュメントタイプに、対応する場合もあるからです。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

MQSeriesコネクタ（書き込み）の設定

MQSeriesコネクタ（書き込み）の設定では、コネクタがどのMQSeriesキュー内にメッセージを書き込むか、を設定できます。コネクタは、読み取ったConnect-Itドキュメントを基にメッセージをキュー内に書き込みます。

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 4.3. MQSeriesコネクタ（書き込み）の設定

コネクタに名前を付け、役割を説明する [p. 14]
処理モードを選択する [p. 134]
キューマネージャへ接続する [p. 134]
再接続のパラメータを設定する [p. 27]
サーバとの時間差を指定する [p. 29]
キューの選択 [p. 136]
DTD / XSDを選択する [p. 137]
スケジュールのポインタを設定する [p. 32]
ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

処理モードを選択する

【基本設定】ウィザードの2ページ目では、MQSeriesコネクタの処理モードを選択します。【書き込み】モードを選択します。

キューマネージャへ接続する

このページでは以下の内容を入力します。

- MQSeriesキューマネージャへ接続するためのMQSeriesコネクタのパラメータ
- MQSeriesサーバへ再接続するためのMQSeriesコネクタのパラメータ
- Connect-ItとMQSeriesサーバ間の時間差

キューマネージャへの接続

MQSeriesコネクタがキューマネージャへ接続できるように、4つのパラメータを指定します。

- **【サーバ名】**
このフィールドには、ネットワーク上のMQSeriesサーバのDNS名、またはIPアドレスを入力します。
- **【接続ポート】**
キューマネージャのポート番号を指定します。このポートのデフォルト値は「1414」です。

- **【チャンネル名】**
キューマネージャが通信パスとして使用する接続チャンネル名を入力します。デフォルト値は「SYSTEM.DEF.SVRCONN」です。
- **【キューマネージャ名】**
キューマネージャ名を入力します。MQSeriesコネクタの付属シナリオを使用する場合、キューマネージャ名はconnect.itです。このフィールドが入力されていない場合、MQSeriesコネクタは、MQSeriesのデフォルトのキューマネージャに接続します。
- **【MQSeriesサーバはOS/390上で稼動する】**
このオプションを使用すると、OS/390がサポートしない全機能を無効できます。
このオプションを選択すると、以下の機能がコネクタのルール内で使用不可能になります。

取り込み用ルール

自動セグメント化は不可能

GroupIDの指定は不可能

生成用ルール

【グループの全メッセージが使用可能な場合にのみメッセージを取得する】 オプションはオフ

WHERE句ではGroupIDキーワードは作動不可能

コネクタの生成用または取り込み用ドキュメントタイプでは、GroupIDノードがMessageInfo構造体から消えます。



注意:

【拡張接続を使用する】 デフォルトオプションを選択しないと、**【チャンネル名】**と**【キューマネージャ名】**フィールドのみが入力可能になります。

MQSeriesサーバへの再接続用パラメータ

再接続オプションでは、MQSeriesサーバへの接続が失敗した場合に、MQSeriesコネクタが再接続する方法を指定できます。

再接続オプション用のフィールドは以下の通りです。

- **【基本待ち時間】**
【基本待ち時間】は、コネクタが何秒後にMQSeriesサーバへの再接続を実行するかを指定します。再接続に失敗すると、Connect-Itは基本待ち時間を2倍にして接続を再試行します。例えば、基本待ち時間が2秒の場合、2回目の再接続は4秒後に実行され、3回目の再接続は8秒後に実行されます。
- **【制限時間】**

この [制限時間] で、コネクタがサーバへの再接続を何分（時間）後に中止するかを指定できます。

 **注意:**

再接続に成功すると、中断されたConnect-Itセッションは最初から再始動されません。

[サーバとの時間差]

[サーバとの時間差] フィールドでは、Connect-ItサーバとMQSeriesサーバ間の時間差を指定できます。このフィールドのデフォルト値は「0秒」です。この時間差は、MQSeriesコネクタ用のシナリオをテスト段階で試行してから入力するようにします。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

キューの選択

このページでは、MQSeriesコネクタがどのキュー内にメッセージを書き込むかを指定します。MQSeriesの付属シナリオを使用する場合、キューの名前は「PEREGRINE.OUT」です。

[モデルキュー] オプション

[キュー名] フィールドに入力されたキューがモデルキューである場合は、このオプションを選択します。

使用するダイナミックキュー名のフォーマットを指定します。「*」は、作成されるダイナミックキュー名が固有になるように、キューマネージャにより置換されます。

[モデルキュー] オプションを選択すると、[終了時のオプション] がアクティブになります。以下の3つのオプションがあります。

- 永久ダイナミックキューは保存され、一時ダイナミックキューは破棄されません。

- 空の永久ダイナミックキューは破棄されます。一時ダイナミックキューも破棄されます。
- メッセージは永久ダイナミックキューから消去され、永久ダイナミックキューは破棄されます。一時ダイナミックキューも破棄されます。

DTD / XSDを選択する

XMLファイル进行处理するために、MQSeriesコネクタはDTD (Document Type Definition) またはXSD (Extended Schema Definition) を使用する必要があります。このページでは、使用するDTDまたはXSDを [DTD / XSD] フィールドで指定できます。

 をクリックすると、 [ファイルの場所] ページが表示されます。このページでは、接続プロトコルの種類に応じてDTDまたはXSDのパス名を指定できます。

ファイルの場所

以下の3つのオプションがあります。

- [ローカル / ネットワーク]

Connect-Itがインストールされているコンピュータやネットワーク上の、DTDまたはXSDの完全パス名を入力します。

- [FTP]

[サーバ名]、[ログイン]、[パスワード] と [パス] フィールドにFTPパラメータを入力します。

[パス] フィールドでは、DTDまたはXSDのサーバ上の場所を指定します。例えば、DTDまたはXSDがサーバ「ftp.mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、 [サーバ名] フィールドに「ftp.mycompany.com」を入力し、 [パス] フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。

 **注意:**

スラッシュ (/) はサーバ名の最後には入れませんが、DTDまたはXSDパスの最初に入れます。

- [HTTP]

[サーバ名]、[ログイン]、[パスワード] と [パス] フィールドにHTTPパラメータを入力します。

[パス] フィールドでは、DTDまたはXSDのサーバ上の場所を指定します。例えば、DTDまたはXSDがサーバ「http://mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、 [サーバ名] フィールドに「http://mycompany.com」を入力し、 [パス] フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。

 **注意:**

スラッシュ (/) はサーバ名の最後には入れませんが、DTDまたはXSDパスの最初に入れます。

[XMLファイルで取得したDTDを参照しない]

このオプションはデフォルトのオプションです。

[取得したDTD / XSDをXMLファイル内に書き込む]

このオプションを選択すると、MQSeriesコネクタに書き込まれたMQSeriesメッセージは、ヘッダーにDTDを含みます。

[取得したDTD / XSDを外部ファイルに書き込み、XMLファイル内で参照する]

このオプションを選択する場合は、作成されるMQSeriesメッセージのDTDを保存する場所を、フィールドに入力する必要があります。DTDまたはXSDは、ローカルまたはネットワークファイル上に保存されなければなりません。

 **警告:**

シナリオを起動すると複数のセッションが実行されます。各セッションは、スケジューラがシナリオのソースコネクタを起動する度に開始します (例: ソースコネクタは毎週ドキュメントを生成する、などのスケジュール)。

DTDまたはXSDは、シナリオの第1回目のセッション時にのみ外部ファイルに書き込まれます。つまりその後のセッションでも、第1回目のセッション時に取得され外部ファイルに保存されたDTDまたはXSDが、使用されることとなります。

新規のDTD / XSDが外部ファイルに書き込まれるようにするには、シナリオを停止し、再起動しなければなりません。

ルート要素を指定する

DTDまたはXSDは複数の要素から構成されており、これらの要素は更に別の要素を含んでいます。(例: 文書編集用のDTDでは、複数の「Section」要素が複数の「Chapter」要素に含まれており、更に「Chapter」要素は1つの要素「Book」に含まれています。) 他の要素に含まれない要素は、ルート要素と呼ばれます。DTDまたはXSDは、1つまたは複数のルート要素から構成されることもあれば、全くルート要素が存在しないこともあります。

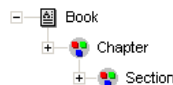
ルート要素（つまり、MQSeriesコネクタに発行される使用可能なドキュメントタイプに当たります）を指定するためには、以下の2つのオプションがあります。

- [DTD / XSD内に見つかった各ルート要素に使用可能なドキュメントタイプを発行する(推奨オプション)]
- [ユーザの選択する各ルート要素に使用可能なドキュメントタイプを発行する(コンマで値を区切る)]

[DTD / XSD内に見つかった各ルート要素に使用可能なドキュメントタイプを発行する(推奨オプション)]

このオプションを選択すると、MQSeriesコネクタはDTDまたはXSDの全てのルート要素を検索し、見つかった各ルート要素ごとに使用可能なドキュメントタイプを発行します。

例：文書編集用のDTD / XSDで、ルート要素（「Book」）が1つだけ見つかったとします。この場合、MQSeriesコネクタの使用可能なドキュメントタイプは1つのみです。



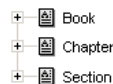
注意:

ルート要素が見つからない場合（全ての要素が他の要素と包含関係にある場合）、Connect-Itは、DTDまたはXSD内で最初に検索された要素をルート要素と見なします。ルート要素が見つからないのは、例えば、DTDまたはXSDがリレーショナル（関係型）データベースの構造と同様に構成されている場合などです。リレーショナルデータベースでは、全要素（つまり全テーブル）がお互いを包含し合っているため、ルート要素はありません。例：[資産]のテーブルはユーザのテーブルに関連し、ユーザのテーブルはユーザの使用する資産のテーブルに関連しています。

[ユーザの選択する各ルート要素に使用可能なドキュメントタイプを発行する(コンマで値を区切る)]

このオプションを選択すると、ユーザはコンマで値を区切って、ルート要素を自分で指定することができます。

例：文書編集用のDTD / XSDで、ユーザが、3つのルート要素「Book」、
「Chapter」と「Section」を選択するとします。この場合MQSeriesコネクタは、
ユーザの選んだルート要素ごとに使用可能なドキュメントタイプを発行します。



注意:

DTDがリレーショナルデータベースの構造（全要素つまり全テーブルがお互いを包含し合う関係）を取っている場合、このオプションを使用すると、テーブルごとに使用可能なドキュメントタイプを発行するように指定できます。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

MQSeriesコネクタの生成用ルール

MQSeriesコネクタの生成用ルールにより、設定で指定されたキュー内でコネクタが読み取るメッセージに、フィルタをかけることができます。



以下の内容を指定してルールを作成します。

- WHERE句
- 取得用のオプション

ルールを作成するには、

- 1 MQSeriesコネクタをダブルクリックします。

- 2 ダイアログボックスの [生成用ドキュメントタイプ] 枠内で、ドキュメントタイプを1つ選択します。
- 3 生成用ドキュメントタイプのルート要素を選択します。
- 4 生成用ルールを入力します。

WHERE句

WHERE句では、MQSeriesコネクタに読み取られるメッセージにフィルタをかけます。WHERE句は以下のMQSeriesシンタックスで作成されます。

[キーワード]=[値]

以下のキーワードがあります。

- MsgID (メッセージのID)
- GroupID (メッセージグループのID)
- CorrelID (相関ID)

キーワードは16進数形式で入力します。

MQSeriesコネクタ付属のMQSeries - Asset Managementシナリオでは、生成用ドキュメントタイプ用の相関ID (CorrelID) は以下の通りです。

- CONNIT.MQAM.REQUEST.ACK
(434f4e4e49542e4d51414d2e524551554553542e41434b) : ExtRequestAcksドキュメントタイプ用
- CONNIT.MQAM.RECEIPT.ACK
(434f4e4e49542e4d51414d2e524543454950542e41434b) : ExtReceiptAcksドキュメントタイプ用
- CONNIT.MQAM.VENDOR (434f4e4e49542e4d51414d2e56454e444f52) : Vendorsドキュメントタイプ用
- CONNIT.MQAM.COSTCENTER
(434f4e4e49542e4d51414d2e434f535443454e544552) : CostCentersドキュメントタイプ用

MQSeriesコネクタ付属のMQSeries - ServiceCenterシナリオでは、相関ID (CorrelID) 1つがExternal Contacts生成用ドキュメントタイプ用に使用されます : CONNIT.MQSC.CONTACT (434f4e4e49542e4d51414d2e434f535443454e544552)

これらの相関IDは、外部アプリケーションがPEREGRINE.INキューに置くMQSeriesメッセージに適用されます。

取得するメッセージの種類

このリストでは、コネクタが取得するメッセージの種類を選択できます。

[グループの全メッセージが使用可能な場合にのみメッセージを取得する] オプション

このオプションでは、MQSeriesコネクタがメッセージをキューから取得する方法を指定します。

MQSeriesコネクタの取り込み用ルール

MQSeriesコネクタの取り込み用ルールにより、設定で選択されたキュー内にコネクタがメッセージを書き込む方法を指定できます。



以下の内容を指定してルールを作成します。

- 自動セグメント化の許可または禁止
- 特定のIDの指定
- 送信するメッセージの種類

ルールを作成するには、

- 1 MQSeriesコネクタをダブルクリックします。
- 2 ダイアログボックスの [取り込み用ドキュメントタイプ] 枠内で、ドキュメントタイプを1つ選択します。
- 3 取り込み用ドキュメントタイプのルート要素を選択します。
- 4 取り込み用ルールを入力します。

または

- 1 書き込みモードのMQSeriesコネクタにリンクしているマッピングボックスを、ダブルクリックします。
- 2 MQSeriesコネクタがターゲットコネクタとしてリンクされているマッピングを、編集または作成します。
- 3 [メッセージのオプション] タブを選択します。
- 4 取り込み用ルールを入力します。

自動セグメント化

メッセージのセグメント化を許可すると、キューで許容されている最大サイズよりも大きいメッセージは、複数の物理メッセージに分割されます。これらの物理メッセージは1つの論理メッセージ内にまとめられます。

メッセージのセグメント化を禁止すると、キューで許容されている最大サイズよりも大きいメッセージは、MQSeriesコネクタに拒否されます。

ID指定

このオプションでは特定のIDを指定できます。

- グループID
- 関連ID

IDは16進数形式で入力します。

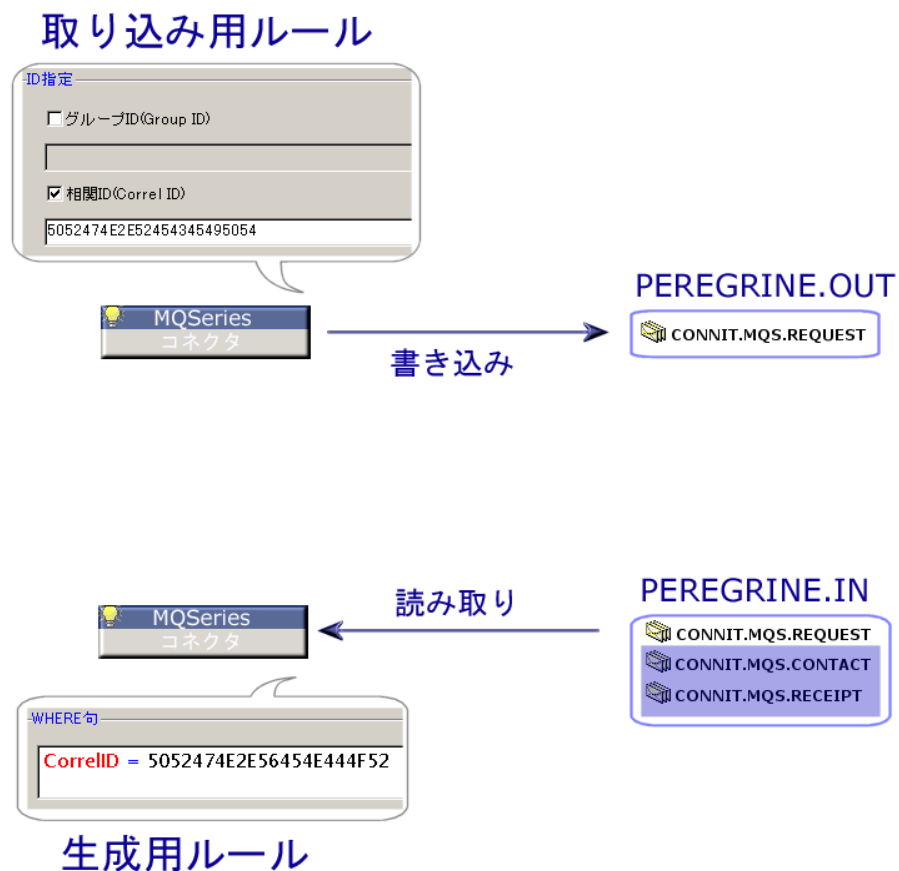
MQSeriesコネクタ付属のMQSeries - Asset Managementシナリオでは、取り込み用ドキュメントタイプ用の関連ID (CorrelID) は以下の通りです。

- CONNIT.MQAM.REQUEST (434f4e4e49542e4d51414d2e52455155455354) : Requestドキュメントタイプ用
- CONNIT.MQAM.RECEIPT (434f4e4e49542e4d51414d2e52454345495054) : Receiptドキュメントタイプ用

MQSeriesコネクタ付属のMQSeries - ServiceCenterシナリオでは、関連ID (CorrelID) 1つがContacts from ServiceCenter取り込み用ドキュメントタイプ用に使用されます : CONNIT.MQSC.CONTACT (434f4e4e49542e4d51414d2e434f535443454e544552)

指定されたIDは、PEREGRINE.OUTキュー内の発注や受領伝票に対応するメッセージを識別します。メッセージを読み取る外部アプリケーションは、メッセージを取得するために同じIDを使用しなければなりません。

図 4.1. MQSeriesコネクタ - 関連IDによるMQSeriesメッセージの識別



メッセージの種類

この枠では以下の操作を実行できます。

- コネクタが送信するメッセージの種類を選択する
- コネクタが取り込む「要求」型のドキュメントタイプに対する応答を含むキューを選択する

追加情報

本節ではコネクタに関する追加情報を説明します。

取り込み用ドキュメントタイプ内のMessageInfoドキュメントタイプ

コネクタの取り込み用ドキュメントタイプには、MessageInfo構造体があります。この構造体は処理レポートで使用できます。

処理レポートの詳細については、Connect-It『ユーザガイド』の「**処理レポート**」の章を参照してください。

このドキュメントタイプには以下のフィールドがあります。

- Expiry
このフィールドは、コネクタがメッセージを送信する際の送信期日を指定します。このフィールドの値は、10分の1秒単位で入力されます。メッセージに期日がない場合、フィールド値は「-1」になります。
- Persistence
このフィールドはメッセージの有効性を表しており、以下の3つの値が入力されます。
 - 0
メッセージの有効性なし
 - 1
メッセージの有効性
 - 2
キューのデフォルトの有効性。これは、このフィールドのデフォルト値です。
- Priority
このフィールドではメッセージの優先順位を指定できます。このフィールド値は、0以上の整数でなければなりません。このフィールドに値が入力されていないと、キューのデフォルトの優先順位が使用されます。
- CorrelationID
このフィールドでは、ドキュメントのCorrelationIDの値を指定できます。このフィールドをマップすると、このフィールド値は、コネクタのルール内で指定されたCorrelationIDの値を上書きして保存されます。
- GroupID
このフィールドでは、送信するメッセージのGroupIDを指定できます。このフィールドをマップすると、このフィールド値は、コネクタのルール内で指定されたGroupIDの値を上書きして保存されます。

生成用ドキュメントタイプ内のMessageInfoドキュメントタイプ

コネクタの生成用ドキュメントタイプには、MessageInfo構造体があります。この構造体は処理レポートで使用できます。

処理レポートの詳細については、Connect-It『ユーザガイド』の「**処理レポート**」の章を参照してください。

このドキュメントタイプには以下のフィールドがあります。

- BackoutCount
- Expiry
- MsgType
- Persistence
- Priority
- PutDate
- ReplyToQ
- ReplyToQMgr
- Report
- UserID
- MsgID
- CorrelationID
- GroupID

拡張処理レポート

取り込みモード（メッセージの送信）で、コネクタは拡張処理レポート（ProcessReportMessageInfoドキュメントタイプ）を発行します。このドキュメントタイプは、送信されるメッセージの記述に対応するMessageInfo構造体を含みます。MessageInfo構造体には以下のフィールドが含まれています。

- MsgType
- PutDate
- ReplyToQ
- Report
- UserID
- MsgID
- CorrelationID
- GroupID

付属シナリオ - MQSeriesコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- `mqseries/acmq/mqtoprgn.scn` [p. 311]
- `mqseries/acmq/prgntomq.scn` [p. 312]
- `mqseries/acmq/mqsc.scn` [p. 312]

NTセキュリティコネクタ

ベースコネクタ

- 生成 (ソース)

本節ではNTセキュリティコネクタについて説明します。このコネクタは、Connect-Itがインストールされているコンピュータの、NTドメインに関する情報を取得します。この情報はコンピュータ、ユーザや、NTドメインのユーザグループに関するものです。

このコネクタはドキュメントを生成できますが、取り込むことはできません。

NTセキュリティコネクタの互換性

NTセキュリティコネクタは、Windows 32ビットのオペレーティングシステムを使用するコンピュータで機能します。

 注意:

新規のMS Active Directoryセキュリティモデルを使用する場合は、LDAPコネクタの使用をお勧めします。このセキュリティモデルはNTセキュリティコネクタにはサポートされていません。

NTセキュリティコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 4.4. NTセキュリティコネクタの設定

 _____ コネクタに名前を付け、役割を説明する [p. 14]

 _____ NTドメインを入力する [p. 148]

 _____ ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

NTドメインを入力する

[ドメイン] フィールドで、NTセキュリティ情報をどのドメインから取得するかを指定します。デフォルトで表示されるドメイン名は、Connect-It起動中のNTセッションのドメインに一致します。

注意:

複数ドメインの情報を取得する場合は、処理するドメインの名前をセミコロン (;) で区切って入力します。

コンピュータからアクセス可能な全NTドメインを取得する場合は、アスタリスク (*) を入力します。この操作を行うとコンピュータのメモリに負荷がかかるため、Connect-Itの実行が中断される可能性があります。

警告:

シナリオ図内でコネクタ名の下に表示される短い説明は、常にConnect-Itの現在のドメイン名とユーザ名です。このページで別のドメイン名を入力しても、コネクタの短い説明内には表示されません。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

NTセキュリティコネクタの使用可能なドキュメントタイプ

NTセキュリティコネクタには3つの使用可能なドキュメントタイプがあります。

- Machine

- NtDomain
- User

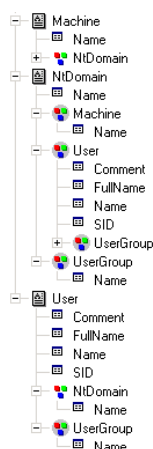
 **注意:**

「NtDomain」ドキュメントタイプは、コネクタの設定時に指定されたNTドメインに一致しません。

NTセキュリティコネクタの使用可能なドキュメントタイプには、以下のサブノード（構造体とコレクション）があります。

- ドメインのコンピュータ（Machineコレクション）
- ドメインのユーザ（Userコレクション）
- ドメインのユーザグループ（UserGroupコレクション）

図 4.2. NTセキュリティコネクタ - 使用可能なドキュメントタイプ



NTセキュリティコネクタの生成用ルール

コネクタの生成用ルールの入力方法については、「コネクタのルール（ディレクトクティブ） [p. 39]」章の「生成用ルール [p. 40]」の節を参照してください。

NTセキュリティコネクタの生成用ルールでは、ドキュメントタイプのコレクションの構成要素にフィルタを適用できます。

NTドメインの要素に、名前フィルタをかける

[名前] フィールドに入力すると、生成されるドキュメント内でコレクションの構成要素にフィルタを適用できるようになります。

このフィルタを作成するには、特殊文字「*」と「?」を使用する必要があります。

警告:

フィルタを作成するには、コネクタの生成用ドキュメントタイプ内でコレクションを1つ選択しなければなりません。

表 4.5. NTセキュリティコネクタ - 生成用ルールの例

コレクション	フィルタ	取得する値
Machine	Plateforme1	Plateforme1コンピュータ
Machine	a*	名前が「a」で始まるドメインの全コンピュータ
User	Dupon?	ドメインのユーザの内、名前の始めが「Dupon」で、名前の終わりの文字が不明である全ユーザ

付属シナリオ - NTセキュリティコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- ntsec/htim5/htim.scn [p. 305]
- ntsec/htac36/addcpu.scn [p. 313]
- ntsec/htac36/adduser.scn [p. 313]
- ntsec/htac40/addcpu.scn [p. 313]
- ntsec/htac40/adduser.scn [p. 313]
- ntsec/htac41/addcpu.scn [p. 314]
- ntsec/htac41/adduser.scn [p. 314]

5 | プロトコルコネクタ

プロトコルコネクタは、特定の情報プロトコルで処理されるデータ用のコネクタです。

データベースコネクタ

オプションコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

データベースコネクタは、ODBCデータソース、またはOracle、SybaseおよびMySQLデータベースから直接データを処理します。

! 警告:

ODBC接続を使用するには、ODBCアドミニストレータが先ずインストールされていなければなりません。

データベースコネクタの互換性

データベースコネクタは、以下の表のドライバとODBC Administratorバージョン3.520.4403.2とのテストに成功しています。

ODBC	Oracle
Accessバージョン4.00.4403.02	Oracle 7バージョン2.05.0301
Excelバージョン4.00.4403.02	Oracle 8.0バージョン8.00.0500
Sybase 11.9バージョン3.11.0001	Oracle 8iバージョン8.01.0500
Sybase SQL Anywhereバージョン5.05.052739	
Textバージョン4.00.4403.02	
Microsoft SQL Serverバージョン3.70.08.20	
IBM DB2バージョン7.01.00.00 (バージョン6より後のDB2で)	
INFORMIX 3.33 32 BIT バージョン3.33.00.10688	

データベースコネクタは、全てのODBCドライバのバージョン3.xxと互換性があります。

データベースコネクタの既知の制約点

データベースコネクタを適切に使用するためには、ODBCドライバとODBCアドミニストレータのDLLバージョンが、同一である必要があります。

例：

ODBC Administratorバージョン3.0と、ODBC Microsoft Accessドライババージョン4.00を使用していると、データベースコネクタは機能しません。

サポートされているフィールドのタイプ

データベースコネクタの使用可能なドキュメントタイプは、フィールドから構成されています。各フィールドには、それぞれ特定のフィールドタイプ（テキストフィールド、整数フィールド、Blobフィールドなど）があります。

使用されているデータベースドライバの種類により、データベースコネクタにサポートされるフィールドタイプは異なります。

Oracle 8.0 - Oracle用Microsoft ODBCドライバ、バージョン2.573.6526.00

フィールドタイプ	サポート可/不可
bfile	不可
blob (可変長バイナリフィールド)	可
char	可
clob	可
Date (日付)	可
Float (浮動小数点)	可
integer	可
long整数 (32ビット)	可
long raw	可
nchar	可
nchar	不可
nvarchar	不可
number	可
ms_label	可

MSSQL 7.0 - SQL Serverドライバ、バージョン2000.80.194.00

フィールドタイプ	サポート可/不可
binary	可
bit	可
char	可
datetime	可
decimal	可
Float (浮動小数点)	可
image	可
int	可
money	可
nchar	不可
ntext	可
numeric	可
nvarchar	不可
real	可
smalldatetime	可
smallint	可

フィールドタイプ	サポート可 / 不可
smallmoney	可
text	可
timestamp	可
tinyint	可
uniqueidentifier	不可
varbinary	可
varchar	可

Sybase12 : Sybase System 11バージョン3.11.00.01











フィールドタイプ	サポート可 / 不可
binary	可
char	可
datetime	可
decimal	可
Float (浮動小数点)	可
image	可
int	可
money	可
nchar	可
ntext	可
numeric	可
nvarchar	可
real	可
smalldatetime	可
smallint	可
smallmoney	可
text	可
tinyint	可
varbinary	可
varchar	可

データベースコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 5.1. データベースコネクタの設定

	コネクタに名前を付け、役割を説明する [p. 14]
	接続タイプを選択する [p. 15]
	接続を選択する [p. 16]
	高度な設定 [p. 24]
	トランザクションを設定する [p. 34]
	再接続のパラメータを設定する [p. 27]
	サーバとの時間差を指定する [p. 29]
	スケジュールのポインタを設定する [p. 32]
	キャッシュを設定する [p. 35]
	ピボットドキュメントタイプの使用 [p. 36]

接続条件

コネクタの設定の前に、データベースへのODBC接続が、ODBCアドミニストレータ内で宣言されていなければなりません。

有効な宣言は、ODBC接続用の適切なログインとパスワードを使用しなければなりません。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続タイプを選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続タイプを選択する [p. 15]」の節を参照してください。

接続を選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続を選択する [p. 16]」の節を参照してください。

高度な設定

このページの使用方法については、「コネクタの設定 [p. 13]」章の「高度な設定 [p. 24]」の節を参照してください。

トランザクションを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

データベースコネクタの生成用ルール

コネクタの生成用ルールの入力方法については、「コネクタのルール（ディレクティブ） [p. 39]」章の「生成用ルール [p. 40]」の節を参照してください。

このコネクタのルールとしてWHERE句とORDERBY句を作成すると、ソースデータベースのレコードにフィルタを適用できるようになります。これらの句の詳細については、「コネクタのルール（ディレクティブ） [p. 39]」章の「生成用ルール [p. 40]」節、「WHERE句とORDER BY句 [p. 41]」を参照してください。

AQLシンタックスの詳細については、「AQLクエリ [p. 331]」の章を参照してください。

sysdateフィールド

AQLで作成されるWHERE句では、sysdateフィールドは、getdate()関数に置換されなければなりません。

例：SQLクエリSELECT * from AmAsset where dinstall>sysdate は、AQLクエリSELECT * from AmAsset where dinstall>getdate()に置き換えられなければなりません。

データベースコネクタの取り込み用ルール

コネクタの取り込み用ルールの入力方法については、「コネクタのルール（ディレクティブ） [p. 39]」章の「取り込み用ルール [p. 43]」の節を参照してください。

取り込み用ルールを作成するには、**[整合性]**と**[高度な整合性チェック]**タブ内で整合性用のパラメータを入力します。

これらの整合性用タブの詳細については、「コネクタのルール（ディレクティブ） [p. 39]」章の「取り込み用ルール [p. 43]」節内「整合性 [p. 44]」を参照してください。

追加情報

本節では以下の内容について説明します。

- ODBCソースの宣言に関する詳細

ODBCソースの宣言に関する詳細

データベースコネクタを含むシナリオが、Windows 32ビット下のサービスとして起動する場合は、ODBCデータソースは、ユーザデータソース（ユーザDNS）ではなく、システムデータソース（システムDNS）として宣言されていなければなりません。

ODBCデータソースがシステムデータソースであることを確認する（サービスモードの場合）

データソースがシステムデータソースであることを確認するには、

- 1 ODBCアドミニストレータを起動します。
- 2 コネクタに使用されているODBCデータソースが、**[システムDNS]**タブ内にあるかどうか確認します。

[システムDNS]タブ内にはない場合はデータソースを削除し、新規のODBCソースを作成します。

付属シナリオ - データベースコネクタ

このコネクタ用のシナリオはありません。

Eメールコネクタ

オプションコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

Eメールコネクタにより、Eメールメッセージの受信と送信が可能になります。Eメールコネクタ (受信) により受信されるメッセージの場合、メッセージの情報は、Connect-Itで処理 (マッピング) された後外部アプリケーション内に統合されます。例: E-mail - ServiceCenterタイプのシナリオでは、メールサーバに受信されたメッセージの情報を使用して、ServiceCenter内にチケットを作成できます。

Eメールコネクタ (送信) により送信されるメッセージの場合、Connect-It内で、外部アプリケーションのデータがEメールのフィールドにマップされます。例: Asset Management - E-mailの付属シナリオ (finreque.scn) では、[依頼] テーブルの各レコードごとにメッセージが作成され、依頼者の責任者にメッセージが送信されます。

本節では以下の内容について説明します。

- コネクタの互換性
- Eメールコネクタの制約点

Eメールコネクタの互換性

Eメールコネクタは、以下のアプリケーションとのテストに成功しています。

プロトコル	例
Lotus VIM	Lotus Notesバージョン4.6と5 Lotus cc:Mail
Microsoft MAPI	Outlookバージョン98 と2000 Exchange 5.0
POP 3-SMTP	Outlook Express Eudora Internet mail

 **警告:**

Connect-ItがインストールされているコンピュータにLotus NotesとOutlook Expressがインストールされていると、Eメールコネクタは両アプリケーションと共に正常に機能します。

Eメールコネクタの既知の制約点

Eメールコネクタをシナリオ内で使用する前に、以下の内容を熟読してください。

サポートされているコードページ

Eメールコネクタにサポートされているコードページは以下の通りです。

- Connect-ItのANSIバージョンでは、
 - Latin-1言語（英語、フランス語、など）と互換性のあるオペレーティングシステム：ISO-8859-1、ASCII 7
 - Latin-2言語（ポーランド語などの中央ヨーロッパ言語）と互換性のあるオペレーティングシステム：ISO-8859-2、ASCII 7
- Connect-ItのUNICODEバージョン：ISO-8859-1、ISO-8859-2、ASCII 7、SHIFT-JS、EUC-JP、UTF-8、UNICODE

その他の制約点

Eメールコネクタの様々な制約点は以下の通りです。

- Eメールの「プライオリティ」フラグを処理できるのはSMTPサーバのみです。
- コネクタはBase64とQuoted-Printableのエンコードのみをサポートします。
- UnicodeサポートはSMTPとMAPIサーバとのみ使用可能です。
- Quoted-PrintableでエンコードされたEメールが適切に処理されていても、エラーメッセージが表示されます。

 **警告:**

Connect-It クライアントコンピュータの設定：Eメールコネクタが適切に機能するためには、Connect-It がインストールされているコンピュータに、Microsoft Internet Explorerバージョン5.xxがインストールされていなければなりません。

Eメール（受信）コネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 5.2. Eメール（受信）コネクタの設定

■	コネクタに名前を付け、役割を説明する [p. 14]
■	接続用パラメータを定義する [p. 160]
■	処理後のアクションを定義する [p. 162]
■	メッセージの内容を処理する [p. 162]
■	ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続用パラメータを定義する

このページでは、Eメール（受信）コネクタのメッセージプロトコルを選択します。

このページに表示されるフィールドの数は、[メールエンジンの種類] フィールドで選択されたプロトコルの種類により変化します。Eメール（受信）コネクタで利用できるプロトコルは以下の通りです。

- Lotus VIM (Vendor Independent Messaging)
- POP3 (Post Office Protocol)
- Microsoft MAPI (Messaging Application Program Interface)

Lotus VIM

VIMプロトコルを選択する場合は、2つのフィールドに入力します。

- [ログイン]
コンピュータにインストールされているVIMメッセージシステムへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

POP3

POP3プロトコルを選択する場合は、4つのフィールドに入力します。

- [POP3サーバ]
POP3サーバの名前を指定します。
- [接続ポート]
サーバに使用されているコンピュータの接続ポートを指定します。このフィールドのデフォルト値は「110」です。
- [ログイン]
POP3サーバへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

Eメール処理のオプション

POP3プロトコルを使用する場合、以下のオプションがあります。

- [Eメールを取得した後サーバから削除しない]
- [Eメールを取得した後サーバから削除する]
このオプションを選択すると、Connect-Itにメッセージが取り出された後、メッセージは削除されます。これにより、処理済みのメッセージが受信トレイ内にたまるのを回避できます。
- [DOS/Windowsテキスト形式を使用する。CR + LF（復帰+改行）を使用する。]
外部アプリケーションが、DOSまたはWindows32ビットをオペレーティングシステムとして使用している場合、このオプションを選択しなければなりません。両システムでは、復帰（Carriage Return - CR）の次に改行（Line feed - LF）が来ます。ターゲットアプリケーションが、UNIXをオペレーティングシステムとして使用している場合は、このオプションは使用しません。

MAPI

MAPIプロトコルを選択する場合は、2つのフィールドに入力します。

- [プロファイル]
コンピュータにインストールされたMAPIメッセージシステムへのアクセスを許可するログインを入力します。
- [パスワード]
オペレーティングシステム用のユーザパスワードを入力します（例：Windows NTパスワード）。パスワードは入力時に非表示になり、暗号化されてから保存されます。

[テスト] ボタン

このボタンを押すと接続をテストできます。

接続をテストするには、

- 1 接続用パラメータを入力します。
- 2 [テスト]をクリックします。

[接続のテスト]ウィンドウが開き、接続の成功または失敗を表示します。
接続に失敗すると、失敗の原因が表示されます。

- 3 [閉じる]をクリックして設定ウィザードへ戻ります。

処理後のアクションを定義する

このページでは、コネクタが生成したドキュメントの処理後に、コネクタが適用するアクションを指定します。

[メッセージの取得直後にアクションを適用する]

このオプションを選択すると、シナリオの他のコネクタが送信する処理レポートに応じて、アクションが適用されます。

処理レポートの詳細については、『ユーザガイド』の「処理レポート」の章を参照してください。

処理後のアクションの種類

[メッセージ処理に成功した場合]と[メッセージ処理に失敗した場合]の枠内で、以下の2つのオプションのいずれかを選択します。

- [サーバのEメールを変更しない]
- [Eメールをサーバから削除する]

メッセージの内容を処理する

Eメールコネクタでは、DTDファイルに基づいて受信するEメールの本文と添付ファイルを解析できます。このDTDファイルは、Eメール本文と添付ファイルの構成をXML要素で説明するものです。

DTDファイルを使ってメッセージの内容を処理する

- 1 [Eメール本文の内容を処理する]オプションを選択します。
- 2 [DTDファイル]フィールドに、Eメール本文の処理を可能にするDTDファイルのパスを指定します。

DTDファイルを使って添付ファイルの内容を処理する

- 1 [Eメールの添付ファイルの内容を処理する] オプションを選択します。
- 2 [DTDファイル] フィールドに、Eメール添付ファイルの処理を可能にするDTDファイルのパスを指定します。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

Eメール（送信）コネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 5.3. Eメール（送信）コネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]
接続を設定する [p. 163]
送信の頻度を定義する [p. 165]
ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続を設定する

[コネクタの設定] ウィザードの2ページ目では、Eメール（送信）コネクタのメッセージプロトコルを選択します。

このページに表示されるフィールドの数は、[メールエンジンの種類] フィールドで選択されたプロトコルの種類により変化します。Eメール（送信）コネクタで使用できるプロトコルは以下の通りです。

- Lotus VIM (Vendor Independent Messaging)
- Microsoft SMTP (Simple Mail Transfer Protocol)
- MAPI (Messaging Application Program Interface)

Lotus VIM

VIMプロトコルを選択する場合は、2つのフィールドに入力します。

- [ログイン]
コンピュータにインストールされているVIMメッセージシステムへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

SMTP

SMTPプロトコルを選択する場合は、4つのフィールドに入力します。

- [SMTPサーバ]
SMTPサーバの名前を指定します。
- [接続ポート]
SMTPサーバに使用されている接続ポートを指定します。このフィールドのデフォルト値は「25」です。
- [表示名]
メッセージの受信時に、メッセージの [送信者] フィールドに表示される名前を指定します。
- [返信アドレス]
メッセージの受信者が、返信する際に使用できるEメールアドレス（例えば Reply@peregrine.comのようなアドレス）を入力します。

MAPI

MAPIプロトコルを選択する場合は、2つのフィールドに入力します。

- [ログイン]
コンピュータにインストールされているMAPIメッセージシステムへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

[テスト] ボタン

このボタンを押すと接続をテストできます。

接続をテストするには、

- 1 接続用パラメータを入力します。
- 2 [テスト] をクリックします。

[接続のテスト] ウィンドウが開き、接続の成功または失敗を表示します。接続に失敗すると、失敗の原因が表示されます。

- 3 [閉じる] をクリックして設定ウィザードへ戻ります。

送信の頻度を定義する

[コネクタの設定] ウィザードの3ページ目では、一定数のメッセージがメッセージサーバに送信される毎に、送信を一時停止するよう指定できます。これにより、メッセージが継続的にサーバをふさぐことを回避できます。

このページには2つのフィールドがあります。

- [メッセージ数]

2回の一時停止時間の間にサーバに送信されるメッセージの数を指定します。例えば、メッセージ数として「100」を入力し「30s」の一時停止時間を入力すると、100のメッセージがコネクタに送信されると30秒停止し、また100メッセージが送信される、というようになります。

 **注意:**

[一時停止時間] フィールドと [メッセージ数] フィールドの適切な値は、メッセージシステムの性能に左右されます。

- [一時停止時間]

一時停止時間を入力する際には「s」を秒、「m」を分、「h」を時間用に使います。例えば、「100」のメッセージ用に「30s」の一時停止時間を入力すると、コネクタは100メッセージをメッセージサーバに送信してから30秒停止し、また100メッセージを送信する、というようになります。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

Eメール（受信）コネクタの生成用ドキュメントタイプ

Eメール（受信）コネクタはEメールを受信します。受け取った情報を基に、コネクタは使用可能なドキュメントタイプInMailMessageを発行します。このドキュメントタイプはEメールの様々なコンポーネントを含んでおり、別のコネクタの取り込み用ドキュメントタイプにマップされます。

Eメール（受信）コネクタの生成用ドキュメントタイプ InMailMessage

Eメール（受信）コネクタの生成用ドキュメントタイプのコンポーネントは次の表の通りです。

表 5.4. Eメール（受信）コネクタの生成用ドキュメントタイプ

生成用ドキュメントタイプの構成要素	要素
InMailMessageルートノード	メッセージ本文（【Body】フィールド） メッセージの日付（【Date】フィールド） メッセージのプライオリティ（優先順位）（【Priority】フィールド） メッセージの件名（【Subject】フィールド）
From構造体	メッセージ送信者のEメールアドレス（【Address】フィールド） メッセージ送信者の名前（【Name】フィールド） メッセージ送信者のタイプ（【Type】フィールド）
MailInfo構造体	【UniquelD】フィールドを含みます。このフィールドは、他のコネクタが送信する処理レポートによって使用されます。 これにより、コネクタが生成したメッセージを一意的な方法で識別できるようになります。
Attachmentコレクション	添付ファイルの内容は3つのフィールドに分かれます。 <ul style="list-style-type: none"> 添付ファイルのデータを含む【BlobFromMail】フィールド ファイル名を含む【FileFromMail】フィールド 添付ファイルのタイプを指定する【MimeType】フィールド

生成用ドキュメントタイプの構成要素	要素
Bccコレクション	メッセージのBcc受信者のEメールアドレス ([Address] フィールド) メッセージのBcc受信者の名前 ([Name] フィールド) Bccのタイプ ([Type] フィールド)
Ccコレクション	メッセージのCc受信者のEメールアドレス ([Address] フィールド) メッセージのCc受信者の名前 ([Name] フィールド) Ccのタイプ ([Type] フィールド)
Toコレクション	メッセージ受信者のEメールアドレス ([Address] フィールド) メッセージ受信者の名前 ([Name] フィー ルド) メッセージ受信者への送信のタイプ ([Type] フィールド)

Eメール (受信) コネクタに生成されるメッセージ本文の処理

DTDファイルをコネクタの設定中に指定すると (本節内の「Eメール (受信) コネクタの設定 [p.160]」参照)、Eメール (受信) コネクタに生成されるメッセージ本文を、解析できるようになります。この解析により、生成用ドキュメントタイプには新しい要素が追加されます。各要素は、DTDファイル内で定義された要素に一致します。

例 5.1. DTDにより拡張されたドキュメントタイプの例

Eメール (受信) - Asset Management 用例シナリオ (「newemplo.scn」ファイル) では、DTD「newemplo.dtd」が、InMailMessageドキュメント内に新規の構造体 (amEmplDept構造体) を追加します。

図 5.1. Eメールコネクタ - 拡張されていないInMailMessage ドキュメントタイプ

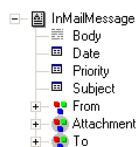
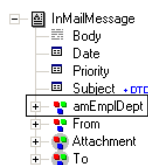


図 5.2. Eメールコネクタ - DTDにより拡張されたInMailMessage ドキュメントタイプ



Eメール（送信）コネクタの取り込み用ドキュメントタイプ

Eメール（送信）コネクタはEメールを送信します。コネクタは、Eメールの様々なコンポーネントを含む使用可能なドキュメントタイプOutMailMessage、を発行します。このドキュメントタイプの情報は、別のコネクタの生成用ドキュメントタイプのフィールドにマップされます。（例えばAsset Management - Eメール付属シナリオ（finreque.scn）の、Asset Managementコネクタ）

Eメール（送信）コネクタの取り込み用ドキュメントタイプ

Eメール（送信）コネクタの取り込み用ドキュメントタイプのコンポーネントは次の表の通りです。

表 5.5. Eメール（送信）コネクタの取り込み用ドキュメントタイプ

取り込み用ドキュメントタイプの構成要素	要素
OutMailMessageルートノード	メッセージ本文（【Body】フィールド） メッセージのプライオリティ（優先順位）（【Priority】フィールド） メッセージの件名（【Subject】フィールド）
Attachmentコレクション	添付ファイルの内容は3つのフィールドに分かれます。 <ul style="list-style-type: none"> 添付ファイルのデータを含む【BlobToSend】フィールド ファイル名を含む【FileToSend】フィールド
Bccコレクション	メッセージのBcc受信者のEメールアドレス（【Address】フィールド） メッセージのBcc受信者の名前（【Name】フィールド） Bccのタイプ（【Type】フィールド）
Ccコレクション	メッセージのCc受信者のEメールアドレス（【Address】フィールド） メッセージのCc受信者の名前（【Name】フィールド） Ccのタイプ（【Type】フィールド）
Toコレクション	メッセージ受信者のEメールアドレス（【Address】フィールド） メッセージ受信者の名前（【Name】フィールド） メッセージ受信者への送信のタイプ（【Type】フィールド）

添付ファイルの処理

添付ファイルは、メッセージに添付されたファイル（画像、ビデオ、音声、実行可能ファイルなど）を指します。Eメールコネクタの使用可能なドキュメントタイプでは、添付ファイルはAttachmentコレクションのフィールドとして表されています。

添付ファイルのマッピング

本節では、受信または送信される添付ファイルのマッピング方法を説明します。

受信される添付ファイルのマッピング

添付ファイルは、Eメールコネクタの生成用ドキュメントタイプ内の3つのフィールドで表されています。

- 添付ファイルのデータを含む [BlobFromMail] フィールド
- ファイル名を含む [FileFromMail] フィールド
- 添付ファイルのタイプを指定する [MimeType] フィールド

受信される添付ファイルをマップするには、 [BlobFromMail] フィールドを、ターゲットコネクタのバイナリタイプのフィールドへマップします。

Eメール（受信） - Asset Management用例シナリオ（newemplo.scn）では、 [BlobFromMail] フィールドは、Asset ManagementコネクタのamEmpDeptドキュメントタイプの [Photo.blobData] フィールドへマップされています。

amEmpDept	
bDepartment	[amEmpDept.bDepartment]
E Mail	[amEmpDept.E Mail]
FirstName	[amEmpDept.FirstName]
FullName	[amEmpDept.FullName]
IDNo	[amEmpDept.IDNo]
MiMrs	[amEmpDept.MiMrs]
Name	[amEmpDept.Name]
Phone	[amEmpDept.Phone]
Title	[amEmpDept.Title]
Photo	IF [Attachment(0).MimeType] <> "image"
blobData	[Attachment(0).BlobFromMail]
Name	[Attachment(0).FileFromMail]

送信される添付ファイルのマッピング

送信される添付ファイルは、Eメールコネクタの取り込み用ドキュメントタイプの2つのフィールドで表されています。

- 添付ファイルのデータを含む [BlobToSend] フィールド
- ファイル名を含む [FileToSend] フィールド

送信される添付ファイルのフィールドをマップするには、

- 1 ソースコネクタのバイナリフィールドを、 [BlobToSend] フィールドへマップします。
- 2 ソースコネクタのテキストフィールドを、 [FileToSend] フィールドにマップします。このフィールドは、添付ファイルの名前を含んでいなければなりません。

Asset ManagementコネクタのamAssetドキュメントタイプの [Photo.blobData] フィールドと [Photo.Name] フィールドは、 [BlobToSend] フィールドと [FileToSend] フィールドにマップされます。

付属シナリオ - Eメールコネクタ

このコネクタ用のシナリオはありません。

このコネクタを使用する付属シナリオは以下の通りです。

- mail /mailac36 /finconfi.scn [p. 310]
- mail /mailac36 /finreque.scn [p. 309]
- mail /mailac36 /newemplo.scn [p. 310]
- mail /mailac40 /finconfi.scn [p. 310]
- mail /mailac40 /finreque.scn [p. 311]
- mail /mailac40 /newemplo.scn [p. 311]

LDAPコネクタ

オプションコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

LDAPコネクタは、LDAPプロトコルを使用するサーバから送信されるデータを処理できます。例えば、コネクタはX500電子ディレクトリにアクセスできます。現バージョンのLDAPコネクタでは、LDAPソースから来るデータの読み取りと、LDAPソースへの書き込みが可能になっています。

LDAPコネクタの互換性

LDAPコネクタは、LDAP v2とLDAP v3プロトコルと互換性があります。

LDAPコネクタは、LDAP v3プロトコルの自動記述機能を使用し、ユーザに様々な情報 (オブジェクトクラスの定義、属性タイプの定義、名前付けコンテキスト、サポートされているコントロール、など) を提供します。

LDAPコネクタは、以下のアプリケーションとのテストに成功しています。

- Novell NetWare Directory Service
- OpenLDAP
- Microsoft ActiveDirectory
- iPlanet / Netscape Directory Server

LDAPコネクタ使用上の注意

LDAPコネクタの制約点は、使用するLDAPのリソース (割り当てられたCPUやメモリ、など) に左右されます。

コネクタのテスト中やシナリオの実行中にLDAPリソースの限界を超えると、データ一部または全体の処理に失敗する可能性があります。

LDAPコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 5.6. LDAPコネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]
LDAPコネクタの設定 [p. 172]
再接続のパラメータを設定する [p. 27]
サーバとの時間差を指定する [p. 29]
スケジュールのポイントを設定する [p. 32]
高度な設定 [p. 173]
キャッシュを設定する [p. 35]
ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

LDAPコネクタの設定

[基本設定] ウィザードの2ページ目では、LDAP接続を設定します。

このページのフィールドでは、次の内容を指定できます。

- LDAPサーバ名
- サーバの接続ポート
- ログイン
- パスワード
- サーバタイプ
- コードページ

[サーバ名]

LDAPサーバ名を入力します。サーバ名（例：ldap-server.unil.ch）または数値のIPアドレス（例：207.68.137.42）を入力できます。

[接続ポート]

LDAPサーバの接続ポートを入力します。デフォルト値「389」は多くの場合適切な値です。（この値はプロトコルに推奨されているデフォルト値です。）

[ログイン]

LDAPサーバへのアクセスを許可するログインを入力します。

[パスワード]

ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

[サーバタイプを指定してください]

以下の3タイプのLDAPサーバが使用可能です。

- Microsoft Exchange、Novell Directory Service、Netscape Directory Server、など
- Microsoft Active Directory
- OpenLDAP

[サーバのコードページを指定してください]

ドロップダウンリストから、LDAPサーバのコードページを選択します。

[テスト]

このボタンを押すと接続をテストできます。

接続をテストするには、

- 接続用パラメータを入力します。
- [テスト] をクリックします。

[接続のテスト] ウィンドウが開き、接続の成功または失敗を表示します。接続に失敗すると、失敗の原因が表示されます。

- [閉じる] をクリックして設定ウィザードへ戻ります。

高度な設定

このページでは以下の内容を指定します。

- ページサイズ
デフォルト値は500です。
このオプションの値は、MicrosoftのLDAPサーバでのみ使用されます。
- スキーマのDN（distinguished name、識別名）

稀に、スキーマの識別名がLDAPコネクタによって自動的に取得されないことがあります。

識別子が取得されなかった場合は、Connect-Itログ内に警告が表示されます。名前を取得するには、ルートDSE (Directory Service Entries) の subschemaSubentryエントリを参照してください。

- 日付形式

日付の形式については、本節内の「追加情報 [p. 181]」の節の「LDAP日付形式 [p. 183]」を参照してください。

このフィールドのデフォルト値は「%4Y%2M%2D%2H%2N%2SZ」です。

このフィールドで指定される日付と時刻の記号は以下の通りです。

- Y
年
- M
月
- D
日
- H
時間
- N
分
- S
秒
- Z

日付が「GMT0」形式であることを示す必須記号

% [数] [記号] 文字列は、記号の内容を表現するために使用される数字の数を指定します。例：「%4Y」は、年が4つの数字で表現されることを指定します (2003、1997など)。

サーバがミリ秒をサポートする場合、ミリ秒はピリオドで区切られた小数で表示されます。ピリオドの後の数字は、小数点以下の桁数を表しています。

例：「%4Y%2M%2D%2H%2N%2.1SZ」は、ミリ秒の精度が小数点以下1桁であることを表しています。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

[ルートDSEエントリ]

[ルートDSEエントリ] フィールドでは、LDAPツリー構造内のサーバの現在の日付に対応するエントリを入力できます。

デフォルトのエントリは「CurrentTime」です。

スケジュールのポインタを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポインタを設定する [p. 32]」の節を参照してください。

[最終変更のフィールド]

このフィールドには、LDAPコネクタがスケジュールのポインタとして使用するDSEのフィールドを指定します。このフィールドのデフォルト値は「modifyTimestamp」です。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

取り込み用ルール

コネクタの取り込み用ルールの入力方法については、「コネクタのルール (ディレクティブ) [p. 39]」章の「取り込み用ルール [p. 43]」の節を参照してください。

コネクタの取り込み用ルールを定義するには、以下の操作を行います。

- [整合性] と [高度な整合性チェック] タブ内で整合性用のパラメータを入力します。

これらの整合性用タブの詳細については、「コネクタのルール (ディレクティブ) [p. 39]」章の「取り込み用ルール [p. 43]」節内「整合性 [p. 44]」を参照してください。

- SCOPE句とSEARCH DN句を作成します。

SCOPE句

SCOPE句は、LDAPディレクトリのツリー構造内で、DN（識別名）により識別されるエン트리ポイントに対して、クエリのスコープ（適用範囲）を指定します。

- [基本スコープ]
このオプションを選択すると、クエリの対象はエン트리ポイントに含まれるデータのみになり、このエン트리ポイント下のサブツリー内は検索されません。
- [1レベルスコープ]
このオプションを選択すると、クエリの対象はエン트리ポイントに含まれるデータと、このエン트리ポイントすぐ下の子ノード内のデータになります。
- [サブツリースコープ]
このオプションを選択すると、クエリの対象はエン트리ポイントに含まれるデータと、このエン트리ポイント下のすべての子ノード内のデータになります。

SEARCH DN句

SEARCH DN句は、LDAPサーバの接続先のディレクトリのツリー構造内で、エン트리ポイントのDN（識別名）を指定します。コネクタの起動時に、ディレクトリは複数のエン트리ポイントを提案します。これらのエン트리ポイントは、LDAPサーバに表示される名前付けコンテキストに当たります。

名前付けコンテキストは、[SEARCHDN句]下のドロップダウンリスト内に表示されます。このフィールドに直接別のエン트리ポイントを入力することも可能です。各DNはRDN（相対的識別名）から成っています。以下の表は使用頻度の高いRDNをまとめています。

RDN（相対的識別名）	キー
CommonName	CN
LocalityName	L
StateOrProvinceName	ST
OrganizationName	O
OrganizationalUnitName	OU
CountryName	C
StreetAddress	STREET

 注意:

【SEARCHDN句】フィールドに値を入力しておくことをお勧めします。フィールドが空白であると全ディレクトリ内でクエリを実行することになり、LDAPサーバのダウンロード許容量を超過する恐れがあります。

LDAPコネクタの生成用ルール

LDAPコネクタの生成用ルールにより、LDAPコネクタは特定のクエリに従ってLDAPソースからデータを抽出します。

LDAPクエリには、他の生成用ドキュメントタイプのルールタブ内で作成する句とは違う種類の3つの句があります。

- SCOPE句
- SEARCH DN句
- FILTER句

SCOPE句

SCOPE句は、LDAPディレクトリのツリー構造内で、DN（識別名）により識別されるエントリポイントに対して、クエリのスコープ（適用範囲）を指定します。

- [基本スコープ]
このオプションを選択すると、クエリの対象はエントリポイントに含まれるデータのみになり、このエントリポイント下のサブツリー内は検索されません。
- [1レベルスコープ]
このオプションを選択すると、クエリの対象はエントリポイントに含まれるデータと、このエントリポイントすぐ下の子ノード内のデータになります。
- [サブツリースコープ]
このオプションを選択すると、クエリの対象はエントリポイントに含まれるデータと、このエントリポイント下のすべての子ノード内のデータになります。

SEARCH DN句

SEARCH DN句は、LDAPサーバの接続先のディレクトリのツリー構造内で、エントリポイントのDN（識別名）を指定します。コネクタの起動時に、ディレクトリは複数のエントリポイントを提案します。これらのエントリポイントは、LDAPサーバに表示される名前付けコンテキストに当たります。

名前付けコンテキストは、【SEARCHDN句】下のドロップダウンリスト内に表示されます。このフィールドに直接別のエントリポイントを入力することも可能です。各DNはRDN（相対的識別名）から成っています。以下の表は使用頻度の高いRDNをまとめています。

RDN（相対的識別名）	キー
CommonName	CN
LocalityName	L
StateOrProvinceName	ST
OrganizationName	O
OrganizationalUnitName	OU
CountryName	C
StreetAddress	STREET

 **注意:**

【SEARCHDN句】フィールドに値を入力しておくことをお勧めします。フィールドが空白であると全ディレクトリ内でクエリを実行することになり、LDAPサーバのダウンロード許容量を超過する恐れがあります。

FILTER句

FILTER句は、選択したエントリポイント下のデータにフィルタをかけます。FILTER句は、LDAPクエリに使用されるシンタックスに従う必要があります。このシンタックスはRFC2254内にあります。

例：名前が「A」で始まる人全員のリストを取得するには、以下の句を作成します。

```
(&(ObjectClass=personn)(cn=A*))
```

このリストに、名前が「Ar」で始まる人のリストを入れない場合は、以下の句を作成します。

```
(&(ObjectClass=personn)(&(cn=A*)!(cn=Ar*)))
```

💡 ヒント:

LDAPサーバがディレクトリ項目の変更日を管理するかどうかをテストするには、

- LDAPドキュメントタイプを作成します。
- FILTER句用に以下の値を入力します。

```
modifyTimestamp=*
```

これにより、[modifyTimestamp] フィールドが入力されているディレクトリの項目全体が取得されます。このフィールドが入力されていない項目がある場合、全項目を取得するためには起動時ごとにコネクタを設定しなければなりません。

LDAPフィルタのシンタックス

LDAPデータにフィルタを適用する際に使用されるシンタックスは、以下の表の通りです。

表 5.7. LDAPフィルタ

フィルタ

filter= "(" filtercomp ")"
filtercomp = and / or / not / item
and = "&" filterlist
or = " " filterlist
not = "!" filter
filterlist = 1*filter
item = simple / present / substring / extensible
simple = attr filtertype value
filtertype = equal / approx / greater / less
equal = "="
approx = "-="
greater = ">="
less = "<="
extensible = attr [":dn"] [": matchingrule"] ":=" value / [":dn"] [": matching rule"] ":=" value
present = attr "=*"
substring = attr "=" [initial] any [final]
initial = value any = "*" *(value "**")
final = value
attr = AttributeDescription (RFC 2251のセクション4.1.5)
matchingrule = MatchingRuleId (RFC 2251のセクション4.1.9)

フィルタ

value = AttributeValue (RFC 2251のセクション4.1.6)

LDAPコネクタが処理するドキュメントの値の中で、一部の文字を取得する方法は以下の表の通りです。

表 5.8. マップテーブルファイル (ASCII値の文字)

必要な文字	ASCII値
*	0x2a
(0x28
)	0x29
/	0x5c
NULL	0x00

文字は、「/」 (ASCII 0x5c) とエンコードされた文字のASCII値を表す2数値でコード化されなければなりません。16進数である2数値は大文字小文字のどちらでも構いません。例：RDNcnが文字*を含むかどうかを調べるには、以下のフィルタを作成します。

```
"(cn=* /2a*)"
```

フィルタの例

フィルタの目的	作成方法
全クラス内を検索する	(objectclass=*)
名前がAで始まる人をフィルタで抽出する	(& (objectclass=person)(cn=A*))
LDAP内で全タイプの人をフィルタで抽出する	((objectclass=person)(objectclass=organizationalPerson)(objectclass=inetOrgPerson)(objectclass=residentialPerson)(objectclass=newPilotPerson))
名前がAで始まる人を抽出し、そのうちArで始まる人は抽出しない	(& (objectclass=person)(& (cn=A*)(!(cn=Ar*))))
名前がH、Y、Eで始まらない人を抽出する場合、次のようには書けません。	(& (objectclass=person)(!(cn=H*)(cn=Y*)(cn=E*))) "not" ("!") は単項の演算子です。
名前がH、Y、Eで始まらない人を抽出する場合は、次のように書かなければなりません。	(& (objectclass=person)(& (!(cn=H*)(!(cn=Y*)(!(cn=E*)))))

追加情報

本節ではコネクタに関する追加情報を説明します。

LDAPプロトコルに関する情報元

LDAPプロトコルの詳細に関しては、以下のRFC (requests for comments) を参照してください。

- RFC 1274 : The COSINE and Internet X.500 Schema
- RFC 1777 : Lightweight Directory Access Protocol
- RFC 1778 : The String Representation of Standard Attribute Syntaxes
- RFC 1617 : Naming and Structuring Guidelines for X.500 Directory Pilots
- RFC 2253 : Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names
- RFC 2251 : Lightweight Directory Access Protocol (v3)
- RFC 2252 : Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions
- RFC 2254 : The String Representation of LDAP Search Filters
- RFC 2256 : A Summary of the X.500(96) User Schema for use with LDAPv3
- RFC 1823 : The LDAP Application Program Interface
- RFC 1798 : Connection-less Lightweight X.500 Directory Access Protocol
- RFC 2259 : Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2
- RFC 2279 : UTF-8, a transformation format of ISO 10646 RFC 2116 : X.500 Implementations Catalog-96
- RFC 2255 : The LDAP URL Format

コレクションの一意のフィールドを別のアプリケーションのフィールドにマップする

LDAPディレクトリのエントリは、多くの場合一意フィールドのコレクションです (つまり、コレクションの各構成要素は1つのフィールドに当たります)。

各フィールドは情報の種類 (性質) を表しており、1つまたは複数の値 (複数値のフィールド) を含んでいます。例: コレクションのフィールド値が1つのEメールアドレスである。

最良の処理方法は、このフィールドをターゲットアプリケーションのコレクション内のフィールドにマップすることです。

マッピングについては、『ユーザガイド』の「ドキュメントタイプのマッピング」の章を参照してください。

コレクションがない場合は以下の解決策があります。

- 1 コレクションの構成要素の内1つだけを別のアプリケーションのフィールドにマップする

- Basicスクリプトをマッピング内で使用する。このスクリプトはコレクションの全てのフィールド値を連結します。連結された値は、外部アプリケーションの一意フィールドにマップされます。

コレクションの構成要素の内1つだけをターゲットフィールドにマップする

コレクションのフィールドの内1つだけをターゲットフィールドにマップするには、

- マッピングの編集用ウィンドウ内で、LDAPドキュメントタイプのコレクションのフィールドを、ターゲットフィールドへマップします。
- ターゲットフィールドへマップするフィールドの番号を【マッピングスクリプト】フィールドに入力します。（注意：Connect-Itのコレクションの構成要素は0から番号が付けられています。第1の構成要素には番号0が、第2の構成要素には番号1がついています。）

例：「Person.cn」（人の通称）一意フィールドのコレクション内にある第1の構成要素を指定するには、[cn(0).cn]をマッピングスクリプト内に入力します。

Basicスクリプトの使用

Basicスクリプトを使用するには、

- マッピングの編集用ウィンドウ内で、LDAPドキュメントタイプのコレクションの一意フィールドを、ターゲットフィールドへマップします。
- 【マッピングスクリプト】フィールドにマッピングスクリプトを入力します。

スクリプトの前半部分ではコレクション内の構成要素の数を数え、後半部分では属性を連結して1つの文字列にします。シンタックスは以下の通りです。

```
Dim iCollectionCount As Integer
iCollectionCount = PifGetItemCount("cn")
Dim strCollapse As String
Dim iItem As Integer
For iItem = 0 to iCollectionCount - 1
    strCollapse = strCollapse + [cn(iItem)]
Next iItem
RetVal = strCollapse
```

最大のデータを含むLDAP要素を識別する

LDAPコネクタは、以下の情報を含む使用可能なドキュメントタイプを発行します。


- オブジェクトのクラス

- これらのクラスのフィールド
- クラス間の継承

LDAPデータソースの使用において最も重要な情報は以下の通りです。

- データを含むオブジェクトクラスが必要かどうか。
例：レコードを含むテーブル
- クラス間の継承を使うと、高性能なクエリを作成できるか。
例：Top、PersonまたはOrganizationalPersonクラスのうち、どれを使用すべきか。

これらの情報を取得するには、

- 1 シナリオ図部分でLDAPコネクタを選択するかまたは作成します。
- 2 **[ドキュメントタイプ]** タブを選択します。
- 3 ルート要素がTopである生成用ドキュメントタイプを作成します。
全LDAPディレクトリは、全サーバクラスでルート要素の役割を果たすオブジェクトクラスTopを発行します。
- 4  をクリックし、LDAPソースのデータを表示します。
- 5 最多のデータを返すクラスを探し、このクラスをドキュメントタイプの定義内で使用します。

LDAP日付形式

LDAPディレクトリ用の「日付と時刻」（timestamps）型フィールドの形式は、以下のシンタックスに従います。

```
[year][month][day][hour][minute][second]Z
```

例：2003年4月5日22時40分34秒は、文字列「20030405224034Z」で表現されません。

最後の「Z」は、日時がGMT 0（Greenwich Meridian Time）タイムゾーン上であることを示しています。

一部のLDAPサーバは、日付/時刻フィールド用に別の形式を使用します。例えば、ミリ秒に対応する数字を追加するサーバもあります（この数字はピリオド「.」で区切られます）。例：20030405224034.5Z

付属シナリオ - LDAPコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- `ldap/dapac36/complete.scn` [p. 308]
- `ldap/dapac36/simple.scn` [p. 308]

コマンドラインコネクタ

ベースコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

本節ではコマンドラインコネクタの内容を説明します。

- データの取り込み時に、コマンドラインコネクタは、Connect-Itがインストールされているコンピュータ上にコマンドラインを実行します。
- データの生成時に、コマンドラインコネクタは、Connect-Itのインストール先コンピュータで実行されたコマンドライン用に以下の要素を取得します。
 - 1 戻り値
 - 2 標準出力
 - 3 標準入力

コネクタはドキュメントを自動的に生成します。シナリオでは、コマンドラインコネクタがドキュメントを取り込む（他のコネクタがコマンドラインを発信することに相当します）たびに、対応するドキュメントを生成します。

コマンドラインコネクタの互換性

コマンドラインコネクタにより、Connect-Itがインストールされているオペレーティングシステム上にコマンドラインを起動できるようになっています。

UNIXでは、コネクタはコマンドの戻り値のコードのみを取得します。標準出力と標準エラーは取得されません。

コマンドラインコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 5.9. コマンドラインコネクタの設定

☐	コネクタに名前を付け、役割を説明する [p. 14]
☐	実行の頻度 [p. 185]
☐	ピボットドキュメントタイプの使用 [p. 36]

Connect-Itのコネクタの設定は、ウィザードで実行できます。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

実行の頻度

このページでは、一定数のドキュメントが処理される毎に一時停止するよう指定できます。これにより、コンピュータやサーバに過度の負担がかかるのを防ぐことができます。

このページには以下のフィールドがあります。

- [ドキュメントの数]

2回の一時停止時間の中にコンピュータが処理するドキュメントの数を指定します。例えば、ドキュメントの数として「100」を入力し「30s」の一時停止時間を入力すると、100ドキュメントがコネクタに処理されると30秒停止し、また100ドキュメントが処理される、というようになります。

注意:

入力できる[一時停止時間]のフィールド値と[ドキュメントの数]のフィールド値は、コンピュータの性能に左右されます。適切な値を入力するにはテストを実行する必要があります。

- [一時停止時間]

一時停止時間を入力する際には「s」を秒、「m」を分、「h」を時間用に使います。例えば、「100」のドキュメント用に「30s」の一時停止時間を入力すると、100ドキュメントがコネクタに処理されると30秒停止し、また100ドキュメントが処理される、というようになります。

- [タイムアウト]

[タイムアウトを過ぎたらプロセスを強制終了する]オプションを選択すると、このフィールドに入力できるようになります。タイムアウトを秒数で入力します。

警告:

UNIXでは、[タイムアウトを過ぎたらプロセスを強制終了する]オプションは使用不可能です。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

使用可能なドキュメントタイプ

コマンドラインコネクタには2つの使用可能なドキュメントタイプがあります。

- **Commandドキュメントタイプ (取り込み用)**
このドキュメントタイプは、Connect-Itがインストールされているコンピュータ上に実行されるコマンドラインに相当します。この使用可能なドキュメントタイプにより、コマンドラインコネクタの取り込み用ドキュメントタイプを指定できます。
- **CommandReturnドキュメントタイプ (生成用)**
このドキュメントタイプは、Connect-Itがインストールされているコンピュータ上に実行されるコマンドラインの戻り値に相当します。この使用可能なドキュメントタイプにより、コマンドラインコネクタの生成用ドキュメントタイプを指定できます。

Commandドキュメントタイプ

このドキュメントタイプには3つのフィールドがあります。

- **[Arg] フィールド**
このフィールドはコマンドラインの引数に当たります。
- **[Name] フィールド**
このフィールドは実行可能ファイル (ターゲット) の完全パスに当たります。例 : C:Program Files/Peregrine/AssetCenter/
実行可能ファイルがオペレーティングシステムのPath環境変数内に定義されている場合は、実行可能ファイル名を直接指定できます。例 : notepad.exe
- **[Path] フィールド**
このフィールドはコマンドラインの実行フォルダに当たります。フィールド **[作業フォルダ]** に当たります。

CommandReturnドキュメントタイプ

このドキュメントタイプには3つのフィールドがあります。

- **[ReturnCode] フィールド**
このフィールドは、実行されたコマンドの戻り値のコードに当たります。
- **[StdErr] フィールド**
このフィールドにより、標準のエラー出力に関するデータを取得できます。
- **[StdOut] フィールド**
このフィールドにより、標準の出力のデータを取得できます。

🚩 重要項目:

UNIXでは「StdErr」と「StdOut」フィールドは使用不可能です。

取り込み用ルール (ディレクティブ)

コマンドラインコネクタには、ルール (ディレクティブ) が1つあります。このルールは、コネクタの取り込み用ドキュメントタイプの編集用ウィンドウにある [同期実行] オプションです。同期モード (デフォルトのモード) では、コマンドラインは前のコマンドラインの実行が終了してから次のコマンドラインを実行します。コマンドライン実行から発生する全メッセージ (エラーメッセージも含む) は、コネクタに処理されます。

付属シナリオ - コマンドラインコネクタ

このコネクタ用のシナリオはありません。

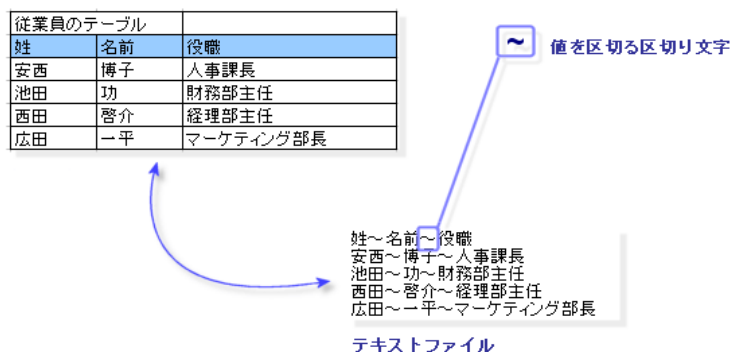
テキストコネクタ

ベースコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

テキストコネクタは、データベースのレコードを単純なテキスト形式で記録するファイル进行处理します。データベース内テーブルの各フィールド値は、テキストファイル内では、固定幅 (文字数で表現)、または区切り文字 (句読点、タブ、またはその他の文字) で区切られています。

図 5.3. テキストコネクタ - テキストファイル内のデータの記録法



テキストコネクタは以下のモードで機能します。

- 読み取りモード
読み取りモードでは、テキストコネクタはテキストファイルのデータを読み取ります。そして、Connect-Itコネクタ（データベースコネクタ、Asset Managementコネクタ、ServiceCenterコネクタなど）が使用できるデータベース内に、テキストファイルのデータを送信します。
- 書き込みモード
書き込みモードでは、テキストコネクタはテキストファイルを作成します。作成されるテキストファイル内のデータは、Connect-Itコネクタ（データベースコネクタ、Asset Managementコネクタ、ServiceCenterコネクタなど）が使用できるデータベースから来るものです。

テキストコネクタでは、以下の操作が可能です。

- Connect-Itがインストールされているコンピュータやネットワーク上にあるテキストファイルを処理する。
これらのファイルはgzipコマンドで圧縮できます。
- FTPサイトやWebサイトにあるテキストファイルを処理する（Webサイトのテキストファイルでは読み取りのみ可能）。

テキストコネクタの既知の制約点

テキストコネクタでの唯一の制約点は、「日付」型と「日付+時刻」型フィールドのフォーマットに関するものです。これらのフィールドを適切に処理するには、フィールドのフォーマットが、Connect-Itがインストールされているコンピュータの地域設定に一致しなければなりません。

「文字列」型のフィールドを使用し、マッピング時にこれをユーザフォーマットに関連付けると、この制約点を回避できるようになります。

ユーザフォーマットの詳細については、『ユーザガイド』の「マッピングスクリプト」の章の「ユーザフォーマット」の節を参照してください。

テキストコネクタの設定（読み取り）

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 5.10. テキストコネクタの設定（読み取り）

コネクタに名前を付け、役割を説明する [p. 14]	
処理モードを選択する [p. 190]	
接続のプロトコルを選択する [p. 190]	
ファイルまたはフォルダを選択する [p. 190]	HTTP Webサイトへの接続 [p. 191]
FTPサーバ接続 [p. 193]	
処理後のアクション [p. 191]	処理後のアクションを定義する [p. 194]
書式設定ファイル（DSCファイル）を選択する [p. 194]	
ピボットドキュメントタイプの使用 [p. 36]	

警告:

ファイルまたはフォルダのパス

コネクタの設定時に、コンピュータのネットワークドライブの文字に関連付けられたフォルダ内に位置する、ファイルまたはフォルダのパスを指定する場合は、パスにネットワークドライブの文字を入力せずに、ドライブの文字に関連付けられたフォルダを入力します。

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、『ユーザガイド』の「統合シナリオをプロダクションモードにする」の章、「Connect-Itサービスの作成」 / 「WindowsでConnect-Itサービスを作成する」 / 「シナリオのパス」を参照してください。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

処理モードを選択する

このページでは、コネクタが読み取りモードまたは書き込みモードのどちらで使用されるのかを指定できます。

接続のプロトコルを選択する

以下の3つのオプションがあります。

- [HTTP Webサイト]
- [FTPサーバ]
- [ローカルファイル/ネットワークファイル]

ファイルまたはフォルダを選択する

ローカルファイルまたはネットワーク上のファイルを読み取る場合、以下の操作を実行します。

- 1 ファイルまたはフォルダを選択する
- 2 ファイル処理後のアクションを選択する

ファイルまたはフォルダを選択する

このページでは、コネクタが読み取るテキストファイルを選択します。以下の2つのオプションがあります。

- [ファイルを読み取る]
- [フォルダを読み取る]

[ファイルを読み取る]

このオプションを選択する場合は、コンピュータやネットワーク上にあるテキストファイルのパスを指定します。

[フォルダを読み取る]

このオプションを選択する場合は、以下の2フィールドに入力します。

- [フォルダ名]
ファイルを含むフォルダのパスを入力します。
- [拡張子]
コネクタが読み取るファイルの拡張子を入力します。例：「txt」

[サブフォルダを読み取る]

このオプションを選択すると、コネクタは、選択したフォルダのサブフォルダ内のファイルも読み取ります。

処理後のアクション

テキストコネクタがファイルを読み取った後のアクション用に、3つのオプションがあります。

- [フォルダに保存する]
- [フォルダから削除する]
- [次のフォルダに移す]

3番目のオプションを選択する場合は、処理されたファイルの移動先フォルダのパスを入力します。

コネクタがテキストファイルの処理に成功または失敗した場合用に、上記の3オプションから1つを選択します。

シナリオ内の他のコネクタやマッピングボックスに取り込まれるドキュメントに、処理後のアクションが実行されるようにするには、各コネクタやマッピングボックスが生成する処理レポートを使用する必要があります。詳細については、『ユーザガイド』の「**処理レポート**」の章を参照してください。

HTTP Webサイトへの接続

WebサイトにあるXMLファイルの読み取りを選択する場合は、HTTP接続パラメータを入力します。

- [アドレス]
[プロトコル]://[アドレス]:[ポート]/[パス]のようにアドレスを入力します。
[アドレス]の部分のみが必須の場合もあります。
HTTPサーバ用の通常のポート番号は80です。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

[セキュリティで保護された接続 (HTTPSプロトコル)]

このオプションでは、セキュリティで保護された接続 (HTTPS) でサイトに接続するかどうかを指定できます。

重要項目:

Webサイトアドレスの [プロトコル] の部分がHTTPSプロトコルでない場合でも、 [**セキュリティで保護された接続 (HTTPSプロトコル)**] オプションを使用すると、HTTPSプロトコルが強制的に使用されます。

HTTPSサーバ用の通常のポート番号は443です。

HTTPアドレスの例

アドレス	セキュリティで保護された接続 (HTTPS) のオプション	接続モード
https://a.b.c.d/	選択	https port 443
https://a.b.c.d:9000/	選択	https port 9000
http://a.b.c.d:9000/	非選択	https port 9000
http://a.b.c.d:9000/	選択	https port 9000
https://a.b.c.d:9000/	非選択	https port 9000
http://a.b.c.d:443/	非選択	https port 443
https://a.b.c.d:80/	選択	https port 80

[クライアント証明書]

このフィールドでは、コンピュータに既存のHTTPS証明書の中から1つを選択します。

重要: コネクタがドキュメントを読み込んだ後またはドキュメントに書き込んだ後に、このフィールドに指定されているクライアント証明書を変更したり削除したりする場合は、シナリオビルダを一度終了してから再起動して、この変更事項を有効にします。

クライアント証明書のリストを管理する

設定ウィザードにあるクライアント証明書のリストは、Microsoft Managementアプリケーションの「コンソールルート / 証明書 - 現在のユーザー / 個人 / 証明書」フォルダにあるリストと一致します。

Windows 2000でこのフォルダに証明書を追加するには、

- 1 Windowsの [スタート] メニューで [**ファイル名を指定して実行**] を選択します。
- 2 [**名前**] フィールドに「MMC」と入力します。
- 3 [**コンソール / スナップインの追加と削除**] を選択します。
- 4 表示されるウィンドウで [**追加**] をクリックします。

- 5 表示されるウィンドウで **[証明書]** を選択します。
- 6 **[追加]** をクリックします。
- 7 表示されるダイアログボックスで **[ユーザーアカウント]** オプションを選択します。
- 8 **[完了]** をクリックします。
- 9 **[OK]** をクリックします。
- 10 「**証明書 - 現在のユーザー / 個人 / 証明書**」フォルダ内でファイルを追加または削除します。

FTPサーバ

FTPサイトにあるテキストファイルの読み取りを選択する場合は、以下の操作を実行します。

- 1 FTP接続パラメータを選択する
- 2 ファイル処理後のアクションを選択する

FTPサーバ接続

このページでは、FTPサーバへの接続用に以下の3つのフィールドに入力します。

- **[サーバ]**
FTPサーバ名を指定します。
- **[ログイン]**
サイトへのアクセスを許可するログインを入力します。
- **[パスワード]**
ログイン用のパスワードを入力します。

[ファイルを読み取る]

このオプションを選択する場合は、コンピュータやネットワーク上にあるテキストファイルのパスを指定します。

[フォルダを読み取る]

このオプションを選択する場合は、以下の2フィールドに入力します。

- **[フォルダ名]**
ファイルを含むフォルダのパスを入力します。
- **[拡張子]**
コネクタが読み取るファイルの拡張子を入力します。

[サブフォルダを読み取る]

このオプションを選択すると、コネクタは、選択したフォルダのサブフォルダ内のファイルも読み取ります。

注意:

FTPモードでの接続中にエラー「Error: 12015」が発生する可能性があります。このエラーは、同時に多数の接続を実行しているためディレクトリにアクセスできないことを示しています、

処理後のアクションを定義する

テキストコネクタがファイルを読み取った後のアクション用に、3つのオプションがあります。



- [フォルダに保存する]
- [フォルダから削除する]
- [次のフォルダに移す]

3番目のオプションを選択する場合は、処理されたファイルの移動先フォルダのパスを入力します。

コネクタがテキストファイルの処理に成功または失敗した場合用に、上記の3オプションから1つを選択します。

書式設定ファイル (DSCファイル) を選択する

テキストファイルを処理する際、コネクタは書式設定ファイルを使用します。

- 書式設定ファイルが既に存在する場合
[DSCファイル] フィールド内に完全パスを入力します。  をクリックしてファイルの場所を検索することも可能です。
- 書式設定ファイルが存在しない場合
作成する書式設定ファイルの名前とパスを [DSCファイル] フィールド内に入力し、  をクリックします。すると、書式設定ファイル作成用のウィザードが起動します。このウィザードについては、本節の「書式設定ファイル (DSCファイル) の作成 [p. 201]」を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

テキストコネクタの設定（書き込み）

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

テキストコネクタの設定（書き込み）の内容は、選択する接続プロトコルにより変わります。

表 5.11. テキストコネクタの設定（書き込み）

コネクタに名前を付け、役割を説明する [p. 14]	
処理モード [p. 196]	
接続のプロトコルを選択する [p. 196]	
ファイルまたはフォルダ <input type="checkbox"/> HTTP Webサイト [p. 196] <input type="checkbox"/> FTPサーバ接続 [p. 198] を選択する [p. 197]	
2セッション間の設定 [p. 199]	2セッション間の設定 [p. 199]
書式設定ファイル（DSCファイル）を選択する [p. 199]	
ピボットドキュメントタイプの使用 [p. 36]	

警告:

ファイルまたはフォルダのパス

コネクタの設定時に、コンピュータのネットワークドライブの文字に関連付けられたフォルダ内に位置する、ファイルまたはフォルダのパスを指定する場合は、パスにネットワークドライブの文字を入力せずに、ドライブの文字に関連付けられたフォルダを入力します。

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、『ユーザガイド』の「統合シナリオをプロダクションモードにする」の章、「Connect-Itサービスの作成」 / 「WindowsでConnect-Itサービスを作成する」 / 「シナリオのパス」を参照してください。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

処理モード

このページでは、コネクタが読み取りモードまたは書き込みモードのどちらで使用されるのかを指定できます。

接続のプロトコルを選択する

以下の3つのオプションがあります。

- [HTTP Webサイト]
- [FTPサーバ]
- [ローカルファイル/ネットワークファイル]

HTTP Webサイト

HTTPサイトにファイルを書き込む場合は、以下の操作を実行します。

- HTTP接続パラメータを選択する
- ファイル処理後のアクションを選択する

HTTP Webサイトへの接続

このページでは、HTTPサーバへの接続用に以下の3つのフィールドに入力します。

- [アドレス]
HTTPサーバへの接続用アドレスを指定します。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

HTTPSプロトコル

セキュリティで保護されたプロトコルを使用するサーバへ接続する場合は、このオプションを選択します。

HTTPSサーバ用の通常のポート番号は443です。

HTTPサーバ用の通常のポート番号は80です。

[クライアント証明書]

このフィールドでは、コンピュータに既存のHTTPS証明書の中から1つを選択します。

重要：コネクタがドキュメントを読み込んだ後またはドキュメントに書き込んだ後に、このフィールドに指定されているクライアント証明書を変更したり削

除したりする場合は、シナリオビルダを一度終了してから再起動して、この変更事項を有効にします。

クライアント証明書の一覧を管理する

設定ウィザードにあるクライアント証明書の一覧は、Microsoft Managementアプリケーションの「コンソールルート / 証明書 - 現在のユーザー / 個人 / 証明書」フォルダにある一覧と一致します。

Windows 2000でこのフォルダに証明書を追加するには、

- 1 Windowsの【スタート】メニューで【ファイル名を指定して実行】を選択します。
- 2 【名前】フィールドに「MMC」と入力します。
- 3 【コンソール/スナップインの追加と削除】を選択します。
- 4 表示されるウィンドウで【追加】をクリックします。
- 5 表示されるウィンドウで【証明書】を選択します。
- 6 【追加】をクリックします。
- 7 表示されるダイアログボックスで【ユーザーアカウント】オプションを選択します。
- 8 【完了】をクリックします。
- 9 【OK】をクリックします。
- 10 「証明書 - 現在のユーザー / 個人 / 証明書」フォルダ内でファイルを追加または削除します。

書き込み用のコマンド

HTTPサーバへの書き込みには、以下の2タイプのコマンドが使用されます。

- POST
指定されたアドレスに位置するプログラムヘッダを送信します。PUTメソッドと異なり、送信されたデータは処理されなければなりません。
- PUT
データを送信し、指定されたアドレスにデータが保存されるようにします。
HTTPサーバ用のコマンドを選択します。

ファイルまたはフォルダを選択する

ローカルまたはネットワーク上にファイルを書き込む場合、以下の操作を実行します。

- 1 ファイルまたはフォルダを選択する
- 2 ファイル処理後のアクションを選択する

ファイルまたはフォルダを選択する

このページでは、コネクタが書き込むテキストファイルを選択します。以下の2つのオプションがあります。

[1つのファイルにドキュメントを書き込む]

ローカルコンピュータまたはネットワーク上のテキストファイルのパス名を指定します。このファイルが既に存在すると、コネクタの起動時に既存ファイルの内容は上書きされます。

[各ドキュメントごと別々のファイルに書き込む]

ファイルの書き込み先フォルダ名を指定します。ファイル名は、テキストコネクタの取り込み用ルール内に指定した汎用名に一致します（本節の「テキストコネクタ（書き込み）の取り込み用ルール [p. 207]」を参照してください）。

FTPサーバ

FTPサイトにテキストファイルを書き込む場合は、以下の操作を実行します。

- 1 FTP接続パラメータを選択する
- 2 ファイル処理後のアクションを選択する

FTPサーバ接続

このページでは、FTPサーバへの接続用に以下の3つのフィールドに入力します。

- [サーバ]
FTPサーバ名を指定します。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

[1つのファイルにドキュメントを書き込む]

テキストコネクタで作成するファイルのパス名を指定します。このファイルが既に存在する場合は、既存ファイルの内容が上書きされます。

[各ドキュメントごと別々のファイルに書き込む]

ファイルの書き込み先フォルダ名を指定します。ファイル名は、テキストコネクタの取り込み用ルール内に指定した汎用名に一致します（本節の「テキストコネクタ（書き込み）の取り込み用ルール [p. 207]」を参照してください）。

2セッション間の設定

このページでは、データ書き込みの2セッション間のドキュメント処理を選択します。1つのファイルにデータを書き込むか、または1つのフォルダの複数のファイルにデータを書き込むかにより、動作オプションは変化します。

[1つのファイルにドキュメントを書き込む]

1つのファイルに書き込む場合は、以下のオプションがあります。

- [同じファイルの後に書き込む]
このオプションを選択すると、コネクタは、前回のセッションが中断した時に開いていたファイル内に、データの書き込みを再開します。
- [前回の全ファイルに上書きする]
コネクタは、前回のセッション時にデータを書き込んでいたファイルを削除します。
- [異なるファイルに番号を付ける]
コネクタはファイルを作成し、前回のファイル名を基に名前を付けます。
例：ファイル.xml、ファイル1.xml、ファイル2.xmlなど。

[各ドキュメントごと別々のファイルに書き込む]

各ドキュメントごと別々のファイルに書き込む場合は、以下のオプションがあります。

- [前回の全ファイルを削除し、番号付けを再開する]
フォルダ内の全ファイルが削除されます（前回のセッションのみでなく、前回までの全セッションのファイルが削除されます）。
コネクタは、書き込む全ファイルへの番号付けを開始します。
- [ファイルの番号付けを続行する]
コネクタはファイルへの書き込みを再開し、既存ファイルを削除せずに番号を付けます。

書式設定ファイル（DSCファイル）を選択する

テキストファイルを処理する際、コネクタは書式設定ファイルを使用します。

- 書式設定ファイルが既に存在する場合
[DSCファイル] フィールド内に完全パスを入力します。🔍をクリックしてファイルの場所を検索することも可能です。
- 書式設定ファイルが存在しない場合
作成する書式設定ファイルの名前とパスを [DSCファイル] フィールド内に入力し、🔍をクリックします。すると、書式設定ファイル作成用のウィザード

ドが起動します。このウィザードについては、本節の「**書式設定ファイル (DSCファイル) の作成** [p. 201]」を参照してください。

ブラインドモード

[**ブラインドモードを使用する**] オプションを選択すると、テキストコネクタは、書式設定ファイルなしでテキストファイルを作成します。作成されるテキストファイル内の値は、選択した区切り文字で区切られます。

区切り文字には以下のものがあります。

- [**タブ**]
 - [**コンマ**]
 - [**セミコロン**]
 - [**スペース**]
 - [**その他**]
- [**その他**] フィールドに、区切り文字として使用する記号を入力します。

ブラインドモードでのテキストコネクタの機能

ブラインドモードでテキストコネクタを使用するには、

- [**ブラインドモードを使用する**] オプションを選択し、テキストコネクタを設定します。
- データベース型のコネクタ（例：Asset Managementコネクタ）を選択し、このコネクタ用に生成用ドキュメントタイプを作成します。ターゲットのテキストファイル内に作成する値を含むフィールドを、この生成用ドキュメントタイプ内で選択します。

注意:

ブラインドモードでは、生成用ドキュメントタイプのルートノードのすぐ下にあるフィールド値のみが、ファイルテキストに書き込まれます。生成用ドキュメントタイプのルートノード下に、構造体やコレクションがあると、構造体とコレクション内の値はテキストファイルには書き込まれません。

- シナリオ図内で、**マッピングボックスを通さず**にデータベース型のコネクタをテキストコネクタに接続します（[**Shift**] キーを押したままコネクタ間のリンクを作成すると、マッピングボックスなしの直接リンクが作成されます）。
- データベース型のコネクタを生成モードにします（[**ツール / 生成する**] コマンド）。

ブラインドモードで生成されるテキストファイル

[1つのファイルにドキュメントを書き込む] オプションが [基本設定] ウィザードで選択されている場合、

- ブラインドモードで作成されるファイル内の各行は、テキストコネクタに取り込まれた各ドキュメントに対応します。
- ブラインドモードで作成されるファイルには、ソースコネクタの生成用ドキュメントタイプと同じ名前が付けられます。

[各ドキュメントごと別々のファイルに書き込む] オプションが [基本設定] ウィザードで選択されている場合、

- テキストコネクタに取り込まれる各ドキュメントの値は、ブラインドモードで作成される各ファイルに書き込まれます。
- ブラインドモードで作成されるファイルには、以下の2要素から成る名前が付けられます。
 - ソースコネクタの生成用ドキュメントタイプの名前 (例: Test)
 - ドキュメントが生成される順番に対応する番号 (例: test_01、test_02、test_03)

書式設定ファイル (DSCファイル) を書き込む

[DSCファイルを書き込む] オプションを使用すると、ブラインドモードで作成されたテキストファイルのデータに基づいて、DSCファイルを作成、保存できます。[ブラインドモードを使用する] が選択されていると、このオプションはデフォルトで選択されます。

既存するファイル名を使用すると、Connect-Itは、自動的に「_01」を追加してから保存します。書式設定ファイルは、ブラインドモード用に選択した区切り文字を使用します。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

書式設定ファイル (DSCファイル) の作成

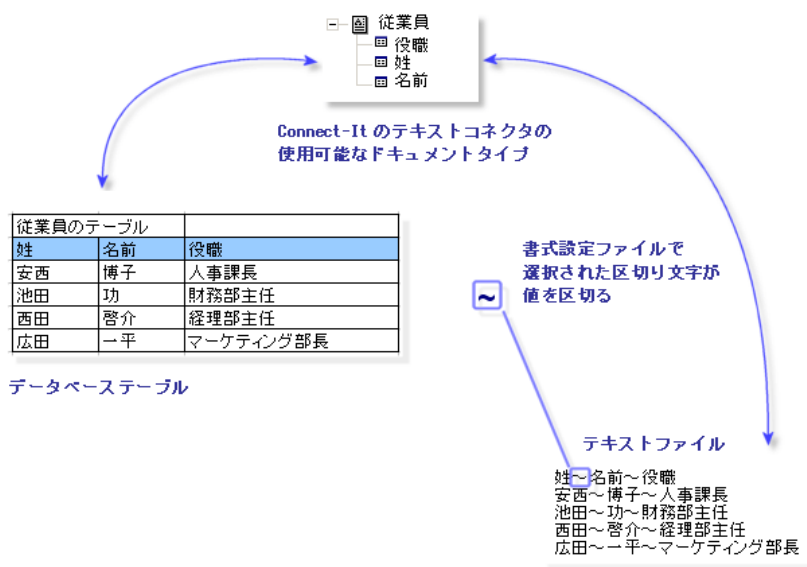
書式設定ファイル (ファイルの拡張子「.dsc」) は、テキストファイル内のデータがどのように構成されているかを説明するファイルです。ファイル内のデータは、データベースのテーブル内のフィールド値に当たります。


テキストコネクタの使用可能なドキュメントタイプは、書式設定ファイル内で作成されたドキュメントタイプです。

例: 書式設定ファイルで以下の内容を指定します。






- テキストファイルは1つのテーブル（従業員のテーブル）のデータを含みません。
- テキストファイルの1行目では、列のタイトルを指定します（各列はテーブルのフィールド1つに対応します）。
- 値は「~」で区切ります。

図 5.4. テキストコネクタ - データ処理



書式設定ファイル作成用のウィザードにアクセスするには、【コネクタの設定】ウィザードの【書式設定ファイル（DSCファイル）を選択する】ページで、をクリックします。


書式設定ファイルの設定ウィザードの各ページは以下の通りです。

-  ドキュメントタイプ [p. 203]
-  プレビューするファイルの選択 [p. 203]
-  区切り文字または固定幅 [p. 203]
-  データ処理のオプション [p. 204]
-  列 [p. 206]


ドキュメントタイプ

このページでは、テキストコネクタの使用可能なドキュメントタイプを作成できます。各ドキュメントタイプはデータベース内の1テーブルに当たります。

ドキュメントタイプを作成するには、

-  をクリックします。
- **[ドキュメントタイプ]** 列内のテキストゾーンをクリックし、作成するドキュメントタイプの名前を入力します。
- **[次へ]** をクリックして、次のページへ進みます。

ドキュメントタイプを削除するには、

- ドキュメントタイプを選択します。
-  をクリックします。

プレビューするファイルの選択

このページでは、区切り文字で区切られたテキストファイルを指定します。ウィザードは、書式設定ファイルがこのファイルのデータを処理する方法をプレビュー表示します。

処理するプレビューファイル

スペース、タブ、特殊文字などで区切られたデータを含むファイルを指定します。このファイルは、シナリオ内のコネクタが処理するファイルに相当します。

ヒント:

データのプレビュー表示を効率よく実行するには、以下のファイルをプレビューに使用するようにお勧めします。

- シナリオの実行時にコネクタが処理するファイルのローカルコピー
- 小さなサイズのファイル (サイズに特に制限はありません)

プレビューする行の数

ウィザードの次のページの **[データのプレビュー]** で表示する行の数を指定します。

区切り文字または固定幅

このページでは、テキストファイル内のデータを区切る方法を指定します。以下の2つのオプションがあります。

- **[固定幅]**

このオプションを選択すると、ウィザードの【列】ページの【幅】フィールドで、テキストファイルの各列用に固定幅を指定できます。

列を作成するには、プレビューゾーンを直接クリックします。

列の幅を調整するには、列をクリックしドラッグして移動させます。

列を削除するには、列をクリックしてプレビューゾーン外へドラッグします。

- 【区切り文字】

このオプションを選択すると、テキストファイル内のデータ値用に区切り文字を指定できます。以下の区切り文字が使用可能です。

- 【タブ】
- 【コンマ】
- 【セミコロン】
- 【スペース】
- 【区切り文字として使用する文字のリスト】

区切り文字として使用する文字を指定します。このリスト内で文字を区切るための句読点記号はありません。

データ処理のオプション

このページでは、区切られたテキストファイルのデータを処理する際のオプションを指定できます。

ここでは、セミコロンで区切られた次のテキストを使って、ウィザードの処理オプションを説明します。

```
名前;メーカー;ID
Inspiron;Dell;Comp111
"Inspiron";Dell;Comp112
"Inspiron";Dell;Comp113
"Inspiron" ;Dell ;Comp114
// "Inspiron";Dell;Comp115
Inspiron /;Dell;Comp /,116
```

第1行目から列のタイトルをインポートする

このオプションを選択すると、区切られたテキストファイルの1行目の値は、列のタイトルとして使用されます。

	名前	メーカー	ID	列4
1	Inspiron	Dell	Comp111	
2	"Inspiron"	Dell	Comp112	
3	'Inspiron'	Dell	Comp113	
4	"Inspiron"	Dell	Comp114	
5	\\"Inspiron\"	Dell	Comp115	
6	Inspiron\		Dell	Comp\

引用符（クォーテーションマーク）

このフィールドには、テキスト内で使用されている引用符の文字を指定します。

- 一重引用符（'）
- 二重引用符（"）
- 区切り文字として選択した文字以外の全記号

二重引用符を指定すると、次の結果が得られます。

	名前	メーカー	ID	列4
1	Inspiron	Dell	Comp111	
2	Inspiron	Dell	Comp112	
3	'Inspiron'	Dell	Comp113	
4	Inspiron	Dell	Comp114	
5	\\"Inspiron\"	Dell	Comp115	
6	Inspiron\		Dell	Comp\

[値の前後の引用符を保持する] オプションを選択すると、上記で選択した引用符はプレビュー枠内に表示されます。

コメント行の開始

区切られたテキストファイル内のコメントに、印をつけるための文字列を指定します。デフォルト値は//です。

プレビュー枠内でコメントは反転表示されます。

この例で値「//」を入力すると、以下の結果が得られます。

	名前	メーカー	ID	列4
1	Inspiron	Dell	Comp111	
2	"Inspiron"	Dell	Comp112	
3	'Inspiron'	Dell	Comp113	
4	"Inspiron"	Dell	Comp114	
5	//Inspiro	n",Dell;Co	mp115	
6	Inspiron\		Dell	

エスケープ文字

エスケープ文字を入力します。


この例で値「/」を入力すると、以下の結果が得られます。

	名前	メーカー	ID
1	Inspiron	Dell	Comp111
2	"Inspiron"	Dell	Comp112
3	'Inspiron'	Dell	Comp113
4	"Inspiron"	Dell	Comp114
5	//Inspiro	n",Dell;Co	mp115
6	Inspiron;	Dell	Comp;116

列

テキストファイル内の各列は、データベースのテーブル内のフィールドに当たります。つまり、テキストファイル内の各列は、各書式設定ファイル用に作成されるドキュメントタイプのフィールドに当たります。

このページでは、

- テキストファイルの各列を作成し、列に名前を付けます。
 - 列を作成するには、
 -  をクリックします。
 - **[名前]** 列内のテキストゾーンをクリックし、作成する列の名前を入力します。
 - 作成する各列用のデータ値のタイプを指定します。

データ値タイプを入力するには、[タイプ]列をクリックし、ドロップダウンリストから1項目を選択します。使用可能なタイプは、[テキスト]、[数値]、[日付]、[日付と時刻]と[通貨]です。

- [区切り文字または固定幅]ページで[固定幅]オプションが選択されている場合は、各列の固定幅を指定します。

固定幅値を入力するには、[幅]列をクリックし必要な文字数を入力します。

列を削除するには、

- 列を選択します。
- × をクリックします。

列の並べ替え

列の順番は、テキストファイル内でのフィールドの順番に一致しなければなりません。

列を並べ替えるには、

- 列を選択します。
- この列を ↓ または ↑ をクリックして移動させます。

注意:

列の作成や列の名前の変更が可能であっても、区切り文字モードではこれらの操作は無意味です。

テキストコネクタ（書き込み）の取り込み用ルール

テキストコネクタの取り込み用ドキュメントタイプに適用されるルール（ディレクトティブ）を指定できます。

- [汎用名]

このフィールドのデフォルト値は、テキストコネクタの取り込み用ドキュメントタイプの名前です。

- [ファイルの拡張子]

拡張子はピリオドなしで入力します。例えば「txt」が正しい入力方法で、「.txt」は間違いです。デフォルトでは「txt」が入力されています。

テキストコネクタの設定時に、[各ドキュメントごと別々のファイルに書き込む]を選択した場合のみ、これらの2フィールドを使用します。

詳細については、「[テキストコネクタの設定（書き込み）](#) [p. 195]」の節の「[ファイルまたはフォルダを選択する](#) [p. 198]」を参照してください。

シナリオの起動時に書き込まれるファイルの名前は、汎用名、ファイル作成の順番に一致する番号（_01、_02、_03など）と、指定されたフィルの拡張子から構成されます。

例：「employee」と「txt」を入力すると、作成されるファイルの名前は、employee_01.txt、employee_02.txt、employee_03.txtなどとなります。

 **警告:**

テキストコネクタに複数の取り込み用ドキュメントタイプがある場合、同じ汎用名を使用してはなりません。同じ汎用名が使用されていると、先に作成されたファイルが、後に作成されたファイルに削除され上書きされます。

図 5.5. テキストコネクタ - 取り込み用ルール

汎用名
amAssetDst
ファイルの拡張子
txt

付属シナリオ - テキストコネクタ

このコネクタ用のシナリオはありません。

XMLコネクタ

オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

XMLコネクタでは、以下の操作が可能です。

- Connect-Itがインストールされているコンピュータやネットワーク上にあるXMLファイルを処理する。
- FTPサイトやWebサイトにあるXMLファイルを処理する。

 **注意:**

圧縮ファイル（gzip）も処理可能です。

XMLコネクタの既知の制約点

XMLコネクタに処理されるドキュメントタイプでは、以下のフィールドタイプはサポートされていません。

- Any
- PCDATA (Blob)
- CDATA
- Namespaces

XMLコネクタの設定（読み取り）

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 5.12. XMLコネクタの設定（読み取り）

コネクタに名前を付け、役割を説明する [p. 14]		
処理モードを選択する [p. 210]		
接続のプロトコルを選択する [p. 210]		
[ローカルファイル / ネットワークファイル] [p. 210]	HTTP Webサイトへの接続 [p. 212]	FTPサーバ接続 [p. 213]
処理後のアクションを定義する [p. 211]	処理後のアクションを定義する [p. 214]	
DTD / XSDを選択する [p. 215]		
ピボットドキュメントタイプの使用 [p. 36]		

警告:**ファイルまたはフォルダのパス**

コネクタの設定時に、コンピュータのネットワークドライブの文字に関連付けられたフォルダ内に位置する、ファイルまたはフォルダのパスを指定する場合は、パスにネットワークドライブの文字を入力せずに、ドライブの文字に関連付けられたフォルダを入力します。

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、『ユーザガイド』の「ドキュメントタイプのマッピング」の章、「マッピングの編集」/「ドキュメント要素のパス」を参照してください。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

処理モードを選択する

このページでは、コネクタが読み取りモードまたは書き込みモードのどちらで使用されるのかを指定できます。

接続のプロトコルを選択する

以下の3つのオプションがあります。

- HTTP Webサイト
- [FTPサーバ]
- [ローカルファイル/ネットワークファイル]

[ローカルファイル/ネットワークファイル]

ローカルファイルまたはネットワーク上のファイルを読み取る場合、以下の操作を実行します。

- 1 ファイルまたはフォルダを選択する
- 2 ファイル処理後のアクションを選択する

ファイルまたはフォルダを選択する

このページでは、コネクタが読み取るXMLファイルを選択します。以下の2つのオプションがあります。

- [ファイルを読み取る]
- [フォルダを読み取る]

警告:**ファイルまたはフォルダのパス**

コンピュータのネットワークドライブの文字に関連付けられたフォルダ内に位置する、ファイルまたはフォルダのパスを指定する場合は、パスにネットワークドライブの文字を入力せずに、ドライブの文字に関連付けられたフォルダを入力します。

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、『ユーザガイド』の「ドキュメントタイプのマッピング」の章、「マッピングの編集」/「ドキュメント要素のパス」を参照してください。

[ファイルを読み取る]

このオプションを選択する場合は、コンピュータやネットワーク上にあるXMLファイルのパスを指定します。

[フォルダを読み取る]

このオプションを選択する場合は、以下の2フィールドに入力します。

- **[フォルダ名]**
ファイルを含むフォルダのパスを入力します。
- **[拡張子]**
コネクタが読み取るファイルの拡張子を入力します。例：「xml」

[サブフォルダを読み取る]

このオプションを選択すると、コネクタは、選択したフォルダのサブフォルダ内のファイルも読み取ります。

処理後のアクションを定義する

XMLコネクタがファイルを読み取った後のアクション用に、3つのオプションがあります。

- **[フォルダに保存する]**
- **[フォルダから削除する]**
- **[次のフォルダに移す]**

3番目のオプションを選択する場合は、処理されたファイルの移動先フォルダのパスを入力します。

コネクタがXMLファイルの処理に成功または失敗した場合用に、上記の3オプションから1つを選択します。

HTTP Webサイトへの接続

WebサイトにあるXMLファイルの読み取りを選択する場合は、HTTP接続パラメータを入力します。

- [アドレス]
[プロトコル]://[アドレス]:[ポート]/[パス]のようにアドレスを入力します。
[アドレス]の部分のみが必須の場合もあります。
HTTPサーバ用の通常のポート番号は80です。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

セキュリティで保護された接続 (HTTPSプロトコル)

このオプションでは、セキュリティで保護された接続 (HTTPS) でサイトに接続するかどうかを指定できます。

重要項目:

Webサイトアドレスの [プロトコル] の部分がHTTPSプロトコルでない場合でも、[セキュリティで保護された接続 (HTTPSプロトコル)] オプションを使用すると、HTTPSプロトコルが強制的に使用されます。

HTTPSサーバ用の通常のポート番号は443です。

HTTPアドレスの例

[アドレス]	セキュリティで保護された接続 (HTTPSプロトコル)	接続モード
https://a.b.c.d/	選択	https port 443
https://a.b.c.d:9000/	選択	https port 9000
http://a.b.c.d:9000/	非選択	https port 9000
http://a.b.c.d:9000/	選択	https port 9000
https://a.b.c.d:9000/	非選択	https port 9000
http://a.b.c.d:443/	非選択	https port 443
https://a.b.c.d:80/	選択	https port 80

[クライアント証明書]

このフィールドでは、コンピュータに既存のHTTPS証明書の中から1つを選択します。

重要：コネクタがドキュメントを読み込んだ後またはドキュメントに書き込んだ後に、このフィールドに指定されているクライアント証明書を変更したり削除したりする場合は、シナリオビルダを一度終了してから再起動して、この変更事項を有効にします。

クライアント証明書のリストを管理する

設定ウィザードにあるクライアント証明書のリストは、Microsoft Managementアプリケーションの「コンソールルート / 証明書 - 現在のユーザー / 個人 / 証明書」フォルダにあるリストと一致します。

Windows 2000でこのフォルダに証明書を追加するには、

- 1 Windowsの【スタート】メニューで【ファイル名を指定して実行】を選択します。
- 2 【名前】フィールドに「MMC」と入力します。
- 3 【コンソール/スナップインの追加と削除】を選択します。
- 4 表示されるウィンドウで【追加】をクリックします。
- 5 表示されるウィンドウで【証明書】を選択します。
- 6 【追加】をクリックします。
- 7 表示されるダイアログボックスで【ユーザーアカウント】オプションを選択します。
- 8 【完了】をクリックします。
- 9 【OK】をクリックします。
- 10 「証明書 - 現在のユーザー / 個人 / 証明書」フォルダ内でファイルを追加または削除します。

FTPプロトコル

FTPサイトにあるXMLファイルの読み取りを選択する場合は、以下の操作を実行します。

- 1 FTP接続パラメータを選択する
- 2 ファイル処理後のアクションを選択する

FTPサーバ接続

このページでは、FTPサーバへの接続用に以下の3つのフィールドに入力します。

- 【サーバ】
FTPサーバ名を指定します。
- 【ログイン】
サイトへのアクセスを許可するログインを入力します。
- 【パスワード】
ログイン用のパスワードを入力します。

[**ファイルを読み取る**] または [**フォルダを読み取る**] オプションを選択します。 [**フォルダを読み取る**] を選択する場合以下の内容も指定します。

- FTPサイト上のフォルダのパス
- ファイルの拡張子
フィールドのデフォルト値は、「xml」です。

[**サブフォルダを読み取る**] オプションを選択することもできます。

処理後のアクションを定義する

XMLコネクタがファイルを読み取った後のアクション用に、3つのオプションがあります。

- [**フォルダに保存する**]
- [**フォルダから削除する**]
- [**次のフォルダに移す**]

3番目のオプションを選択する場合は、処理されたファイルの移動先フォルダのパスを入力します。

コネクタがXMLファイルの処理に成功または失敗した場合用に、上記の3オプションから1つを選択します。

シナリオ内の他のコネクタやマッピングボックスに取り込まれるドキュメントに、処理後のアクションが実行されるようにするには、各コネクタやマッピングボックスが生成する処理レポートを使用する必要があります。詳細については、『ユーザガイド』の「**処理レポート**」の章を参照してください。

[**ファイルを読み取る**]

このオプションを選択する場合は、FTPサイト上にあるXMLファイルのパスを指定します。

[**フォルダを読み取る**]

このオプションを選択する場合は、以下の2フィールドに入力します。

- [**フォルダ名**]
ファイルを含むフォルダのパスを入力します。
- [**拡張子**]
コネクタが読み取るファイルの拡張子を入力します。

[**サブフォルダを読み取る**]

このオプションを選択すると、コネクタは、選択したフォルダのサブフォルダ内のファイルも読み取ります。

DTD / XSDを選択する

XMLファイル进行处理するために、コネクタはDTD (Document Type Definition) またはXSD (Extended Schema Definition) を使用する必要があります。このページでは、使用するDTDまたはXSDを【DTD / XSD】フィールドで指定できます。

注意:

学習モード (下記参照) でXMLコネクタを使用する場合、【DTD】フィールドには空のファイル名を指定します。学習モードでデータ処理が終わった時点で、このファイルの内容が充填されます。このフィールドに記入された名前が既存ファイルの名前と一致すると、Connect-ItはこのDTDファイルに自動的に番号を付けて保存します。(例 : request_01.dtd) コネクタが数回学習モードで起動されると、起動の度に作成されるそれぞれのDTDには、作成の順番に従って番号が付けられます。(例 : request_01.dtd、request_02.dtd、request_03.dtd。「request_03.dtd」が最も最近に保存されたDTDファイルです。)

 をクリックすると、【ファイルの場所】ページが表示されます。このページでは、接続プロトコルの種類に応じてDTDまたはXSDのパス名を指定できます。

ファイルの場所

以下の3つのオプションがあります。

- 【ローカル/ネットワーク】

Connect-Itがインストールされているコンピュータやネットワーク上の、DTDまたはXSDの完全パス名を入力します。

- 【FTP】

【サーバ名】、【ログイン】、【パスワード】と【パス】フィールドにFTPパラメータを入力します。

【パス】フィールドでは、DTDまたはXSDのサーバ上の場所を指定します。例えば、DTDがサーバ「ftp.mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、【サーバ名】フィールドに「ftp.mycompany.com」を入力し、【パス】フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。

注意 : FTPサイトに学習モードでDTDを書き込むことはできません。

注意:

スラッシュ (/) はサーバ名の最後には入れませんが、DTDパスの最初に入れます。

- 【HTTP】

【サーバ名】、【ログイン】、【パスワード】と【パス】フィールドにHTTPパラメータを入力します。

【パス】フィールドでは、DTDまたはXSDのサーバ上の場所を指定します。例えば、DTDがサーバ「http://mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、【サーバ名】フィールドに「http://mycompany.com」を入力し、【パス】フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。

注意： HTTPサイトに学習モードでDTDを書き込むことはできません。

 **注意:**

スラッシュ（ / ）はサーバ名の最後には入れませんが、DTDパスの最初に入れます。

【第1レベルの各コレクションに使用可能なドキュメントタイプを発行する】

このオプションを選択すると、XMLコネクタは、選択したDTDまたはXSD内にある第1レベルのコレクションと同じ数の使用可能なドキュメントタイプを発行します。

例：

例

あるDTDのルート要素が「会社」で、第1レベルに3つのコレクション「従業員」、「サプライヤ」と「場所」があるとします。上記のオプションが選択されていないと、コネクタが発行する使用可能なドキュメントタイプのルート要素は「会社」になります。

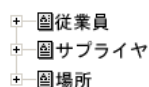
 **注意:**

DTDの第1レベルのコレクションは、コネクタの使用可能なドキュメントタイプのルート要素のすぐ下にあるコレクションになります。



【第1レベルの各コレクションに使用可能なドキュメントタイプを発行する】オプションが選択されていると、XMLコネクタは、DTDの第1レベルのコレクションごとに使用可能なドキュメントタイプを発行します。この場合DTDのルート

要素（要素「会社」）は、XMLコネクタの使用可能なドキュメントタイプには表示されません。



ルート要素を指定する

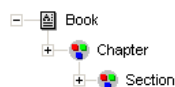
DTDまたはXSDは複数の要素から構成されており、これらの要素は更に別の要素を含んでいます。（例：文書編集用のDTDでは、複数の「Section」要素が複数の「Chapter」要素に含まれており、更に「Chapter」要素は1つの要素「Book」に含まれています。）他の要素に含まれることがない要素は、ルート要素と呼ばれます。DTDまたはXSDは、1つまたは複数のルート要素から構成されることもあれば、全くルート要素が存在しないこともあります。

ルート要素（つまり、XMLコネクタに発行される使用可能なドキュメントタイプに当たります）を指定するためには、以下の2つのオプションがあります。

[DTD / XSD内に見つかった各ルート要素に使用可能なドキュメントタイプを発行する(推奨オプション)]

このオプションを選択すると、XMLコネクタはDTDまたはXSDの全てのルート要素を検索し、見つかった各ルート要素ごとに使用可能なドキュメントタイプを発行します。

例：文書編集用のDTDで、ルート要素（「Book」）が1つだけ見つかったとします。この場合、XMLコネクタの使用可能なドキュメントタイプは1つのみです。



 **注意:**

ルート要素が見つからない場合（全ての要素が他の要素と包含関係にある場合）、Connect-Itは、DTDまたはXSD内で最初に検索された要素をルート要素と見なします。ルート要素が見つからないのは、例えば、DTDがリレーショナル（関係型）データベースの構造と同様に構成されている場合などです。リレーショナルデータベースでは、全要素（つまり全テーブル）がお互いを包含し合っているため、ルート要素はありません。例：資産のテーブルはユーザのテーブルに関連し、ユーザのテーブルはユーザの使用する資産のテーブルに関連しています。

[ユーザの選択する各ルート要素に使用可能なドキュメントタイプを発行する(コンマで値を区切る)]

このオプションを選択すると、ユーザはコンマで値を区切って、ルート要素を自分で指定することができます。

例：文書編集用のDTDで、ユーザが、3つのルート要素「Book」、「Chapter」と「Section」を選択するとします。この場合XMLコネクタは、ユーザの選んだルート要素ごとに使用可能なドキュメントタイプを発行します。

```

+ Book
+ Chapter
+ Section

```

 **注意:**

DTDがリレーショナルデータベースの構造（全要素つまり全テーブルがお互いを包含し合う関係）を取っている場合、このオプションを使用すると、テーブルごとに使用可能なドキュメントタイプを発行するように指定できます。

学習モード

学習モードを使用すると、XMLコネクタは [**ファイルまたはフォルダを選択する**] ページで指定したXMLファイルに基づいてDTDを作成します。

 **注意:**

学習モードで取得されたDTDは、ローカルまたはネットワークファイル内に保存されなければなりません。HTTPまたはFTPサイト上に保存することはできません。

学習モードを使用するには、


- **【学習モード】** オプションを選択し、XMLコネクタを設定します。
- シナリオ図部分でXMLコネクタを選択します。
- **【ツール / 生成する】** メニューを選択します。
または
- 右クリックして、ショートカットメニューから **【生成する】** を選択します。
または
- **【F5】** キーを押します。

学習モードを使用しない場合は、XMLコネクタを再度設定し、**【DTD / XSDを選択する】** ページでオプションをオフにします。

XMLドキュメントとDTD / XSDドキュメントの矛盾

原則として、XMLコネクタに読み取られるファイルのXML要素は、DTD内に定義された要素に一致するはずですが、新規の要素が現れることもあります。要素の少ないXMLファイルを学習モードで処理した時に得られたDTDを使用する際に、このようなXMLとDTDの矛盾が頻繁に起こります。

この矛盾を管理するために、Connect-Itには2つのオプションがあります。

- **ドキュメントを拒否する**
読み取られたXMLファイル内に、DTDまたはXSDで定義されていない要素があると、このファイルは拒否されます。(ドキュメントが保存されるかどうかは**【処理後のアクション】** ページのオプションによります。) ファイルを適切に処理するには、XMLコネクタを学習モードで再起動する必要があります。
- **警告を作成し、処理を続行する**
矛盾が発生しても、XMLファイルは正常に処理されます。 アイコンが、読み取りモードのXMLコネクタに関連するドキュメントログのトラッキング項目内に表示されます。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

XMLコネクタの設定 (書き込み)

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 5.13. XMLコネクタの設定（書き込み）

コネクタに名前を付け、役割を説明する [p. 14]	
処理モードを選択する [p. 220]	
接続のプロトコルを選択する [p. 220]	
ファイルまたはフォルダを選択する [p. 221]	HTTP Webサイトへの接続 [p. 221] FTPサーバ接続 [p. 222]
2セッション間の設定 [p. 222]	2セッション間の設定 [p. 222]
DTD / XSDを選択する [p. 223]	
ピボットドキュメントタイプの使用 [p. 36]	

警告:

ファイルまたはフォルダのパス

コネクタの設定時に、コンピュータのネットワークドライブの文字に関連付けられたフォルダ内に位置する、ファイルまたはフォルダのパスを指定する場合は、パスにネットワークドライブの文字を入力せずに、ドライブの文字に関連付けられたフォルダを入力します。

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、『ユーザガイド』の「ドキュメントタイプのマッピング」の章、「マッピングの編集」 / 「ドキュメント要素のパス」を参照してください。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

処理モードを選択する

このページでは、コネクタが読み取りモードまたは書き込みモードのどちらで使用されるのかを指定できます。

接続のプロトコルを選択する

以下のオプションがあります。

- [FTPサーバ]
- [ローカルファイル / ネットワークファイル]
- [HTTP Webサイト]

HTTP Webサイトへの接続

このページでは、ファイルの書き込み先HTTPサーバへ接続するために、以下の3つのフィールドに入力します。

- [アドレス]
これは必須フィールドです。[プロトコル]:/[アドレス]:[ポート]/[パス]のようにアドレスを入力します。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

セキュリティで保護された接続 (HTTPSプロトコル)

このオプションの使用方法については、本節の「セキュリティで保護された接続 (HTTPSプロトコル) [p. 212]」を参照してください。

書き込み用のコマンド

HTTPサーバへの書き込みには、以下の2タイプのコマンドが使用されます。

- POST
指定されたアドレスに位置するプログラムヘータを送信します。PUTメソッドと異なり、送信されたデータは処理されなければなりません。例：CGI書式のデータの送信
- PUT
データを送信し、指定されたアドレスにデータが保存されるようにします。例：Webサイトのページの更新

HTTPサーバ用のコマンドを選択します。

ローカルファイル/ネットワークファイル

ローカルまたはネットワーク上にファイルを書き込む場合は、ファイルまたはフォルダを選択します。

ファイルまたはフォルダを選択する

このページでは、コネクタが書き込むXMLファイルを選択します。以下の2つのオプションがあります。

- [1つのファイルにドキュメントを書き込む]
ローカルコンピュータまたはネットワーク上のXMLファイルのパス名を指定します。このファイルが既に存在すると、コネクタの起動時に既存ファイルの内容は上書きされます。

- [各ドキュメントごと別々のファイルに書き込む]
ファイルの書き込み先フォルダ名を指定します。ファイル名は、XMLコネクタの取り込み用ルール内に指定した汎用名に一致します（本節の「XMLコネクタ（書き込み）の取り込み用ルール-FTPとファイル [p. 227]」を参照してください）。

FTPプロトコル

FTPサイトにテキストファイルを書き込む場合は、以下の操作を実行します。

- 1 FTP接続パラメータを選択する
- 2 ファイル処理後のアクションを選択する

FTPサーバ接続

このページでは、FTPサーバへの接続用に以下の3つのフィールドに入力します。

- [サーバ]
FTPサーバ名を指定します。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

[1つのファイルにドキュメントを書き込む]

XMLコネクタで作成するファイルのパス名を指定します。このファイルが既に存在する場合は、既存ファイルの内容が上書きされます。

[各ドキュメントごと別々のファイルに書き込む]

ファイルの書き込み先フォルダ名を指定します。ファイル名は、XMLコネクタの取り込み用ルール内に指定した汎用名に一致します（本節の「XMLコネクタ（書き込み）の取り込み用ルール-FTPとファイル [p. 227]」を参照してください）。

2セッション間の設定

このページでは、データ書き込みの2セッション間のドキュメント処理を選択します。1つのファイルにデータを書き込むか、または1つのフォルダの複数のファイルにデータを書き込むかにより、動作オプションは変化します。

1つのファイルに書き込む

1つのファイルに書き込む場合は、以下のオプションがあります。

- [同じファイルの後に書き込む]
このオプションを選択すると、コネクタは、前回のセッションが中断した時に開いていたファイル内に、データの書き込みを再開します。
- [前回の全ファイルに上書きする]
コネクタは、前回のセッション時にデータを書き込んでいたファイルを削除します。
- [異なるファイルに番号を付ける]
コネクタはファイルを作成し、前回のファイル名を基に名前を付けます。
例：ファイル.xml、ファイル1.xml、ファイル2.xmlなど。

複数のファイルに書き込む

各ドキュメントごと別々のファイルに書き込む場合は、以下のオプションがあります。

- [前回の全ファイルを削除し、番号付けを再開する]
フォルダ内の全ファイルが削除されます（前回のセッションのみでなく、前回までの全セッションのファイルが削除されます）。
コネクタは、書き込む全ファイルへの番号付けを開始します。
- [ファイルの番号付けを続行する]
コネクタはファイルへの書き込みを再開し、既存ファイルを削除せずに番号を付けます。

DTD / XSDを選択する

XMLファイル进行处理するために、コネクタはDTD（Document Type Definition）またはXSD（Extended Schema Definition）を使用する必要があります。[DTD / XSDを選択する]のページでは、使用するDTDまたはXSDを[DTD / XSD]フィールドで指定できます。XMLコネクタの使用可能なドキュメントタイプは、このDTDまたはXSDに一致します。

をクリックすると、[ファイルの場所]ページが表示されます。このページでは、接続プロトコルの種類に応じてDTDまたはXSDのパス名を指定できます。

ファイルの場所

以下の3つのオプションがあります。

- [ローカル/ネットワーク]
Connect-Itがインストールされているコンピュータやネットワーク上の、DTDまたはXSDの完全パス名を入力します。
- [FTP]
[サーバ名]、[ログイン]、[パスワード]と[パス]フィールドにFTPパラメータを入力します。

[パス]フィールドでは、DTDまたはXSDのサーバ上の場所を指定します。例えば、DTDがサーバ「ftp.mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、[サーバ名]フィールドに「ftp.mycompany.com」を入力し、[パス]フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。

 **注意:**

スラッシュ (/) はサーバ名の最後には入れませんが、DTDまたはXSDパスの最初に入れます。

- [HTTP]
[サーバ名]、[ログイン]、[パスワード]と[パス]フィールドにHTTPパラメータを入力します。

[パス]フィールドでは、DTDまたはXSDのサーバ上の場所を指定します。例えば、DTDがサーバ「http://mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、[サーバ名]フィールドに「http://mycompany.com」を入力し、[パス]フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。

 **注意:**

スラッシュ (/) はサーバ名の最後には入れませんが、DTDまたはXSDパスの最初に入れます。

[XMLファイルで取得したDTDを参照しない]

このオプションはデフォルトのオプションです。

[取得したDTD / XSDをXMLファイル内に書き込む]

このオプションを選択すると、XMLコネクタが作成するXMLファイルは、ヘッダーにDTDまたはXSDを含みます。

[取得したDTD / XSDを外部ファイルに書き込み、XMLファイル内で参照する]

このオプションを選択する場合は、作成されるXMLドキュメントのDTDまたはXSDを保存する場所を、フィールドに入力する必要があります。DTDまたはXSDは、ローカルまたはネットワークファイル上に保存されなければなりません。

警告:

シナリオを起動すると複数のセッションが実行されます。各セッションは、スケジューラがシナリオのソースコネクタを起動する度に開始します（例：ソースコネクタは毎週ドキュメントを生成する、などのスケジュール）。

DTDまたはXSDは、シナリオの第1回目のセッション時にのみ外部ファイルに書き込まれます。つまりその後のセッションでも、第1回目のセッション時に取得され外部ファイルに保存されたDTDまたはXSDが、使用されることとなります。

新規のDTD / XSDが外部ファイルに書き込まれるようにするには、シナリオを停止し、再起動しなければなりません。

ルート要素を指定する

DTDまたはXSDは複数の要素から構成されており、これらの要素は更に別の要素を含んでいます。（例：文書編集用のDTDでは、複数の「Section」要素が複数の「Chapter」要素に含まれており、更に「Chapter」要素は1つの要素「Book」に含まれています。）他の要素に含まれることがない要素は、ルート要素と呼ばれます。DTDは、1つまたは複数のルート要素から構成されることもあれば、全くルート要素が存在しないこともあります。

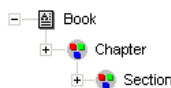
ルート要素（つまり、XMLコネクタに発行される使用可能なドキュメントタイプに当たります）を指定するためには、以下の2つのオプションがあります。

- [DTD / XSD内に見つかった各ルート要素に使用可能なドキュメントタイプを発行する(推奨オプション)]
- [ユーザの選択する各ルート要素に使用可能なドキュメントタイプを発行する(コンマで値を区切る)]

[DTD / XSD内に見つかった各ルート要素に使用可能なドキュメントタイプを発行する(推奨オプション)]

このオプションを選択すると、XMLコネクタはDTDまたはXSDの全てのルート要素を検索し、見つかった各ルート要素ごとに使用可能なドキュメントタイプを発行します。

例：文書編集用のDTDで、ルート要素（「Book」）が1つだけ見つかったとします。この場合、XMLコネクタの使用可能なドキュメントタイプは1つのみです。



 **注意:**

ルート要素が見つからない場合（全ての要素が他の要素と包含関係にある場合）、Connect-Itは、DTDまたはXSD内で最初に検索された要素をルート要素と見なします。ルート要素が見つからないのは、例えば、DTDがリレーショナル（関係型）データベースの構造と同様に構成されている場合などです。リレーショナルデータベースでは、全要素（つまり全テーブル）がお互いを包含し合っているため、ルート要素はありません。例：資産のテーブルはユーザのテーブルに関連し、ユーザのテーブルはユーザの使用する資産のテーブルに関連しています。

[ユーザの選択する各ルート要素に使用可能なドキュメントタイプを発行する(コンマで値を区切る)]

このオプションを選択すると、ユーザはコンマで値を区切って、ルート要素を自分で指定することができます。

例：文書編集用のDTDで、ユーザが、3つのルート要素「Book」、「Chapter」と「Section」を選択するとします。この場合XMLコネクタは、ユーザの選んだルート要素ごとに使用可能なドキュメントタイプを発行します。

```

+ Book
+ Chapter
+ Section

```

 **注意:**

DTDがリレーショナルデータベースの構造（全要素つまり全テーブルがお互いを包含し合う関係）を取っている場合、このオプションを使用すると、テーブルごとに使用可能なドキュメントタイプを発行するように指定できます。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

XMLコネクタ（書き込み）の取り込み用ルール - FTPとファイル

XMLコネクタの取り込み用ドキュメントタイプに適用されるルール（ディレクトリタイプ）を指定できます。

- [汎用名]
このフィールドのデフォルト値は、XMLコネクタの取り込み用ドキュメントタイプの名前です。
- [ファイルの拡張子]
拡張子はピリオドなしで入力します。例えば「xml」が正しい入力方法で、「.xml」は間違いです。デフォルトでは「xml」が入力されています。

XMLコネクタの設定時に、[各ドキュメントごと別々のファイルに書き込む]を選択した場合のみ、これらの2フィールドを使用します。

（本節の「XMLコネクタの設定（書き込み）」[p.219]を参照してください。）

シナリオの起動時に書き込まれるファイルの名前は、汎用名、ファイル作成の順番に一致する番号（_01、_02、_03など）と、指定されたファイルの拡張子から構成されます。

例：「ebizz」と「xml」を入力すると、作成されるファイルの名前は、ebizz_01.xml、ebizz_02.xml、ebizz_03.xmlなどとなります。

警告:

XMLコネクタに複数の取り込み用ドキュメントタイプがある場合、同じ汎用名を使用してはなりません。同じ汎用名が使用されていると、先に作成されたファイルが、後に作成されたファイルに削除され上書きされます。

図 5.6. XMLコネクタ - の取り込み用ルール

汎用名
amAssetDst
ファイルの拡張子
xml

追加情報 - XMLコネクタ

本節ではコネクタに関する追加情報を説明します。

1つまたは複数のファイルに書き込む

書き込みモードのXMLコネクタは、1つまたは複数のファイルにデータを書き込むことができます。

1つのファイルに書き込む

このモードでは、コネクタに取り込まれた各ドキュメントを1つのファイル内に記録します。

例えば、1つのXMLファイルのみが、AssetCenter内の資産の棚卸情報を含むようにする場合に、このモードを使用できます。以下のDTDを使用できます。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT AssetInventory (amAsset*)>
<!ELEMENT amAsset (AssetTag?)>
<!ELEMENT AssetTag (#PCDATA)>
```

このDTDではルート要素（AssetInventory）には子要素amAssetコレクションがあります。このコレクションの構成要素は、AssetCenterのデータベース内の資産のレコードに対応します。各資産は資産タグ（AssetTag）で識別されます。

この場合、XMLコネクタは以下のような内容を含むファイルを書き込みます。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<AssetInventory>
  <amAsset>
    <AssetTag>DEMO-NTR1</AssetTag>
  </amAsset>
  <amAsset>
    <AssetTag>DEMO-OFF1</AssetTag>
  </amAsset>
  <amAsset>
    <AssetTag>DEMO-SFT2</AssetTag>
  </amAsset>
</AssetInventory>
```



ソースとターゲットのマッピング

amAssetドキュメントタイプとXMLコネクタが使用するDTD間のマッピングには、以下のような特徴があります。

- 1 資産タグのフィールド (AssetTag) は、DTDのAssetTagフィールドに直接マップされます。
- 2 ルート要素 (AssetInventory) の子ノード (AssetTag) は、ターゲットドキュメントタイプ内に表示されます。
以下のDTDを使用します。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT AssetInventory (amAsset*, amSoftware)>
<!ELEMENT amAsset (AssetTag?)>
<!ELEMENT AssetTag (#PCDATA)>
<!ELEMENT amSoftware (Version?)>
<!ELEMENT Version (#PCDATA)>
<!ATTLIST AssetInventory ID CDATA #REQUIRED>
```

amSoftware要素とIDフィールドは、XMLコネクタの取り込み用ドキュメントタイプ内では無視され、表示されません。これはこの2つがDTD内で宣言されていないためです。

- 3 **[第1レベルの各コレクションに使用可能なドキュメントタイプを発行する]**
オプションを選択してはなりません。

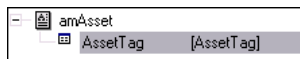
複数のファイルに書き込む

このモードでは、XMLコネクタが取り込む各ドキュメントを別々のファイル内に保存できます。

上記の例で説明すると、XMLコネクタは、資産の各レコードを別々のファイルに書き込むこととなります。

以下のDTDを使用できます。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT amAsset (AssetTag?)>
<!ELEMENT AssetTag (#PCDATA)>
```



この場合、XMLコネクタは複数のファイルに書き込みます。ファイルの内容は以下の例の通りです。

ファイル1

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<amAsset>
  <AssetTag>DEMO-OFF1</AssetTag>
</amAsset>
```

ファイル1

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<amAsset>
  <AssetTag>DEMO-NTR1</AssetTag>
</amAsset>
```

ソースとターゲットのマッピング

amAssetドキュメントタイプとXMLコネクタが使用するDTD間のマッピングには、以下のような特徴があります。

- 1 DTDで宣言されている全要素（コレクション、構造体、フィールド）がターゲット内に表示されます。
- 2 **[第1レベルの各コレクションに使用可能なドキュメントタイプを発行する]** オプションを選択してはなりません。

付属シナリオ - XMLコネクタ

このコネクタを使用する用例シナリオの一部は以下の通りです。

- [acscups /dataprop /01-SCcompany2ACamCompany.scn \[p. 322\]](#)
- [acscups /dataprop /02-SClocation2ACamLocation.scn \[p. 322\]](#)
- [acscups /dataprop /03-SCdept2ACamEmplDept.scn \[p. 322\]](#)
- [acscups /dataprop /04-SCcontacts2ACamEmplDept.scn \[p. 322\]](#)
- [acscups /dataprop /05-SCdevtype2ACamNature.scn \[p. 323\]](#)
- [acscups /dataprop /06-SCocmlcat2ACamNature.scn \[p. 323\]](#)
- [acscups /dataprop /07-SCmodel2ACamModel.scn \[p. 323\]](#)
- [acscups /dataprop /08-SCdevtype2ACamModel.scn \[p. 323\]](#)
- [acscups /dataprop /09-SCdevice2AACamPortfolioAndamComputer.scn \[p. 323\]](#)
- [acscups /dataprop /10-SClaptop2ACamPortfolioAndamComputer.scn \[p. 323\]](#)
- [acscups /dataprop /11-SCworkstation2ACamPortfolioAndamComputer.scn \[p. 324\]](#)
- [acscups /dataprop /12-SCpc2ACamAndamComputer.scn \[p. 324\]](#)
- [acscups /dataprop /13-SCserver2ACamAndamComputer.scn \[p. 324\]](#)
- [acscups /dataprop /14-SCphone2ACamPortfolioAndamPhone.scn \[p. 324\]](#)

6 | インベントリコネクタ

インベントリコネクタは、IT資産のインベントリを実行するアプリケーションのデータを処理するコネクタです。

Intel LANDeskコネクタ

オプションコネクタ

- 生成 (ソース)

Intel LANDeskインベントリコネクタでは、Intel LANDeskのインベントリツールで取得されたデータベースを処理できます。

2種類のIntel LANDeskコネクタがあります。

- Intel LANDesk 6.3と6.4コネクタ
- Intel LANDesk 6.5コネクタ

使用しているIntel LANDeskのバージョンに応じたコネクタを選択してください。このインベントリツールはIT資産の全情報を取得します。

Intel LANDeskコネクタの互換性

Intel LANDeskコネクタは、以下のドライバとODBC Administratorバージョン3.520.4403.2とのテストに成功しています。

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739
- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301
- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500

Intel LANDeskコネクタは、バージョン3.xxの全ODBCドライバと互換性があるものと推察されます。

既知の制約点

Intel LANDeskコネクタを適切に使用するためには、ODBCドライバとODBCアドミニストレータのDLLバージョンが、同一である必要があります。

例：





ODBCAdministratorバージョン3.0と、Access 4.00ODBCドライバを使用していると、Intel LANDeskコネクタは機能しません。






Intel LANDeskコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 6.1. Intel LANDeskコネクタの設定

	コネクタに名前を付け、役割を説明する [p. 14]
	接続タイプを選択する [p. 15]
	接続を選択する [p. 16]
	高度な設定 [p. 24]

-  [再接続のパラメータを設定する \[p. 27\]](#)
-  [サーバとの時間差を指定する \[p. 29\]](#)
-  [スケジュールのポイントを設定する \[p. 32\]](#)
-  [キャッシュを設定する \[p. 35\]](#)
-  [ピボットドキュメントタイプの使用 \[p. 36\]](#)

前提条件

コネクタの設定の前に、データベースへのODBC接続が、ODBCアドミニストレータ内で宣言されていなければなりません。

有効な宣言は、ODBC接続用の適切なログインとパスワードを使用しなければなりません。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続タイプを選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続タイプを選択する [p. 15]」の節を参照してください。

接続を選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続を選択する [p. 16]」の節を参照してください。

高度な設定

このページの使用方法については、「コネクタの設定 [p. 13]」章の「高度な設定 [p. 24]」の節を参照してください。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

Intel LANDeskコネクタの生成用ルール

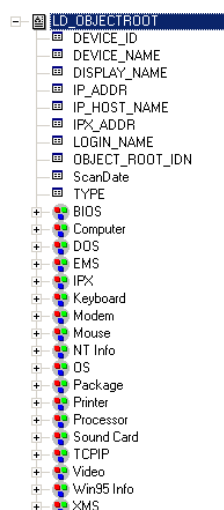
コネクタの生成用ルールの入力方法については、「コネクタのルール（ディレクティブ） [p. 39]」章の「生成用ルール [p. 40]」の節を参照してください。

このコネクタのルールとしてWHERE句とORDERBY句を作成すると、ソースデータベースのレコードにフィルタを適用できるようになります。これらの句の詳細については、「コネクタのルール（ディレクティブ） [p. 39]」章の「生成用ルール [p. 40]」節、「WHERE句とORDER BY句 [p. 41]」を参照してください。

使用可能なドキュメントタイプ

Intel LANDeskコネクタの使用可能なドキュメントタイプは、LD_OBJECTROOTのみです。このドキュメントタイプは、Intel LANDeskがコンピュータに実行するスキャンに相当します。

図 6.1. Intel LANDeskコネクタ - COMPUTER_SYSTEMドキュメントタイプ



付属シナリオ - PDA Intel LANDeskコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- `ild/ild64ac36/ildac.scn` [p. 304]
- `ild/ild65ac36/ildac.scn` [p. 304]

Tivoli Inventoryコネクタ (バージョン3.1と3.6)

オプションコネクタ

- 生成 (ソース)

Tivoli Inventory Managementコネクタでは、Tivoli Inventory アプリケーション (バージョン3.1と3.6) で取得されたデータベースを処理できます。このアプリケーションはTivoli Inventory Managementスイートに含まれています。

Tivoli InventoryはIT資産の全情報を取得します。Tivoli Inventory Management - Asset Management付属シナリオ (`tim2am.scn`) では、これらの情報を基に、Asset Managementアプリケーションの資産のテーブルにレコードを作成します。

Tivoli Inventory Managementコネクタの互換性

Tivoli Inventory Managementコネクタは、以下のドライバとODBC Administratorバージョン3.520.4403.2とのテストに成功しています。

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739
- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301
- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500

Tivoli Inventory Managementは、バージョン3.xxの全てのODBCドライバと互換性があるものと推察されます。

Tivoli Inventory Managementコネクタの既知の制約点

Tivoli Inventory Managementコネクタを適切に使用するためには、ODBCドライバとODBCアドミニストレータのDLLバージョンが、同一である必要があります。

例：

ODBCAdministratorバージョン3.0と、Access 4.00ODBCドライバを使用していると、Tivoli Inventory Managementコネクタは機能しません。

ポイントのステータスの必須の初期化

Tivoli Inventory Managementコネクタを含むシナリオでは、スケジュール編集用のウィンドウで、ポイントに値「0」を必ず割り当てなければなりません。スケジュールモードでシナリオを最初に起動する前に、Tivoli Inventory Managementがこのポイントに値を割り当てることは、デフォルトでは不可能です。

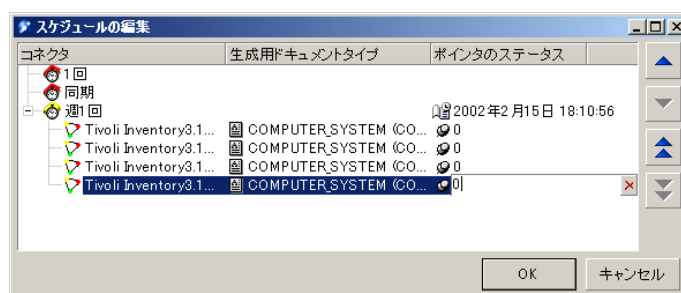
警告:

この操作を実行しないと、Tivoli Inventory Managementはドキュメントを生成できません。

この値を割り当てるには、

- 1 [シナリオ/スケジュール]を選択します。

- 2 Tivoli Inventory Managementコネクタに生成されるドキュメントを、予め作成したスケジューラに関連付けます。
- 3 生成されるドキュメントごとに、[ポインタのステータス]の列をダブルクリックします。
- 4 表示されるテキスト編集用枠に「0」を入力します。



Tivoli Inventory Managementコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 6.2. Tivoli Inventory Managementコネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]
接続タイプを選択する [p. 15]
接続を選択する [p. 16]
高度な設定 [p. 24]
再接続のパラメータを設定する [p. 27]
サーバとの時間差を指定する [p. 29]
スケジュールのポインタを設定する [p. 32]
キャッシュを設定する [p. 35]
ピボットドキュメントタイプの使用 [p. 36]

前提条件

コネクタの設定前に、データベースへのODBC接続が、ODBCアドミニストレータ内で宣言されていなければなりません。

有効な宣言は、ODBC接続用の適切なログインとパスワードを使用しなければなりません。

Connect-Itのコネクタの設定は、ウィザードで実行できます。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続タイプを選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続タイプを選択する [p. 15]」の節を参照してください。

接続を選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続を選択する [p. 16]」の節を参照してください。

高度な設定

このページの使用方法については、「コネクタの設定 [p. 13]」章の「高度な設定 [p. 24]」の節を参照してください。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

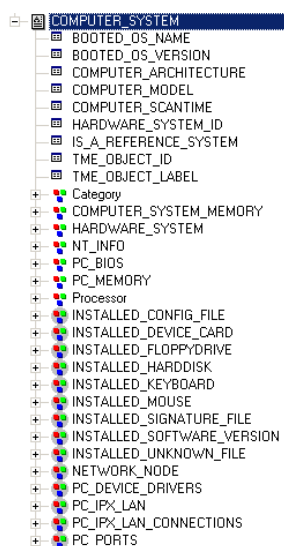
ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

使用可能なドキュメントタイプ

Tivoli Inventory Managementコネクタの使用可能なドキュメントタイプは、COMPUTER_SYSTEMのみです。このドキュメントタイプは、Tivoli Inventoryがコンピュータ上に実行するスキャンに相当します。このドキュメントタイプの要素は、スキャンの設定により変化します。例えばデータベースのテーブルを表すコレクションは、スキャンに応じて表示されたり消滅したりします。

図 6.2. Tivoli Inventory Managementコネクタ - COMPUTER_SYSTEMドキュメントタイプ



警告:

このドキュメントタイプの要素は、Tivoli Inventory Managementデータベースの構造に応じて変化します。このため、COMPUTER_SYSTEMドキュメントタイプでコレクションとして表されるテーブルが、データベースに存在しないこともあります。シナリオのマッピングが存在しないテーブルのフィールドを含んでいる場合、このシナリオは適切に実行されない可能性があります。この問題を解決するには、不在テーブルのフィールドを以下の場所で削除します。

- 1 シナリオのマッピング内
- 2 Tivoli Inventory Managementコネクタの生成用ドキュメントタイプ内

Tivoli Inventory Managementコネクタの生成用ルール

Tivoli Inventory Managementコネクタの生成用ルールにより、コネクタは、ODBCデータソース内のデータにフィルタと並べ替えを直接実行できます。

コネクタが生成するドキュメント用のデータにフィルタと並べ替えを実行するには、SQLクエリ用の句に類似する以下の2種類の句を作成する必要があります。

- WHERE句
- ORDER BY句

図 6.3. Tivoli Inventory Managementコネクタ - 生成用ルール

The screenshot shows a configuration window for a connector rule. At the top, there is a checkbox labeled "配列の取り出しを使用しない" (Do not use array output) which is unchecked. Below this, there are two input fields. The first is labeled "WHERE句" (WHERE clause) and contains the text "[INSTALLED_SOFTWARE_VERSION.CONFIG_CHANGE_TYPE] = 'INSERT'". The second is labeled "ORDER BY句" (ORDER BY clause) and is currently empty.

WHERE句

WHERE句ではODBCデータベースのレコードにフィルタを適用します。例えば、2001年1月23日かそれ以降に実行されたスキャンを取得するとします。この場合、[COMPUTER_SCANTIME]>='2001-01-23'と入力します。

ORDER BY句

ORDER BY句では、並べ替えの定義に使用するフィールドを指定します。これにより、Tivoli Inventory Managementコネクタはこの並べ替え順でドキュメントを生成します。複数のフィールドを指定する場合は、コンマで区切って入力します。

[配列の取り出しを使用しない]

Blobまたはメモタイプのフィールドなどを取得する際、テーブルに主フィールドがない場合は、このオプションを選択する必要があります。配列の取り出しを使用すると、Blobタイプのフィールドは適切に処理されず、切り取られます。このオプションは特に処理の性能を低下させます。

SQLシンタックス用の規則

ドキュメントの生成用ルールに使用するSQLシンタックスは、Tivoli Inventory データベース作成に使用したデータベースエンジンに使われているシンタックスと同一でなければなりません。

SQLシンタックスに加えて、Connect-Itには以下の規則があります。

- フィールド名は角括弧で囲む
例：[フィールド1]
- 文字列はシングルクォーテーションマークで囲む
例：'コンピュータ'
- 数値にはクォーテーションマークをつけない
- 日付は、コンピュータの地域の設定オプションで指定したフォーマット従って入力し、シングルクォーテーションマークで囲む

付属シナリオ - Tivoli Inventory Managementコネクタ (バージョン3.1と3.6)

このコネクタを使用する付属シナリオは以下の通りです。

- `tim/tim3ac35/timac.scn` [p. 326]

Tivoli Inventoryコネクタ (バージョン4.0)

オプションコネクタ

- 生成 (ソース)

Tivoli Inventory Management 4.0コネクタでは、Tivoli Inventory 4.0アプリケーションで取得されたデータベースを処理できます。このアプリケーションはTivoli Inventory Managementスイートに含まれています。

Tivoli InventoryはIT資産の全情報を取得します。Tivoli Inventory Management 4.0 - Asset Management付属シナリオ (tim4am.scn) では、これらの情報を基に、Asset Managementアプリケーションの資産のテーブルにレコードを作成します。

Tivoli Inventory Management 4.0コネクタの互換性

Tivoli Inventory Management 4.0コネクタは、以下のドライバとODBC Administratorバージョン3.520.4403.2とのテストに成功しています。

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739
- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301
- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500
- IBM DB2 ODBCバージョン7.01.00.00

Tivoli Inventory Management 4.0コネクタは、バージョン3.xxのODBCドライバ全てと互換性があるものと推察されます。

DB2

コネクタは、IBM DB2 ODBC 7.01.00.00ドライバともテストされており、正常に作動することが確認されています。

既知の制約点

Tivoli Inventory Management 4.0コネクタを適切に使用するためには、ODBCドライバとODBCアドミニストレータのDLLバージョンが、同一である必要があります。

例：

ODBC Administratorバージョン3.0と、Microsoft Access 4.00 ODBCドライバを使用していると、コネクタは機能しません。

Tivoli Inventory Management 4.0コネクタの設定

設定を行うと、コネクタはTivoli Inventory Management 4.0データソースに接続できるようになります。

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 6.3. Tivoli Inventory Management 4.0コネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]
接続タイプを選択する [p. 15]
接続を選択する [p. 16]
高度な設定 [p. 24]
再接続のパラメータを設定する [p. 27]
サーバとの時間差を指定する [p. 29]
スケジュールのポイントを設定する [p. 32]
キャッシュを設定する [p. 35]
ピボットドキュメントタイプの使用 [p. 36]

前提条件

コネクタの設定前に、データベースへのODBC接続が、ODBCアドミニストレータ内で宣言されていなければなりません。

有効な宣言は、ODBC接続用の適切なログインとパスワードを使用しなければなりません。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続タイプを選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続タイプを選択する [p. 15]」の節を参照してください。

接続を選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続を選択する [p. 16]」の節を参照してください。

高度な設定

このページの使用方法については、「コネクタの設定 [p. 13]」章の「高度な設定 [p. 24]」の節を参照してください。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

Tivoli Inventory Management 4.0 コネクタの生成用ルール

コネクタの生成用ルールの入力方法については、「コネクタのルール (ディレクティブ) [p. 39]」章の「生成用ルール [p. 40]」の節を参照してください。

このコネクタのルールとしてWHERE句とORDERBY句を作成すると、ソースデータベースのレコードにフィルタを適用できるようになります。これらの句の詳細については、「コネクタのルール (ディレクティブ) [p. 39]」章の「生成用ルール [p. 40]」節、「WHERE句とORDER BY句 [p. 41]」を参照してください。

Tivoli Inventory Management 4.0コネクタの取り込み用ルール

コネクタの取り込み用ルールの入力方法については、「コネクタのルール（ディレクティブ） [p. 39]」章の「取り込み用ルール [p. 43]」の節を参照してください。

取り込み用ルールを作成するには、**【整合性】**と**【高度な整合性チェック】**タブ内で整合性用のパラメータを入力します。

これらの整合性用タブの詳細については、「コネクタのルール（ディレクティブ） [p. 39]」章の「取り込み用ルール [p. 43]」節内「整合性 [p. 44]」を参照してください。

付属シナリオ - Tivoli Inventoryコネクタ（バージョン4.0）

このコネクタを使用する付属シナリオは以下の通りです。

- `tim/tim4ac36/timac.scn` [p. 326]
- `tim/tim4sc4/timsc.scn` [p. 327]

SMS 1.xコネクタとSMS 2.xコネクタ

オプションコネクタ

- 生成（ソース）

SMS 1.xとSMS 2.x（Microsoft Systems Management Service）ゲートウェイコネクタでは、SMSデータソースのデータを処理できます。

SMSコネクタでは、SMSデータベースのデータの読み取りのみが可能です。SMSデータベースへのデータの書き込みは不可能です。

SMSコネクタの互換性

SMS 1.xコネクタは、Microsoft Systems Management Serviceのバージョン1.x全てと互換性があります。

SMS 2.xコネクタは、Microsoft Systems Management Serviceのバージョン2.0、ビルド1239とテストされています。このコネクタはSystems Management Serviceのバージョン2.x全てと互換性があると推察されます。

SMSコネクタはMicrosoft SQL Serverドライバ、バージョン3.70.08.20とのテストの結果、正常に作動することが確認されています。このドライバの他のバージョンとも正常に機能するものと推察されます。

コネクタ使用上の重要な情報

ゲートウェイコネクタに関する注意事項

ゲートウェイコネクタを含むシナリオの稼働前に、データのインポート先データベースのコピー（例えばAssetCenterデータベースのコピー）を使ってこのシナリオを予めテストしておくことを強くお勧めします。

このテストでは以下の操作を実行できます。

- マッピングの識別キーの有効性を確認する
無効な識別キーを使用すると、ターゲットデータベースに重複する値が作成される可能性があります。重複する値を手動で削除する操作は非常に困難です。
- マップテーブル、文字列テーブル、グローバル関数を適応させる
Connect-It付属のマップテーブル、文字列テーブルやグローバル関数や、用例シナリオは、ユーザの実情に合わない可能性があります。
例：Brandマップテーブルに項目を追加します。製品のメーカーを含むこのマップテーブル（「genmaps.mpt」ファイル内）には、ユーザが必要なメーカーが含まれていない可能性があります。

SMSコネクタに関する注意事項

全SMSコネクタはSMSデータベースと共にテストされています。以下の場合、付属シナリオが適切に機能しない、または付属シナリオを開くことができない可能性があります。

- SMSコネクタの接続先データベースがカスタマイズされている場合（テーブルの追加や削除、またはフィールド名の変更など）。この場合シナリオでマップされている要素が、コネクタの生成用ドキュメントタイプに含まれていない可能性があります。
- SMSコネクタの接続先データベースがSMSデータベースではない場合。シナリオでマップされている要素は、コネクタの生成用ドキュメントタイプには含まれません。

使用するSMS 2.xデータベースの構造がSMS 1.xデータベースの構造と一致する場合

Microsoftのツールを用いてSMS 1.xデータベースからSMS 2.xデータベースへ移行した場合、データベースの内容はSMS 1.xデータベースと同じです。この場合、以下のことに注意する必要があります。

- SMS 2.xコネクタの使用可能なドキュメントタイプはデータベース構造には一致しません。
- SMS 2.x - Asset Management付属シナリオのマッピングは使用不可能になります。旧SMS 1.xデータベースから継承されたテーブルを考慮しつつ、データベースをカスタマイズする必要があります。

SMSコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 6.4. SMS 1.xコネクタとSMS 2.xコネクタの設定

☐	コネクタに名前を付け、役割を説明する [p. 14]
☐	接続タイプを選択する [p. 15]
☐	接続を選択する [p. 16]
☐	高度な設定 [p. 24]
☐	再接続のパラメータを設定する [p. 27]
☐	サーバとの時間差を指定する [p. 29]
☐	スケジュールのポイントを設定する [p. 32]
☐	キャッシュを設定する [p. 35]
☐	ピボットドキュメントタイプの使用 [p. 36]

前提条件

コネクタの設定の前に、SMSデータベースへの有効な接続が、ODBCアドミニストレータ内で宣言されていなければなりません。

有効な宣言は、ODBC接続用の適切なログインとパスワードを使用しなければなりません。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続タイプを選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続タイプを選択する [p. 15]」の節を参照してください。

接続を選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続を選択する [p. 16]」の節を参照してください。

高度な設定

このページの使用方法については、「コネクタの設定 [p. 13]」章の「高度な設定 [p. 24]」の節を参照してください。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

SMSコネクタの生成用ルール

コネクタの生成用ルールの入力方法については、「コネクタのルール(ディレクティブ) [p. 39]」章の「生成用ルール [p. 40]」の節を参照してください。

このコネクタのルールとしてWHERE句とORDERBY句を作成すると、ソースデータベースのレコードにフィルタを適用できるようになります。これらの句の詳細については、「コネクタのルール(ディレクティブ) [p. 39]」章の「生成用ルール [p. 40]」節、「WHERE句とORDER BY句 [p. 41]」を参照してください。

SMSコネクタ用のWHERE句の例

ID (dwMachineID) が4575であるコンピュータに当たるレコードを取得するには、WHERE句のフィールドに以下の関数を入力します。

```
[dwMachineID]=4575
```

図 6.4. SMSコネクタ - 生成用ルールの例



付属シナリオ - SMS 1.xとSMS 2.xコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- sms \sms1ac35 \smsac.scn [p. 324]
- sms \sms1ac35 \smsact.scn [p. 325]
- sms \sms1ac35 \smsac.scn [p. 325]
- sms \sms2ac35 \smsact.scn [p. 325]

Unicenter AMOコネクタ

オプションコネクタ

- 生成 (ソース)

Unicenter AMOゲートウェイコネクタでは、Unicenter AMOスイートのインベントリツールで取得されたデータベースを処理できます。

このインベントリツールはIT資産の全情報を取得します。Unicenter AMO - Asset Management付属シナリオ (amo2am.scn) では、これらの情報を基に、Asset Managementアプリケーションの資産のテーブルにレコードを作成できます。

Unicenter AMOコネクタの互換性

Unicenter AMOコネクタは、以下のドライバとODBC Administratorバージョン3.520.4403.2とのテストに成功しています。

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739
- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301
- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500

Unicenter AMOコネクタは、バージョン3.xxのODBCドライバ全てと互換性があるものと推察されます。

Unicenter AMOコネクタの既知の制約点

Unicenter AMOコネクタを適切に使用するためには、ODBCドライバとODBCアドミニストレータのDLLバージョンが、同一である必要があります。

例：

ODBCAdministratorバージョン3.0と、MicrosoftAccess 4.00ODBCドライバを使用していると、Unicenter AMOコネクタは機能しません。

Unicenter AMOコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 6.5. Unicenter AMOコネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]
接続タイプを選択する [p. 15]
接続を選択する [p. 16]
高度な設定 [p. 24]
再接続のパラメータを設定する [p. 27]
サーバとの時間差を指定する [p. 29]
スケジュールのポイントを設定する [p. 32]
キャッシュを設定する [p. 35]
ピボットドキュメントタイプの使用 [p. 36]

前提条件

コネクタの設定の前に、データベースへのODBC接続が、ODBCアドミニストレータ内で宣言されていなければなりません。

有効な宣言は、ODBC接続用の適切なログインとパスワードを使用しなければなりません。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続タイプを選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続タイプを選択する [p. 15]」の節を参照してください。

接続を選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続を選択する [p. 16]」の節を参照してください。

高度な設定

このページの使用方法については、「コネクタの設定 [p. 13]」章の「高度な設定 [p. 24]」の節を参照してください。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

Unicenter AMOコネクタの生成用ルール

コネクタの生成用ルールの入力方法については、「コネクタのルール（ディレクトティブ） [p. 39]」章の「生成用ルール [p. 40]」の節を参照してください。

このコネクタのルールとしてWHERE句とORDERBY句を作成すると、ソースデータベースのレコードにフィルタを適用できるようになります。これらの句の詳細については、「コネクタのルール(ディレクティブ) [p. 39]」章の「生成用ルール [p. 40]」節、「WHERE句とORDER BY句 [p. 41]」を参照してください。

使用可能なドキュメントタイプ

Unicenter AMOコネクタには以下の使用可能なドキュメントタイプがあります。

- Application
- Computer
- Engine
- Groupe
- User
- DirectAccessTables

このドキュメントタイプは、コネクタの他の使用可能なドキュメントタイプ全てのコレクションを含みます。このコレクションを、ターゲットコネクタの取り込み用ドキュメントタイプに直接マップすることも可能です。

コレクション - ドキュメントのマッピングについては、『ユーザガイド』の「ドキュメントタイプのマッピング」の章の「マッピングの種類」の節、「コレクションからドキュメントへのマッピング」を参照してください。

付属シナリオ - Unicenter AMOコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- amo /amoac36 /amoac.scn [p. 302]

Tivoli CM Inventory 4.2 コネクタ

オプションコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

Tivoli CM Inventory 4.2コネクタでは、Tivoli Configuration Management バージョン4.2アプリケーションの、インベントリ機能を使って取得したデータを含むデータベースを処理できます。

このインベントリ機能は、IT資産に関する全情報を取得します。
「tim/tcm42ac41/tcmac.scn」付属シナリオは、この情報をAssetCenter 4.1データベースの [コンピュータ] のテーブル (amComputer) へ挿入します。

Tivoli CM - Inventory 4.2コネクタの互換性

Tivoli CM - Inventory 4.2コネクタは、以下のドライバとODBC Administratorバージョン3.520.4403.2とのテストに成功しています。

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739
- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301
- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500

Tivoli CM - Inventory 4.2コネクタは、バージョン3.xxのODBCドライバ全てと互換性があるものと推察されます。

Tivoli CM - Inventory 4.2コネクタの既知の制約点

Tivoli CM - Inventory 4.2コネクタを適切に使用するためには、ODBCドライバとODBCアドミニストレータのDLLバージョンが、同一である必要があります。

例：

ODBCAdministratorバージョン3.0と、Access 4.00ODBCドライバを使用していると、Tivoli CM Inventory 4.2コネクタは機能しません。

ポインタのステータスの必須の初期化

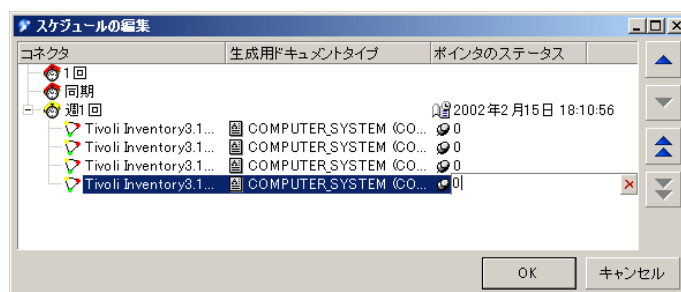
Tivoli CM Inventory 4.2コネクタを含むシナリオでは、スケジュール編集用のウィンドウで、ポインタに値「0」を必ず割り当てなければなりません。スケジュールモードでシナリオを最初に起動する前に、Tivoli CM Inventory 4.2がこのポインタに値を割り当てることは、デフォルトでは不可能です。

警告:

この操作を実行しないと、Tivoli CM Inventory 4.2コネクタはドキュメントを生成できません。

この値を割り当てるには、

- 1 **[シナリオ/スケジュール]** を選択します。
- 2 Tivoli CM Inventory 4.2コネクタに生成されるドキュメントを、予め作成したスケジューラに関連付けます。
- 3 生成されるドキュメントごとに、**[ポインタのステータス]** の列をダブルクリックします。
- 4 表示されるテキスト編集用枠に「0」を入力します。



Tivoli CM Inventory 4.2コネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 6.6. Tivoli Inventory Management 4.0コネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]
接続タイプを選択する [p. 15]
接続を選択する [p. 16]
高度な設定 [p. 24]
再接続のパラメータを設定する [p. 27]
サーバとの時間差を指定する [p. 29]
スケジュールのポインタを設定する [p. 32]
キャッシュを設定する [p. 35]

前提条件

コネクタの設定前に、データベースへのODBC接続が、ODBCアドミニストレータ内で宣言されていなければなりません。

有効な宣言は、ODBC接続用の適切なログインとパスワードを使用しなければなりません。

Connect-Itのコネクタの設定は、ウィザードで実行できます。

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

接続タイプを選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続タイプを選択する [p. 15]」の節を参照してください。

接続を選択する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「接続を選択する [p. 16]」の節を参照してください。

高度な設定

このページの使用方法については、「コネクタの設定 [p. 13]」章の「高度な設定 [p. 24]」の節を参照してください。

再接続のパラメータを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

スケジュールのポイントを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「スケジュールのポイントを設定する [p. 32]」の節を参照してください。

キャッシュを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「キャッシュを設定する [p. 35]」の節を参照してください。

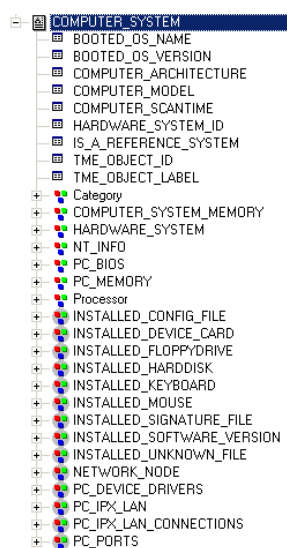
ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

使用可能なドキュメントタイプ

Tivoli CM Inventory 4.2コネクタの使用可能なドキュメントタイプは、COMPUTER_SYSTEMのみです。このドキュメントタイプは、Tivoli Inventoryがコンピュータ上に実行するスキャンに相当します。このドキュメントタイプの要素は、スキャンの設定により変化します。例えばデータベースのテーブルを表すコレクションは、スキャンに応じて表示されたり消滅したりします。

図 6.5. Tivoli CM - Inventory 4.2コネクタ - COMPUTER_SYSTEMドキュメントタイプ



警告:

このドキュメントタイプの要素は、Tivoli Configuration Management 4.2データベースの構造に応じて変化します。このため、COMPUTER_SYSTEMドキュメントタイプでコレクションとして表されるテーブルが、データベースに存在しないこともあります。シナリオのマッピングが存在しないテーブルのフィールドを含んでいる場合、このシナリオは適切に実行されない可能性があります。この問題を解決するには、不在テーブルのフィールドを以下の場所で削除します。

- 1 シナリオのマッピング内
- 2 Tivoli CM Inventory 4.2コネクタの生成用ドキュメントタイプ内

Tivoli CM Inventory 4.2コネクタの生成用ルール

Tivoli CM Inventory 4.2コネクタの生成用ルールにより、コネクタは、ODBCデータソース内のデータにフィルタと並べ替えを直接実行できます。

コネクタが生成するドキュメント用のデータにフィルタと並べ替えを実行するには、SQLクエリ用の句に類似する以下の2種類の句を作成する必要があります。

- WHERE句
- ORDER BY句

図 6.6. Tivoli CM Inventory 4.2コネクタ - 生成用ルール

The screenshot shows a configuration window for a connector rule. At the top, there is a checkbox labeled "配列の取り出しを使用しない" (Do not use array retrieval), which is currently unchecked. Below this, there are two input fields. The first is labeled "WHERE句" (WHERE clause) and contains the SQL snippet: "[INSTALLED_SOFTWARE_VERSION.CONFIG_CHANGE_TYPE] = 'INSERT'". The second is labeled "ORDER BY句" (ORDER BY clause) and is currently empty.

WHERE句

WHERE句ではODBCデータベースのレコードにフィルタを適用します。例えば、2001年1月23日かそれ以降に実行されたスキャンを取得するとします。この場合、[COMPUTER_SCANTIME]>='2001-01-23'と入力します。

ORDER BY句

ORDER BY句では、並べ替えの定義に使用するフィールドを指定します。これにより、Tivoli CM Inventory 4.2コネクタはこの並べ替え順でドキュメントを生成します。複数のフィールドを指定する場合は、コンマで区切って入力します。

[配列の取り出しを使用しない]

Blobまたはメモタイプのフィールドなどを取得する際、テーブルに主フィールドがない場合は、このオプションを選択する必要があります。配列の取り出しを使用すると、Blobタイプのフィールドは適切に処理されず、切り取られます。このオプションは特に処理の性能を低下させます。

付属シナリオ - Tivoli CM Inventory 4.2コネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- `tim/tcm42ac41/tcmac.scn` [p. 327]

7 | Javaコネクタ

Javaコネクタにより、Java言語を使用するアプリケーションと通信できるようになります。

Connect-ItにはJava開発キット（JDK）が付属しており、独自のJavaコネクタをユーザが作成できるようになっています。

この開発キットの詳細については『**ユーザガイド**』の「Connect-ItのJava開発キット（JDK）」を参照してください。

SCAutoリスニングコネクタ

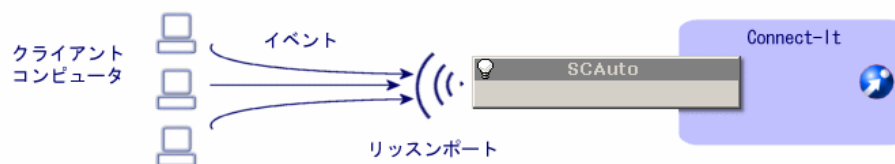
オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

SCAuto リスニングコネクタは、ServiceCenterのSCAutomateサーバのエミュレータ（SCAuto）として機能します。コネクタは、イベントを受信しクライアントの要求を処理することが出来ます。

イベント受信モードのSCAutoリスニングコネクタ

イベント受信モードのSCAutoリスニングコネクタはイベントを受信します。このイベントによりコネクタは、手動またはスケジュールモードでコネクタが起動する度にドキュメントを生成できるようになります。このコネクタは通信路（ソケット）を使用します。



クライアント要求の処理モードのSCAutoリスニングコネクタ

クライアント要求の処理モードでのコネクタの使用は、Connect-Itサーバに送信する要求を処理するリソースコンポーネントの使用に依存しています。



SCAutoリスニングコネクタの制約点

コネクタは以下のアダプタをサポートしません。

- 日本語版のSCAuto OpenView Network Node Manager adapterバージョン3.x
- SCAuto Email adapter
- 以下のコマンドを使用する全アダプタ
 - QUERY2
 - INSERTBLOB
 - CREATEBLOB
 - DESCRIBE OBJECT

- SELECT OBJECTS
- END SELECT OBJECTS
- GETNEXT OBJECT
- STORE OBJECT




SCAutoリスニングコネクタがSCAutoのSDKの「DELETE」コマンドをサポートする場合でも、このコマンドはイベントを削除することは出来ません。イベントを削除するには、処理レポートを使用する必要があります。処理レポートの詳細については、『ユーザガイド』の「**処理レポート**」の章を参照してください。

SCAutoリスニングコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 7.1. SCAutoリスニングコネクタの設定

	コネクタに名前を付け、役割を説明する [p. 14]
	コネクタの導入 [p. 263]
	ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

コネクタの導入

このページでは、SCAutoサーバへのコネクタ接続パラメータを設定できます。

リッスンポート

クライアントの要求を受信するためにコネクタが使用するリッスンポートを指定します。

これは必須フィールドです。

設定ファイル

SCAutoリスニングコネクタの使用可能なドキュメントタイプを記述する、XML設定ファイルのパスを指定します。


これは必須フィールドです。

これらのドキュメントタイプはSCAutoのイベントのタイプに対応します。Connect-Itのシナリオでは、ServiceCenterのイベントタイプから設定ファイルを取得することが可能です。

設定ファイルを取得するには、

- 1 シナリオビルダを起動します。
- 2 「scacfg.scn」シナリオを開きます。

このファイルは、[Connect-Itインストール先フォルダ]scenario\scauto\scacfgにあります。

- 3 シナリオのServiceCenterコネクタを設定します。
- 4  をクリックしてシナリオのコネクタを開きます。
- 5 全スケジューラを起動します（「Ctrl + F5」キー）。

シナリオは、[Connect-Itインストール先フォルダ]scenario\scauto\scacfgにXMLファイルを作成します。

要求用のリソース

Connect-Itサーバに送信する要求を処理するコンポーネントの名前を入力します。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

追加情報

SCAutoリスニングコネクタ設定ファイルを変更する

SCAutoリスニングコネクタのXML設定ファイルを変更すると、サポートできるイベントのタイプを増減できます。また、文字列のコレクションをMemoタイプのフィールドとして処理することも可能です。

設定ファイルには2つのセクションがあります。

- InputEventTypes

SCAutoリスニングコネクタが取り込むイベント

取り込まれるイベントは、SCAutoアダプタが生成するイベントに相当します。

- OutputEventTypes

SCAutoリスニングコネクタが生成するイベント

生成されるイベントは、SCAutoアダプタが要求するイベントに相当します。

イベントタイプは構造体として表現され、構造体の名前はイベントタイプの名前になります。この構造体は、イベントのフィールドを表すサブノードを含みます。

4つのタイプの単一フィールドがサポートされています。

- String (Character)
- Byte (Logical)
- Long (Number)
- TimeStamp (Date)

括弧内のタイプは、ServiceCenterのタイプに相当します。配列 (Array) タイプのフィールドは、単一フィールドのコレクションの形で表されます。これは scacfg.scnシナリオでのデフォルトの表示方法です。Memoタイプの単一フィールドとして表示することも可能ですが、この場合フィールド値は1段落 (パラグラフ) となり、配列内の要素の各値は1つの行で表されます。**注意**：文字列の配列のみにこの表示方法を適用すべきです。この表示方法を文字列以外の配列に適用すると、配列の要素は文字列であると見なされてしまいます。

単一フィールドはインデックス、名前と、タイプで特徴付けられます。

例

```
ATTRIBUTE index="11" name="orig.operator" type="String"/
```

配列フィールドは、インデックス、名前、区切り文字、要素名と、要素タイプで特徴付けられます。要素名は恣意的に付けられるため、フィールド名とは異なる場合があります。

例

```
< COLLECTION index="2" name="comments.2" separator="|">
  < ATTRIBUTE name="comments" type="String"/>
</COLLECTION>
```

この例では、配列フィールドを以下の行で表現することも可能です。

```
< ATTRIBUTE index="2" name="comments.2" type="Memo" separator="|" />
```

以下の例では3つの要素 (「abc」 「def」 と 「ghi」) の配列は、属性値を付与します。

```
"abc
def
ghi"
```

属性「separator」がない場合または空の場合は、デフォルトの区切り文字「|」が使用されます。

「lf」と「cr」の2つは特別な区切り文字です。この2つは改行（line feed）と復帰（carriage return）を意味します。この2つの区切り文字以外の場合、区切り文字は^以外の1文字でなければなりません。

付属シナリオ - SCAutoリスニングコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- scauto /scacfg /scacfg.scn [p. 321]
- scauto /sca-sc /sca-sc.scn

このシナリオでは、同じフォルダ内にある設定ファイル「BasicScaCfg.xml」を、シナリオの設定中に使用する必要があります。

XMLリスニングコネクタ

オプションコネクタ

- 生成（ソース）

XMLリスニングコネクタは、要求をコネクタに転送するサーブレットから以下の操作を実行できます。

- コネクタが受信する要求からドキュメントを生成します。
- 受信する要求を処理する別のコネクタと相互作用します。

XMLデータは、サーブレットへ送信されるHTTP要求の内容から抽出されます。

発行モードでのXMLリスニングコネクタ（ドキュメントの生成）

使用可能なドキュメントタイプ発行モードのXMLリスニングコネクタは、イベントを受信します。このイベントによりドキュメントの生成が可能になります。



相互作用モードのXMLリスニングコネクタ

相互作用モードで、コネクタはクライアントが送信するHTTP要求をリスンします。コネクタはこの要求をリソースコネクタへ転送し、このリソースコネクタがドキュメントを生成します。



XMLリスニングコネクタの互換性

コネクタは、Javaサーブレットをサポートする全Webサーバと互換性があります。

XMLリスニングコネクタの制約点




XMLリスニングコネクタの制約点は、使用するサーブレットにより変化します。

XMLリスニングコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 7.2. XMLリスニングコネクタの設定

	コネクタに名前を付け、役割を説明する [p. 14]
	コネクタの導入 [p. 268]
	ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

コネクタの導入

接続

この枠では、XMLリスニングコネクタへの接続パラメータを設定できます。

- **ポート**
XMLリスニングコネクタが使用するリッスンポートを入力します。
このポートを使用すると、Webサーバは、WebサーバとXMLリスニングコネクタ間の通信路（ソケット）を作成できます。
- **URL**
XMLイベントを生成するHTTP要求（POSTまたはGET）を処理するサブレットのURLを指定します。
- **同時接続の最大数**
コネクタが処理する同時接続の最大数を指定します。フィールドのデフォルト値は20です。
[この最大数を超過する接続を拒否する] オプションはデフォルトでオフになっています。この場合、接続が最大数に達すると、新規接続はコネクタに処理されるまでキュー内に保持されます。

要求の処理

この枠では要求の処理用のパラメータを入力できます。

- HTTP要求の処理ポリシー

コネクタがHTTP要求を処理する際のモードを指定します。[発行]と[相互作用]の選択肢があります。

- 発行用の設定ファイル
コネクタが発行する使用可能なドキュメントタイプを指定するXMLファイルのパスを指定します。
- リソース
相互作用の要求を処理するコネクタを指定します。このコネクタはドキュメントの生成モードで使用できなければなりません。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

XMLリスニングコネクタの生成用ルール

これらのルールは、コネクタがXMLドキュメントの形で処理するHTTP要求内にあります。

XMLリスニングコネクタの使用可能なドキュメントタイプ

発行モードのXMLリスニングコネクタは、設定ファイルで指定されている使用可能なドキュメントタイプを発行します。

付属シナリオ - XMLリスニングコネクタ

このコネクタ用のシナリオはありません。

Tivoli Enterprise Consoleコネクタ（送信）

オプションコネクタ

- 取り込み（ターゲット）

送信モードのTivoli Enterprise Consoleコネクタでは、Tivoli Enterprise Console（TEC）イベントを送信できます。

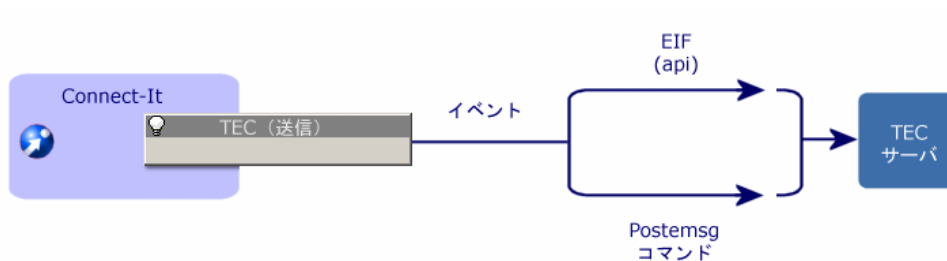
注意:

TECコネクタ（送信）はTECアダプタに相当します。

このコネクタはJCA規格に従って開発されています。

TECコネクタはソースコネクタからくるドキュメントを取り込み、以下のいずれかを使用してドキュメントを送信します。

- TECのEIF API
- postmsgコマンド



前提条件

TECコネクタ（送信）を使用するには、Connect-Itのクラスパス内に「eif.jar」ライブラリを追加する必要があります。

TEC 3.7.1の「eif.jar」ライブラリは、フォルダ[TECのインストール先フォルダ]/bin/generic-unix/TME/TEC内にあります。それより前のバージョンのTECを使用している場合は、Tivoliの顧客サポートまでお問合せください。

TEC adapter（送信）コネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 7.3. TEC adapterコネクタの設定

_____	コネクタに名前を付け、役割を説明する [p. 14]
_____	コネクタの導入 [p. 271]

📖 ピポットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

コネクタの導入

このページでは、TECサーバへの接続パラメータを設定できます。

サーバアドレス

TECサーバのアドレスを入力します。
これは必須フィールドです。

サーバポート

サーバポートを入力します。
これは必須フィールドです。

設定ファイル

TECイベントクラスを含むXMLファイルを指定します。このファイルのイベントクラスは、BAROCファイルで定義されているTECサーバのクラスに一致しなければなりません。

これは必須フィールドです。

TECデータベースのイベントクラスを含むデフォルトの記述ファイルが、Connect-Itに付属しています。ファイルは、[Connect-Itインストール先フォルダ]/config/tec/teccfg.xmlにあります。

イベントバッファを有効にする

このオプションを選択すると、TECサーバへの接続が切断された場合、サーバに送信されるイベントはバッファファイルに格納されます。

🔥 注意:

このオプションを使用しないと、TECサーバへの接続が切断した場合コネクタはドキュメントを拒否します。

バッファファイル

バッファファイルの完全パスを指定します。

[イベントバッファを有効にする] オプションが選択されている場合、このフィールドは必須です。

バッファファイルの最大サイズ (KB)

バッファファイルの最大サイズを指定します。ファイルの最大サイズに達し、サーバへの接続が切断されると、TECコネクタは新規ドキュメントの取り込みを拒否します。

バッファ削除の速度 (イベント/分)

サーバへの接続が回復した際に、バッファ内に格納されるドキュメントをConnect-Itが処理する速度を指定できます。1分ごとのイベントの数はTECサーバの性能に応じて調節されます。

値「0」は、速度制限なしにイベントが削除されることを意味します。

イベントの最大サイズ

このフィールドでは、TECコネクタが送信するイベントの最大サイズを指定します。

サイズは256KBと4096KBの間でなければなりません。

サーバはUTF-8をサポートしない

デフォルトでは、TECサーバへ送信される全イベントは、UTF-8にコード化されます。

サーバがUTF-8をサポートしないは、このオプションを選択します (TECバージョン3.7より前)。

このオプションを選択する場合は、[サーバのコードページ] フィールドにサーバが使用するコードページを入力します。

サーバのコードページ

上記のオプションを選択する場合、サーバが使用するコードページをこのフィールドに入力します。

コマンドラインを使用する

postemsgコマンドを使ってコネクタがTECサーバへイベントを送信する場合、このオプションを選択します。

'postemsg' コマンドのパス

postemsgコマンドの完全パスを指定します。

[コマンドラインを使用する] オプションが選択されている場合、このフィールドは必須です。

このコマンドのパスは、[TECインストール先フォルダ]/bin/[コンピュータの依存フォルダ]/bin/です。

例：Windows 32ビットのTECサーバでは、コンピュータの依存フォルダは「W32ix86」です。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

追加情報

TECコネクタ（送信）にデフォルトで付属しているイベントクラスのファイルを、新規のイベントクラスを使用するために拡張することも可能です。イベントクラスはBAROCファイルで定義されています。本章の「Tivoli Enterprise Console コネクタ（受信） [p. 273]」章の「SCA Autoリスニングコネクタ設定ファイルを変更する [p. 264]」の節を参照してください。

付属シナリオ

シナリオは提供されていません。

Tivoli Enterprise Console コネクタ（受信）

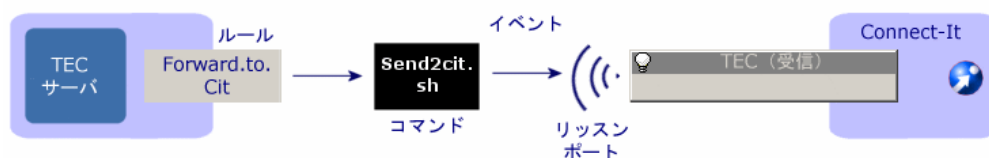
オプションコネクタ

- 生成（ソース）

受信モードのTivoli Enterprise Console（TEC）コネクタは、Tivoli Enterprise Consoleのイベントの受信を可能にするイベントコネクタです。

コネクタの設定内に指定されるリスンポートにより、TECが送信するイベントをリスンできるようになります。

TEC内で定義されるルールにより、TECコネクタへイベントを送信できるようになります。詳細については、本節の「SCAutoリスニングコネクタ設定ファイルを変更する [p. 264]」を参照してください。



TECコネクタ（受信）の互換性

コネクタは、以下のTECとのテストに成功しています。

- TEC 3.6x
- TEC 3.7x

イベント送信用のTECの設定

TECコネクタがイベントを受信できるようにするには、TEC内でルールを作成する必要があります。

TECのルール作成についてはTECの『ルールビルダ用ガイド』を参照してください。

TECコネクタ（受信）の設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 7.4. TECコネクタの設定

_____	コネクタに名前を付け、役割を説明する [p. 14]
_____	コネクタの導入 [p. 275]
_____	ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

コネクタの導入

このページでは、TEC（受信）コネクタ導入用のパラメータを設定できます。

リッスンポート

TECが送信するイベントを受信するためにコネクタが使用するリッスンポートを指定します。

このリッスンポートは、スクリプト `send2cit.sh` のパラメータ内に指定されているポートと一致します。本節の「イベントクラス「TEST_CLASS」のイベントを Connect-Itへ送信する [p. 278]」を参照してください。

これは必須フィールドです。

設定ファイル

TECイベントクラスを含むXMLファイルを指定します。このファイルのイベントクラスは、BAROCファイルで定義されているTECサーバのクラスに一致しなければなりません。

これは必須フィールドです。

TECデータベースのイベントクラスを含むデフォルトの記述ファイルが、Connect-Itに付属しています。ファイルは、[Connect-Itインストール先フォルダ] `config/tec/teccfg.xml` にあります。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

追加情報

teccfg.xml イベントクラスファイルを拡張する

イベントクラスのデフォルトの付属ファイルを、新規イベントクラスを追加するために拡張することも可能です。イベントクラスはBAROCファイル内で定義されています。

BAROCシンタックスを使用するイベントクラスの定義の例は以下の通りです。

```

ENUMERATION EXAMPLE_ENUM
  0 GA
  1 BU
  2 ZO
  3 ME
END

TEC_CLASS:
Example_class ISA Parent_class
DEFINES {
  slot_1 : STRING;
  slot_2 : INTEGER;
  slot_3 : INT32;
  slot_4 : EXAMPLE_ENUM;
  slot_5 : STRING, parse = no;
  slot_6 : INTEGER, default = 0;
  slot_7 : EXAMPLE_ENUM, default = GA;
  slot_9 : LIST_OF STRING, parse = no;
  slot_10 : LIST_OF INTEGER, default = [0,1];
  slot_11 : LIST_OF EXAMPLE_ENUM, default = [GA,BU,ZO,ME], parse = no;

  slot_12 : default = "dummy", parse = no;
};
END

```

BAROCファイルの拡張の例に関する注意

タイプは以下のように対応します。

- STRINGとリストデータはStringになります。
- INT32とINTEGERはIntegerになります。
- REALはDoubleになります。

LIST_OFタイプのフィールドはコレクションとして処理されます。例：LIST_OF STRINGフィールドは、Stringフィールドのコレクションになります。

継承は特別なXMLタグPARENTで表記されます。

クラスが親クラスのフィールドを再定義する場合（デフォルト値、タイプなど）、このフィールドはXML設定ファイル内で完全に再定義されなければなりません。変更されていない親クラスのフィールドは、自動的に新規クラスに追加されます。

TECコネクタのXMLシンタックスに翻訳されると、Example_classの定義は以下のようになります。

```

<STRUCTURE name="Example_class">
  <PARENT name="Parent_class"/>
  <ATTRIBUTE name="slot_1" type="String"/>
  <ATTRIBUTE name="slot_2" type="Integer"/>
  <ATTRIBUTE name="slot_3" type="Integer"/>
  <ATTRIBUTE name="slot_4" type="String"/>
  <ATTRIBUTE name="slot_5" type="String" parse="no"/>
  <ATTRIBUTE name="slot_6" type="Integer" default="0"/>
  <ATTRIBUTE name="slot_7" type="String" default="GA"/>
  <COLLECTION name="slot_9" type="String" parse="no"/>
  <COLLECTION name="slot_10" type="Integer" default="[0,1]"/>
  <COLLECTION name="slot_11" type="String" parse="no" default="[GA,BU,ZO,ME]"/>
  <ATTRIBUTE name="slot_12" type="String" parse="no" default="dummy"/>
</STRUCTURE>

```

 注意:

「slot_12」タイプおよび「parse」と「default」属性は、再定義されていない場合、親クラスから削除されなければなりません。

TECでイベント送信用ルールを作成する

TECがイベントクラスのイベントをConnect-Itへ送信出来るようにするには、コマンド「send2cit」を発信するルールをTEC内に作成する必要があります。

このコマンドは、TEC内で作成されたイベントをTECコネクタ（受信）へ送信します。

コマンドsend2citは、[Connect-Itのインストール先フォルダ]/datakit/tecevent_package.exeフォルダ内にあります。圧縮された自動実行可能ファイル「tecevent_package.exe」には、以下のファイルとフォルダが含まれています。

- send2cit.sh
- tecevent.jar
- tecevent_lang.jar
- 「getvalues.exe」ファイルを含むWindowsフォルダ
- 「getvalues」ファイルを含むSolaris、AIXとHP-UXフォルダ

これらの全ファイルと、使用するオペレーティングシステムに対応する「getvalues.exe」または「getvalues」ファイルは、TECサーバで同一のフォルダ内にコピーされなければなりません。

例 7.1. イベントクラス「TEST_CLASS」のイベントをConnect-Itへ送信する

イベントクラス「TEST_CLASS」のイベントを、Connect-Itへ送信するためのルールを作成するには、

- 1 新規ルールベースを作成します。
- 2 このルールベース内に、例えば「Forward」という名前の新規ルールセットを作成します。
- 3 このルールセット内に、例えば「fwd_test_class」という名前のルールを作成します。
- 4 このルールがイベントクラスTEST_CLASSに適用されることを指定します。
- 5 ルールのアクションを編集します。

ルールのアクションの編集

ルールのアクションを編集するには、

- 1 [イベント受信時] にアクションを作成します。
- 2 [コマンドの起動] を選択します。
- 3 コマンド「send2cit.sh」を選択します。
- 4 コマンドの引数を編集します（ [引き数の編集] ）。
 - -p XXX
「XXX」はTECコネクタ（受信）のリッスンポートに相当します。
この引数は**必須**です。
 - -s citserver
「citserver」は、Connect-Itがインストールされているサーバの名前です。
この引数はオプションです。この引数が指定されていないと、TECがインストールされているコンピュータがサーバになります。
 - -l logfile
「logfile」はログファイルの完全パスです。
この引数はオプションです。この引数が指定されていないと、ログファイルは作成されません。
- 6 ルールベース（Forward）を保存します。
- 7 ルールベースをコンパイルします。
- 8 ルールベースを読み込みます。

一度ルールベースを読み込むと、各TEST_CLASSイベントはTECに受信されず、イベントはConnect-Itへ送信されてからTECコネクタ（受信）により処理されます。

TECのルール作成についてはTECの『ルールビルダ用ガイド』を参照してください。

付属シナリオ

シナリオは提供されていません。

Webサービスコネクタ

オプションコネクタ

- 生成（ソース）
- 自発的な取り込み（ターゲット）

Webサービスコネクタを使用すると、Webサービスと相互作用することができます。Webサービスは、1つのサービスまたは複数の演算からなるサービスの集合を提供します。Webサービスが使用する主要プロトコルは、SOAP、HTTP GETとPOST、MIME、SMTPです。

SOAPメッセージのプロトコルは、XML言語に基づいています。これはDCOM規格とCORBA規格に類似しています。SOAPは、COM、DCOM、Internet ExplorerとMicrosoftのJava実装によってサポートされています。

Webサービスコネクタは、WSDL（Web Services Description Language、Webサービス記述言語）プロトコルを用いて、Webサービスの自動記述を取得します。この自動記述には、サービスが使用する通信プロトコルと、演算が処理できるフォーマットのリストが含まれています。

使用可能なWebサービスのリストは、<http://www.xmethods.com>などの他多数のサイトに記載されています。

統合シナリオでは、Webサービスコネクタはまずドキュメントを取り込み、そのドキュメントをWebサービスへ要求の形で送信します。コネクタは応答を受信すると、自動的に応答をドキュメントに変換して生成します。このコネクタはマッピングボックスのように機能し、送信するデータ型（要求）と受信するデータ型（要求への応答）の間のマッピングを実行します。

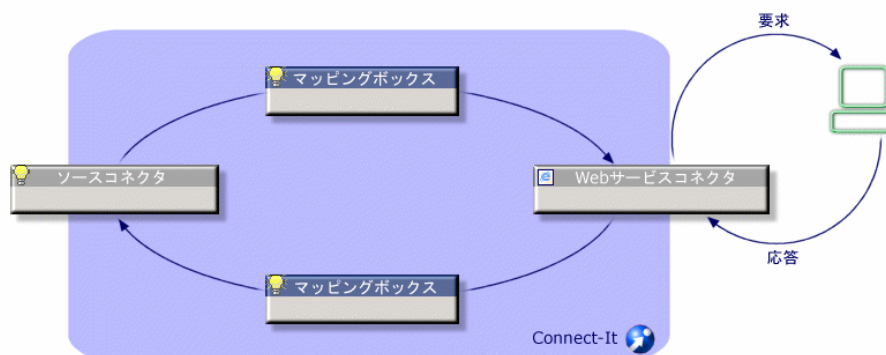
Webサービスコネクタの使用例

- 1 外部アプリケーションは郵便番号を含むドキュメントを生成し、マッピングはWebサービスコネクタが取り込めるように、このドキュメントを変換します。
- 2 コネクタは気温を提供するWebサービスへ要求を送信し、郵便番号をパラメータとして渡します。
- 3 Webサービスは、郵便番号に対応する気温を含む応答を返信します。

これを受信するWebサービスコネクタは、自動的にConnect-It内にドキュメントを生成します。

- 4 生成されるドキュメントは、郵便番号を提供した外部アプリケーションへ、フィードバックとして送信されます。

図 7.1. Webサービスコネクタ - 機能の概要



注意:

一部のWebサービスは要求 / 応答のモデルで機能しないため、この場合応答はありません。

Webサービスコネクタの制約点

コネクタは以下の操作をサポートしません。

- 同名の複数の演算
Webサービスコネクタは、1つのWebサービス内で固有名の演算のみをサポートします。Webサービスが同一名の2つの演算を参照すると、第1の演算のみがコネクタの使用可能なドキュメントタイプとして発行されます。
- 複数の応答がある要求
Webサービスコネクタは、各要求（取り込み用ドキュメントタイプ）に、1つの応答（生成用ドキュメントタイプ）を関連付けます。Webサービスが1つの要求に複数の応答を関連付けると、最初の演算のみがコネクタの使用可能なドキュメントタイプとして発行されます。
- 複数の要求がある応答

Webサービスが1つの応答に複数の要求を関連付けると、最初の演算のみがコネクタの使用可能なドキュメントタイプとして発行されます。

- 再帰的な構造のドキュメントタイプ

Webサービスコネクタは、同一ドキュメントタイプ内で1要素が別の要素を参照する構造のドキュメントタイプを発行しません（例：リレーショナルデータベースのモデル）。

- Webサービスに処理されるフォーマットを記述する部分のWSDLへ、別のフォーマットをインポートする

表 7.5. サポートされるプロトコル - Webサービスコネクタ

プロトコル	サポート可 / 不可
SOAP RPC	可
SOAP Messaging (HTTPプロトコルによる SOAP)	可
HTTP get	可
HTTP post	可
SOAP FTP	不可
SOAP SMTP	不可
MIME	不可

Webサービスコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 7.6. Webサービスコネクタの設定

☰ _____ コネクタに名前を付け、役割を説明する [p. 14]

☰ _____ コネクタの導入 [p. 282]

☰ _____ ピポットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

コネクタの導入

このページでは、Webサービスへのコネクタの接続パラメータを設定できます。

WSDLのアドレス

WSDL URLアドレスによって、各Webサービス用に以下の内容を取得できるようになります。

- 使用される通信プロトコル
- サポートされる演算操作
- 演算のフォーマット (XMLスキーマ、DTD、Javaオブジェクトなど)

例 : <http://www.xmethods.net/sd/2001/TemperatureService.wsdl>

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

Webサービスコネクタの取り込み用ルール

コネクタの取り込み用ルールの入力方法については、「コネクタのルール (ディレクティブ) [p. 39]」章の「取り込み用ルール [p. 43]」の節を参照してください。

適用する演算

コネクタがデータをWebサービスサーバへ送信できるようにするには、演算「query」を選択します。

[適用する演算] フィールドで「なし」を選択しても、「query」演算が使用されます。この場合、この演算が実行されたことを示す警告メッセージが、ドキュメントログ内に表示されます。

使用可能なドキュメントタイプ

Webサービスコネクタは、拡張型処理レポートに対応する使用可能なドキュメントタイプを、生成用に発行します。

処理レポートの詳細については、『ユーザガイド』の「処理レポート」の章を参照してください。

この拡張内容は、処理レポートのルートノード下に、Webサービスの演算に対応する要素として表示されます。

例：XMethods Temperature Serviceサービスは2つのメソッドGetTempとGetWeatherを発行します。サービスに送信される要求内では、この2つのメソッド用に、郵便番号に対応する文字列型のパラメータが予期されています。

取り込み用に使用可能なドキュメントタイプは、Webサービスの演算に当たります。

各演算ごとに、1つの使用可能なドキュメントタイプが取り込み用（要求の送信）と生成用（要求への応答）にそれぞれ発行されます。

使用可能なドキュメントタイプの名前

表 7.7. 拡張型処理レポートの名前 - Webサービスコネクタ

プロトコル	生成用に使用可能なドキュメントタイプ (拡張型処理レポート)の接頭文字
SOAP RPC	ProcessReportSoapRpc-[演算名]
SOAP Messaging (HTTPプロトコルによるSoap)	ProcessReportSoapHttp-[演算名]
HTTP Get	ProcessReportHttpGet-[演算名]
HTTP Post	ProcessReportHttpPost-[演算名]

注意:

Webサービスコネクタは、サポートされていないプロトコルを使用する演算を、使用可能なドキュメントタイプとしては発行しません（例：MIMEプロトコル）。

Webサービスコネクタの使用可能なドキュメントタイプの名前には、通信プロトコルを表す接頭文字と、演算の名前が含まれます。生成用ドキュメントタイプでは、名前に接頭文字ProcessReportが付きます。

付属シナリオ - Webサービスコネクタ

このコネクタを使用する付属シナリオは以下の通りです。

- ws/wsac41/currency.scn [p. 328]

JDBCコネクタ

オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

JDBCコネクタは、SQLクエリ言語を使用するデータベースへのアクセスを可能にするJava DataBase Connectivityインタフェースを使用します。

JDBCコネクタの互換性

JDBCコネクタは、以下のJDBCドライバとのテストに成功しています。

- Sun Systems sun.jdbc.odbc.JdbcOdbcDriver
- Microsoft SQL Server 2000 for JDBC
- Sybase jConnect 4.5/5.5
- Oracle JDBC 9.01

JDBCコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定 [p. 13]」章の「コネクタを設定する [p. 14]」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 7.8. JDBCコネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]
コネクタの導入 [p. 284]
スケジュールのポイントを設定する [p. 32]
トランザクションを設定する [p. 34]
ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

コネクタの導入

このページでは、データベースへの接続パラメータを設定できます。

URL

[JDBCドライバ] フィールドにデフォルトで指定されるドライバと共に、このフィールドでは、シンタックス「jdbc:odbc:[データソース名]」を使用しなければなりません。

例 : jdbc:odbc:AC420FR

デフォルトユーザ

データベースへ接続するユーザに対応する名前を入力します。

デフォルトのパスワード

デフォルトのユーザに関連付けられたパスワードを入力します。

JDBCドライバ

このパラメータのデフォルト値は、「sun.jdbc.odbc.JdbcOdbcDriver」です。このドライバにはJREまたはJDKが付属しています。この汎用ドライバに問題が発生する場合、ユーザはデータベースサーバに適したドライバを使う必要があります。

メタ記述ファイル

このフィールドには、JDBCコネクタのカスタマイズ情報に対応する「customizedSchema.xml」ファイルのパスを入力します。

このファイルは、Connect-Itと共に提供されているStudioバージョンを使って編集されます。Connect-ItとStudio間の同期機能を使用する場合、このファイルは同期操作のたびに自動的に更新されます。

Studioを使ってJavaリソースコネクタを拡張する方法については、『Peregrine Studio 3.1.0 - User's guide』の「Interaction between the Scenario builder and Studio」の章を参照してください。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「サーバとの時間差を指定する [p. 29]」の節を参照してください。

トランザクションを設定する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「トランザクションを設定する [p. 34]」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

JDBCコネクタの生成用ルール

コネクタの生成用ルールの入力方法については、「コネクタのルール（ディレクティブ） [p. 39]」章の「生成用ルール [p. 40]」の節を参照してください。

このコネクタでは、これらのルールはWHERE句やORDERBY句で作成されており、ソースデータベースのレコードにフィルタをかけることができます。これらの句の詳細については「コネクタのルール（ディレクティブ） [p. 39]」章の「生成用ルール [p. 40]」節、「WHERE句とORDER BY句 [p. 41]」を参照してください。

JDBCコネクタの取り込み用ルール

コネクタの取り込み用ルールの入力方法については、「コネクタのルール（ディレクティブ） [p. 39]」章の「取り込み用ルール [p. 43]」の節を参照してください。

取り込み用ルールを作成するには、**[整合性]**と**[高度な整合性チェック]**タブ内で整合性用のパラメータを入力します。

これらの整合性用タブの詳細については、「コネクタのルール（ディレクティブ） [p. 39]」章の「取り込み用ルール [p. 43]」節内「整合性 [p. 44]」を参照してください。

付属シナリオ - JDBCコネクタ

このコネクタ用のシナリオはありません。

8 | ERPコネクタ

ERP（Enterprise Resource Planning、企業資源計画）コネクタは、ERPアプリケーションのデータを処理します。

SAPコネクタ

Connect-Itでは以下のSAPコネクタが提供されています。

- [SAP BAPIコネクタ \[p. 289\]](#)
このコネクタは、SAP R/3サーバに接続するサービスに対応する使用可能なドキュメントタイプを発行します。各サービスはBAPIメソッドを呼び出します。このコネクタではデータの同期処理が可能です。
- [SAP IDocコネクタ \[p. 294\]](#)
このコネクタは、IDOCフォーマットのフラットテキストファイルを使用するサービスに対応する、使用可能なドキュメントタイプを発行します。このIDOCファイルによって、SAP R/3サーバへデータを送信し、SAP R/3サーバからデータを取得することが可能になります。

RFC (Remote Function Calls)

RFC (リモートファンクションコール) は、SAP BAPIコネクタで使用されています。RFCは、SAPまたは非SAPアプリケーションから、ネットワーク上のSAP R/3サーバ機能へアクセスできるようにします。

RFCの機能や特徴は以下の通りです。

- RFCは、SAPサーバで既定されたプロシージャを呼び出して処理します。
- 通信制御、パラメータの転送やエラーメッセージを管理します。
- RFCはSAPの標準認証プロシージャで制御されています。認証プロシージャは、アプリケーションデータが安全かつ一貫性のある方法で読み取られ編集されているかどうかを、検証します。

SAP R/3サーバが提供する数千の関数の大部分には、遠隔アクセスできます。関数は通常、ABAP/4 (Advanced Business Application Programming) 言語で書かれています。

RFCは双方向で機能し、SAPアプリケーションクライアントが、非SAPアプリケーション内の関数へアクセスできるようにします。

RFCライブラリとJavaアーカイブ

SAPコネクタを使用するには、以下の方法で入手できるJavaアーカイブ (.jar) とJavaネイティブライブラリ (.dll) を使用する必要があります。

- 1 インターネットブラウザを起動します。
- 2 アドレス<http://service.sap.com/connectors>を入力します。

注意:

このサイトにアクセスするには、SAPサポートから認証証明書を手に入れる必要があります。

- 3 [SAP Java Connector] 項目を選択します。
- 4 [Tools & Services] 項目を選択します。
- 5 [Download SAP JCo Release 2.0.6] をクリックします。
このダウンロードにより、SAPコネクタ使用に必要なファイルを含む圧縮ファイル (.zip) を入手できます。

Connect-Itのインストール先フォルダに以下のファイルをコピーします。

- 「bin32」サブフォルダに以下のライブラリをコピーします。
 - librfc32.dll
 - sapjcorfp.dll
- 「lib」サブフォルダに「sapjco.jar」アーカイブをコピーします。

SAP BAPIコネクタ

オプションコネクタ

- 取り込み（ターゲット）
- 取り込み - 生成（要求 - 応答）

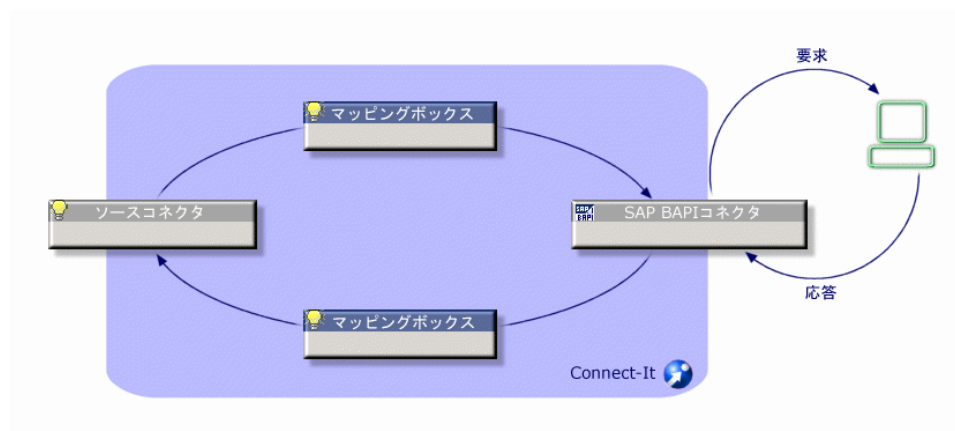
SAP BAPIコネクタはSAP R/3サーバに接続し、ビジネスオブジェクト（Business Object）によって分類されたサービス（BAPI）のリストを取得します。Connect-Itで、これらのサービスはコネクタの使用可能なドキュメントタイプに対応します。

SAP BAPIコネクタが発行する使用可能なドキュメントタイプは以下のように分類されます。

- 取り込み用ドキュメントタイプ
これらのドキュメントタイプは、コネクタがSAPサーバへデータを送信できるようにします。
- 取り込み - 生成用ドキュメントタイプ
これらのドキュメントタイプは、コネクタが以下の操作を実行できるようにします。
 - SAPサーバへ要求を送信する（取り込み）
 - この要求の応答を受信する（生成）

取り込み - 生成用ドキュメントタイプを使用するSAPコネクタは、以下の図のように機能します。

図 8.1. SAPコネクタ - 取り込み - 生成用ドキュメントタイプ



SAP BAPIコネクタの互換性

このコネクタは、SAP R/3の4.5 Bバージョンとのテストに成功しています。
Peregrine Systemsサイト (<http://www.peregrine.com>) にある、Connect-Itの本バージョンの動作環境の表もご参照ください。

ABAPデータ型

ABAPデータ型とConnect-Itデータ型の対応関係は、次の表の通りです。

表 8.1. ABAPデータ型とConnect-Itデータ型の対応関係

ABAPデータ型	Connect-Itデータ型
TYPE_CHAR	テキスト
TYPE_DATE	日付
TYPE_BCD	倍精度整数 (BigDecimal)
TYPE_TIME	日付
TYPE_BYTE	Blob
TYPE_NUM	整数 (BigInteger)
TYPE_FLOAT	倍精度整数 (BigDecimal)
TYPE_INT	整数
TYPE_INT1	整数
TYPE_INT2	整数
TYPE_STRUCTURE	無視されます
TYPE_ITAB	無視されます
TYPE_DATE_1	サポートされていません
TYPE_DATE_2	サポートされていません
TYPE_NULL	サポートされていません
TYPE_WIDE_2	サポートされていません
TYPE_WIDE_4	サポートされていません
TYPE_IUNKNOWN	サポートされていません
TYPE_WSTRING	サポートされていません
TYPE_SAPAUTOMATION	サポートされていません
TYPE_STUB	サポートされていません
TYPE_WCHAR	サポートされていません

SAP BAPIコネクタの制約点

このコネクタは、SAP R/3の3.5 (BAPIの使用を導入したバージョン) より前のバージョンでは機能しません。

SAP BAPIコネクタの設定

ウィザードでコネクタを設定できます。

Asset Managementコネクタなどの他のコネクタと違い、このコネクタの設定ウィザードでは、コネクタが発行する使用可能なドキュメントタイプを編集し、充実させます（ドキュメントタイプの作成、削除または変更）。

コネクタの設定ウィザードの各ページは以下の通りです。

表 8.2. SAP BAPIコネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]
コネクタの導入 [p. 291]
ドキュメントタイプの設定 [p. 292]
ドキュメントタイプに名前を付ける [p. 293]
ドキュメントタイプの内容を定義する [p. 293]
ドキュメントタイプのプレビュー [p. 293]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

コネクタの導入

このページでは、SAPサーバへのコネクタ接続パラメータを設定できます。

[SAPホストのターゲット]

SAPサーバの名前を指定します。

[クライアント]

クライアントポートを入力します。

例：800

[ユーザ名]

ユーザ名を指定します。

[パスワード]

前のフィールドで指定されているユーザ名に関連付したパスワードを入力します。

[言語]

サーバから来る情報をどの言語で表示するかを指定します。

例：英語にはen、ドイツ語にはdeなど。

[システム番号]

SAPサーバと共に使用するシステム番号を指定します。

ドキュメントタイプの設定

このページでは、コネクタが発行する使用可能なドキュメントタイプを編集できます。

ドキュメントタイプを作成、変更、削除するには、対応するボタンをクリックします。

取り込みモードのドキュメントタイプを作成する

- 1 [作成]をクリックします。
- 2 ダイアログボックスが表示されます。
- 3 [取り込み用] オプションを選択します。
- 4 [OK]をクリックします。
- 5 コネクタの使用可能なドキュメントタイプを作成するために、表示されるページに値を入力します。

取り込み - 生成用ドキュメントタイプを作成する

- 1 [作成]をクリックします。
- 2 ダイアログボックスが表示されます。
- 3 [取り込み - 生成用] オプションを選択します。
- 4 [OK]をクリックします。
- 5 コネクタの使用可能なドキュメントタイプを作成するために、表示されるページに値を入力します。

ドキュメントタイプを変更する

- 1 [ドキュメントタイプのリスト] 枠内で、変更するドキュメントタイプを選択します。

- 2 **[編集]** をクリックします。
- 3 ドキュメントタイプ (サービス) 作成用の3ページをまとめたページが表示されます。
- 4 コネクタの使用可能なドキュメントタイプを変更するために、各ページに値を入力します。

ドキュメントタイプを削除する

- 1 **[ドキュメントタイプのリスト]** 枠内で、削除するドキュメントタイプを選択します。
- 2 **[削除]** をクリックします。

ドキュメントタイプに名前を付ける

このページでは、ドキュメントタイプの名前と説明を入力できます。
この名前は、コネクタの同一インスタンス内で固有でなければなりません。

ドキュメントタイプの内容を定義する

このページでは、ドキュメントタイプで表されるBAPIを選択できます。

SAPビジネスオブジェクト

このリストでは、使用中のSAPサーバで使用可能なビジネスオブジェクト順に並べられたBAPIを、選択できます。例：会社の定義に対応する「COMPANY」ビジネスオブジェクト。

BAPIメソッドに対応するドキュメントタイプを定義する

- 1 リストのSAPビジネスオブジェクトを1つ選択しクリックします。
このビジネスオブジェクト用の使用可能なBAPIリストが表示されます。
- 2 BAPIメソッドを選択します。

RFC検索

このフィールドではBAPIの名前でBAPIを検索できます。例：
BAPI_APPLICANT_CREATE

ドキュメントタイプのプレビュー

このページが表示されるようにするには、JVM設定ダイアログボックス (**[Java / JVMを設定する]** メニュー) で、 **[デバッグモード]** を選択しなければなりません。

このページでは、ドキュメントタイプのプレビューを取得できます。また、元のドキュメントタイプでカスタマイズされたビジネスオブジェクトにドキュメントタイプが対応するように、ドキュメントタイプをこのページで編集することもできます。例：ビジネスオブジェクトのフィールドのタイプ「テキスト」が「整数」に変更された場合。

! 警告:

ドキュメントタイプのプレビューページの変更は推奨されていません。

SAP IDocコネクタ

オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

SAP IDocコネクタは、SAP R/3サーバのデータを送信または受信し、同時に、ネットワーク共有フォルダやFTPサーバにあるIDocファイルで、データの読み取りまたは書き込みを実行します。IDocファイルを使うと、発注や配達の納期などのSAPの標準データにアクセスできるようになります。各IDocファイルは、あるIDocタイプに合致しています。

SAP R/3は約100種類のIDocタイプを提供します。ユーザが各自にIDocタイプを作成することや、既存のIDocタイプを拡張することも可能です。

各IDocファイルには3つの部分が含まれています。

- コントロールレコード
このレコードは、SAP R/3サーバがIDocファイルを一意な方法で識別できるようにする文字列を含みます。
- データレコード
これらのデータはセグメントごとに構成されています。
- ステータスレコード
これらのステータスは、SAP R/3サーバからまたはSAP R/3サーバへと通過するIDocファイルのステータスです。

SAP IDocコネクタの互換性

このコネクタは、SAP R/3の4.5 Bバージョンとのテストに成功しています。

SAP IDocコネクタの設定

ウィザードでコネクタを設定できます。

Asset Managementコネクタなどの他のコネクタと違い、このコネクタの設定ウィザードでは、コネクタが発行する使用可能なドキュメントタイプを編集します（生成用または取り込み用ドキュメントタイプの作成、削除または変更）。

取り込み用ドキュメントタイプは、IDocファイルヘータを送信することに当たります。

生成用ドキュメントタイプは、IDocファイル内のデータを読み取ることに当たります。

コネクタの設定ウィザードの各ページは以下の通りです。

表 8.3. SAP IDocコネクタの設定

コネクタに名前を付け、役割を説明する [p. 14]
コネクタの導入 [p. 295]
ドキュメントタイプの設定 [p. 296]
ドキュメントタイプ（サービス）に名前を付ける [p. 297]
IDocタイプを取得する [p. 297]
IDocタイプを選択する [p. 298]
コントロールパラメータを入力する [p. 298]
IDocファイルの接尾語を指定する [p. 299]
オプションのエンコードを指定する [p. 299]
ドキュメントタイプのプレビュー [p. 299]
ピボットドキュメントタイプの使用 [p. 36]

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定 [p. 13]」章の「コネクタに名前を付け、役割を説明する [p. 14]」の節を参照してください。

コネクタの導入

このページでは、SAPサーバへのコネクタ接続パラメータを設定できます。

[ディレクトリ]

コネクタがどのフォルダでIDocファイルを読み書きするかを指定します。

[FTP使用可能]

FTPサーバでのIDocファイル転送には、このオプションを選択します。

[FTPサーバ]

ネットワーク上のFTPサーバの名前を指定します。

[ユーザ名]

サーバのユーザ名を指定します。

[パスワード]

前のフィールドで指定されているユーザ名に関連付したパスワードを入力します。

[テスト] ボタン

このボタンを押すと、FTP接続のパラメータが有効かどうかをテストできます。

ドキュメントタイプの設定

このページでは、コネクタが発行する使用可能なドキュメントタイプを編集できます。

ドキュメントタイプを作成、変更、削除するには、対応するボタンをクリックします。

取り込みモードのドキュメントタイプを作成する

- 1 [作成] をクリックします。
- 2 ダイアログボックスが表示されます。
- 3 [取り込み用] オプションを選択します。
- 4 [OK] をクリックします。
- 5 コネクタの使用可能なドキュメントタイプを作成するために、表示されるページに値を入力します。

生成モードのドキュメントタイプを作成する

- 1 [作成] をクリックします。
- 2 ダイアログボックスが表示されます。
- 3 [生成用] オプションを選択します。
- 4 [OK] をクリックします。

- 5 コネクタの使用可能なドキュメントタイプを作成するために、表示されるページに値を入力します。

ドキュメントタイプを変更する

- 1 **【ドキュメントタイプのリスト】**枠内で、削除するドキュメントタイプを選択します。
- 2 **【編集】**をクリックします。
- 3 ドキュメントタイプ(サービス)作成用のページをまとめたページが表示されます。
- 4 コネクタの使用可能なドキュメントタイプを変更するために、各ページに値を入力します。

ドキュメントタイプを削除する

- 1 **【ドキュメントタイプのリスト】**枠内で、削除するドキュメントタイプを選択します。
- 2 **【削除】**をクリックします。

ドキュメントタイプ(サービス)に名前を付ける

このページでは、ドキュメントタイプの名前と説明を入力できます。
この名前は、コネクタの同一インスタンス内で固有でなければなりません。

IDocタイプを取得する

このページではIDocタイプを取得する方法を選択できます。

- ローカルのIDocタイプ
このオプションを選択すると、コンピュータに既存するIDocタイプを使用できます。
- SAPサーバのIDocタイプ
このオプションを選択すると、SAPサーバのIDocタイプを動的に取得できます。**【接続】**をクリックすると、ウィザードでSAPサーバへの接続を定義できるようになります。

SAPサーバへ接続する

ウィザードの最初のページでは以下の接続パラメータを指定できます。

- SAPサーバ
- クライアント
- ユーザ
- パスワード

- システム番号
第2ページでは、接続パラメータをテストできます。

IDocタイプを選択する

このページではIDocタイプを選択できます。

IDocタイプ

[ローカルのIDocタイプ] オプションを選択した場合は、[エクスプローラ] ボタンを押し、コンピュータ内のIDocタイプの場所を指定します。

[SAPサーバのIDocタイプ] オプションを選択した場合、IDocタイプの名前を入力し、[検証] ボタンを押します。IDoc名が正しい場合は、[検証] ボタンの下に「IDocメタデータが取得/検証されました。」というメッセージが表示されます。

CIMタイプ

このフィールドではCIMタイプを入力できます。2つのドロップダウンリストから以下のバージョンを選択し指定します。

- IDocタイプ
バージョン2または3
- SAPサーバ
使用可能なバージョン : 3.1 Gから4.6 C

コントロールパラメータを入力する

このページでは、このドキュメントタイプに基づいたIDocファイルを、SAPサーバが制御できるようにするパラメータを指定できます。

送信ポート

このIDocドキュメントタイプ用に使用する送信ポートを指定します。

受信ポート

このIDocドキュメントタイプ用に使用する受信ポートを指定します。

バージョン

使用されるSAPのバージョンを指定します。

IDocファイルの接尾語を指定する

このページでは、このドキュメントタイプに基づくIDocファイルの接尾語を入力できます。ドロップダウンリストの接尾語から選択しなければなりません。

オプションのエンコードを指定する

このページでは、IDocファイルデータのオプションのエンコードタイプを指定できます。

例：ISO - 8859 - 1

ドキュメントタイプのプレビュー

このページが表示されるようにするには、JVM設定ダイアログボックス（[Java / JVMを設定する]メニュー）で、[デバッグモード]を選択しなければなりません。

このページでは、ドキュメントタイプのプレビューを取得できます。また、カスタマイズされたIDocタイプにドキュメントタイプが対応するように、ドキュメントタイプをこのページで編集することもできます。

警告:

ドキュメントタイプのプレビューページの変更は推奨されていません。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定 [p. 13]」章の「ピボットドキュメントタイプの使用 [p. 36]」の節を参照してください。

9 | 付属シナリオ

この章では、Connect-Itの付属シナリオについて説明します。

本章で説明されているシナリオは、ソースコネクタごとにまとめられており、また3つの種類に分類されています。

- **用例シナリオ**


この種のシナリオは、コネクタの機能を説明するために作成されたものです。実務的な価値はないため、プロダクションモードでは**決して使用しないでください**。

本章の用例シナリオには、アイコン  が付いています。

- **実務シナリオ（変更必要）**


この種のシナリオは、実際の統合プロセスに対応したものです。しかし、プロダクションモードで使用するためには、ユーザが状況に応じて変更する必要があります。

特に、ソースコネクタとターゲットコネクタ間のマッピングを変更する必要があります。

本章の実務シナリオ（変更必要）には、アイコン  が付いています。

- **実務シナリオ**

この種のシナリオは、実際の統合プロセスに対応したものです。プロダクションモードで直接使用可能です。

本章の実務シナリオには、アイコン  が付いています。

Unicenter AMOシナリオ

本節では、Unicenter AMOコネクタを使用する付属シナリオについて簡潔に説明します。

amo /amoac36 /amoac.scn



このシナリオは、Unicenter AMOデータベースを、ピボットドキュメントタイプを経由してAsset Managementアプリケーションへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア
- 関連付けられた周辺機器

InfraTools Desktop Discoveryシナリオ

本節では、InfraTools Desktop Discoveryコネクタを使用する付属シナリオについて簡潔に説明します。

idd /iddac36 /iddac.scn



このシナリオは、Asset Management 3.6データベースの [資産] テーブルのコンピュータに関する情報を更新します。

idd /iddac40 /iddac.scn



このシナリオは、Asset Management 4.0データベースの [資産] テーブルのコンピュータに関する情報を更新します。

idd /iddac41 /iddac.scn



このシナリオは、Asset Management 4.1データベースの [コンピュータ] テーブルのレコードに関する情報を更新します。

idd /iddac42 /iddac.scn



このシナリオは、Asset Management 4.2データベースの [コンピュータ] テーブルのレコードに関する情報を更新します。

idd /idrrsm4 /iddam.scn



このシナリオは、Remedy Action Request Systemsデータベースを、以下の内容に関するデータを用いて更新します。

- 資産
- コンピュータ
- プリンタ

idd /idrrsm4 /iddhd.scn



このシナリオは、Remedy Action Request Systems (HelpDesk) データベースを、コンピュータに関するデータを用いて更新します。

idd /idrrsm4 /idrrsm.scn



このシナリオは、Remedy Action Request Systemsデータベースを、コンピュータに関するデータを用いて更新します。

idd /iddsc4 /iddsc.scn



このシナリオは、ServiceCenterデータベースを以下の内容に関するデータを用いて更新します。

- コンピュータ

- 連絡先
- ソフトウェア
- プリンタ

Intel LANDesk シナリオ

本節では、Intel LANDeskコネクタを使用する付属シナリオについて簡潔に説明します。

ild /ld64ac36 /ldac.scn



このシナリオは、Intel LANDesk 6.3と6.4のデータベースを、AssetCenter 3.6データベースへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア
- 周辺機器

ild /ld65ac36 /ldac.scn



このシナリオは、Intel LANDesk 6.5のデータベースを、AssetCenter 3.6データベースへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア
- 周辺機器

ild /ld65ac41 /ldac.scn



このシナリオは、Intel LANDesk 6.5のデータベースを、AssetCenter 4.1データベースへ移行します。

移行されるデータは以下の通りです。

- コンピュータ

- ソフトウェア
- 周辺機器

InfraTools Managementシナリオ

本節では、InfraTools Managementコネクタを使用する付属シナリオについて簡潔に説明します。

ntsec /ntim5 /ntim.scn



このシナリオは、NTドメイン特有の情報（ユーザとコンピュータ）をInfraToolsデータベースへ転送します。これらの情報は、InfraTools Administratorモジュールで使用されます。InfraTools Administratorモジュールは、リモートコントロール操作をセキュリティで保護するためにこれらのデータを使用します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
MachineSrc	amComputerDst
UserSrc	amEmplDeptDst

InfraTools Network Discoveryシナリオ

本節では、InfraTools Network Discoveryコネクタを使用する付属シナリオについて簡潔に説明します。

ind /indac36 /indac.scn



このシナリオは、InfraTools Network Discoveryが監視するIT資産に関するデータを、Asset Management 3.6アプリケーションの資産のテーブルへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
FULL_DEVICE	amAsset
FULL_DEVICE_CNX	amAssetCnx
EVENT	amAsset1
EVENT	amAssetDst
EVENT	amTicketDst

ind /indac40 /indac.scn



このシナリオは、Asset Management 4.0アプリケーションで、ind /indac36 /indac.scn [p. 305]シナリオと同じ機能を果たします。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
FULL_DEVICE	amAsset
FULL_DEVICE_CNX	amAssetCnx

ind /indac41 /indac.scn



このシナリオは、Asset Management 4.1アプリケーションで、ind /indac36 /indac.scn [p. 305]シナリオと同じ機能を果たします。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
FULL_DEVICE	amAsset
FULL_DEVICE_CNX	amAssetCnx

ind /indrsm4 /indrsm.scn



このシナリオは、InfraTools Network Discoveryが監視するIT資産に関するデータを、Action Request Systemデータベースへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
FULL_DEVICESrc	IND:InventoryDst

ind /indsc.scn



このシナリオは、InfraTools Network Discoveryが監視するIT資産に関するデータを、ServiceCenterデータベースへ移行します。

Action Request Systemシナリオ

本節では、Action Request Systemコネクタを使用する付属シナリオについて簡潔に説明します。

rsm /rsm4ac36 /cmpo.scn



このシナリオは、Action Request Systemデータベースのデータを、AssetCenter 3.6データベースへ移行します。

このシナリオによって移行されるデータは以下の通りです。

- 購入依頼
- カテゴリ
- モデル

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
CHG:ChangeSrc (Request creation)	amRequestDst1
CHG:ChangeSrc (Category creation)	amCategoryDst1
CHG:ChangeSrc (Model creation)	amProductDst

rsm /rsm4ac36 /sharedat.scn



このシナリオは、Action Request Systemデータベースのデータを、AssetCenter 3.6データベースへ移行します。

このシナリオによって移行されるデータは以下の通りです。

- 従業員
- 場所
- 資産

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
SHR:PeopleSrc1	amEmplDeptDst1
SHR:LocationSrc	amLocationDst
AST:AssetSrc	amAssetDst

LDAPシナリオ

本節では、LDAPコネクタを使用する付属シナリオについて簡潔に説明します。

ldap /ldapac36 /complete.scn



このシナリオは、LDAPディレクトリのデータを、Asset Managementアプリケーションの部署と従業員のテーブルへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
inetOrgPersonSrc	amEmplDeptDst

ldap /ldapac36 /simple.scn



このシナリオは、LDAPディレクトリのデータを、Asset Managementアプリケーションの部署と従業員のテーブルへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
organizationalPersonSrc	amEmplDeptDst

ldap /ldapnote /hames.scn



このシナリオは、LDAPディレクトリのデータを、Lotus Notesデータベースへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
organizationalPersonSrc1	PersonDst

Eメールシナリオ

本節では、Eメールコネクタを使用する付属シナリオについて簡潔に説明します。

mail /mailac36 /finreque.scn



このシナリオは、依頼のテーブルのレコードに対応する購入依頼を、Eメールアドレス（購入責任者のアドレス）へ送信します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
amRequestSrc	OutMailMessageDst

mail/mailac36/finconfi.scn



このシナリオは、購入依頼の承認のメッセージを送信します。Asset Managementアプリケーションの購入依頼のテーブル内で、依頼のステータスが変更されません。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
InMailMessageSrc	amRequestDst

mail/mailac36/hewemplo.scn



このシナリオは、Eメールメッセージに基づいて、Asset Managementアプリケーションの従業員のテーブル内にレコードを作成します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
InMailMessageSrc	amEmplDeptDst

mail/mailac40/finconfi.scn



このシナリオは、購入依頼の承認のメッセージを送信します。Asset Managementアプリケーションの購入依頼のテーブル内で、依頼のステータスが変更されません。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
InMailMessageSrc	amRequestDst

mail/mailac40/finreque.scn



このシナリオは、依頼のテーブルのレコードに対応する購入依頼を、Eメールアドレス（購入責任者のアドレス）へ送信します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
amRequestSrc	OutMailMessageDst

mail/mailac40/hewemplo.scn



このシナリオは、Eメールメッセージに基づいて、Asset Managementアプリケーションの従業員のテーブル内にレコードを作成します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
InMailMessageSrc	amEmplDeptDst

MQSeriesシナリオ

本節では、MQSeriesコネクタを使用する付属シナリオについて簡潔に説明します。

mqseries/acmq/mqtoprgn.scn



このシナリオは、mqseriesメッセージに基づいて、Asset Managementアプリケーションの以下のテーブル内にレコードを作成します。

- コストセンタのテーブル（amCostCenter）
- サプライヤのテーブル（amCompany）
- 発注のテーブル（amPOrder）
- 受領のテーブル（amDeliv）

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
CostCenterSrc	amCostCenterDst
VendorSrc	amCompanyDst
ExtPOAckSrc	amPOOrderDst
ExtReceiptAckSrc	amDelivDst

mqseries /acmq/prgntomq.scn



このシナリオは、

- Asset Managementアプリケーションの以下のテーブルのレコードに基づいて、mqseriesメッセージを作成します。
 - 発注のテーブル (amPOOrder)
 - 受領のテーブル (amDeliv)
- 上記のテーブルを以下のキューに基づいて更新します。
- PEREGRINE.OUTキュー
- PEREGRINE.INキュー

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
amPOOrderSrc	PO
amDeliv	Receipt

mqseries /scmq/mqsc.scn



このシナリオは、

- PEREGRINE.INキューにあるMQSeriesメッセージに基づいて、ServiceCenterのcontactファイル内にレコードを作成します。
- ServiceCenterのcontactファイルのレコードに基づいて、PEREGRINE.OUT.PERSONキュー内にMQSeriesメッセージを作成します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
External Contacts	Contacts from Exterior
Contacts of ServiceCenter	Contacts from ServiceCenter

NTセキュリティシナリオ

本節では、NTセキュリティコネクタを使用する付属シナリオについて簡潔に説明します。

ntsec /htac36 /addcpu.scn



このシナリオは、指定されたNTドメイン上のコンピュータのNT情報を、AssetCenter 3.6データベースへインポートします。

ntsec /htac36 /adduser.scn



このシナリオは、指定されたNTドメイン上の従業員と部署のNT情報を、AssetCenter 3.6データベースへインポートします。

ntsec /htac40 /addcpu.scn



このシナリオは、指定されたNTドメイン上のコンピュータのNT情報を、AssetCenter 4.0データベースへインポートします。

ntsec /htac40 /adduser.scn



このシナリオは、指定されたNTドメイン上の従業員と部署のNT情報を、AssetCenter 4.0データベースへインポートします。

ntsec /htac41 /addcpu.scn



このシナリオは、指定されたNTドメイン上のコンピュータのNT情報を、AssetCenter 4.1データベースへインポートします。

ntsec /htac41 /adduser.scn



このシナリオは、指定されたNTドメイン上の従業員と部署のNT情報を、AssetCenter 4.1データベースへインポートします。

ServiceCenterシナリオ

本節では、ServiceCenterコネクタを使用する付属シナリオについて簡潔に説明します。

「acscups /dataprop」フォルダのシナリオは、シナリオの接頭番号の順番で実行されなければなりません。例えば、01-SCcompany2ACamCompany.scnシナリオを1番目に実行してから、02-SCLocation2ACamLocation.scnシナリオを実行します。

scac /sc3ac36 /scac.scn



この複製（レプリケーション）シナリオは、Asset 3.6のデータベースからServiceCenter3.6へデータを作成または更新します。

Asset Managementアプリケーションの設定

本節では、シナリオを使用できるようにAsset Managementアプリケーションを設定する方法を説明します。

「amdb.ini」ファイル

このファイルは、Asset Managementアプリケーションがインストールされているコンピュータ上に作成されているデータベース接続のリストを含んでいます。このファイルは、Asset Managementアプリケーションのインストール時に自動的に作成され、Windowsのシステムフォルダに含まれます。

例

C:WINNT\amdb.ini.

Asset Managementコネクタは、このファイルを読み込み、コネクタの設定時に使用可能な接続のリストを表示します。使用中のコンピュータが、Asset Managementアプリケーションへアクセスできない場合、Asset Managementコネクタは使用不可能です。

任意管理項目のテーブル

Asset Managementアプリケーションのデータベースでは、任意管理項目を用いてデータモデルを拡張できます。

Connect-Itにおいて、任意管理項目はAsset Managementコネクタの使用可能なドキュメントタイプ内で、フィールドまたはコレクションの形で表示されます。

例：

例：amAssetドキュメントタイプでは、資産（この場合はコンピュータ）に使用されているDOSのバージョンはfv_DOSVersionフィールドに表示されます。

AssetCenterへのインポートについては、マニュアル『管理』の「インポート」の章を参照してください。

Asset Managementデータベースの必須フィールド

Asset Managementアプリケーションでは、テーブルのフィールドやリンクが必須である場合があります。これはデフォルトで必須に設定されているか、またはAsset Managementアプリケーションの管理者により必須に設定されています。

整合性チェック時に、Asset Managementコネクタの使用可能なドキュメントタイプ内の各構造体は、1つのレコードに当たります。この構造体内の1要素が必須フィールドであるにも関わらず入力されていない場合は、要素の所属する構造体は拒否されます。

例：

資産のテーブル内では[AssetTag]は必須フィールドです。資産のテーブルに対応するドキュメントタイプ内で、このフィールド用の要素が入力されていないと、要素の所属先の構造体は拒否されます。

API

Asset Managementコネクタを適切に使用するには、AssetCenter、AssetCenter Cable and Circuit、InfraCenterの完全インストールを実行するか、またはカスタムインストールで、コンポーネント「API」を選択する必要があります。

ServiceCenterの設定

本節では、シナリオを使用できるようにServiceCenterを設定する方法を説明します。

入力イベントの作成

シナリオを適切に機能させるためには、以下の2UNLファイルのいずれかをServiceCenter内にインポートする必要があります。

- scac.unl
- scacsp2.unl

これらのファイルは、「device」に関連付けられ、ICM接頭文字を含むイベントを更新します。また、「model」と「modelvendor」ファイルに関連付けられた妥当性の規則も変更します。これらのファイルはServiceCenterの標準データキットも整理します。



警告:

これらのファイルを、プロダクションモードで使用するServiceCenterデータベース内へはインポートしないでください。これらのファイルは、ペレグリンシステムズのサポートから入手できます。

入力イベントの同期をとる

ServiceCenter内の入力イベントは同期式または非同期式に処理されています。非同期モードでは、ServiceCenter用スケジュールが、イベントキューに一時的に保存されているイベントを処理する時期を決定します。同期モードでは、イベントはServiceCenterイベントキューに挿入されるとすぐに処理されます。

シナリオのテスト中に、シナリオが機能するかどうかすぐに確認する場合は、同期モードを選択します。

Connect-Itは同期モードで機能するため、非同期モードで処理すると、ServiceCenterでデータが適切に処理されたかどうかを即時に確認することができなくなります。この場合、ドキュメント処理が待機中であることを知らせる警告メッセージが表示されます。

同期モードでは、ドキュメント処理の成功または失敗のメッセージが表示されます。

適用コンテキスト

このシナリオは、Asset ManagementアプリケーションとServiceCenterに付属しているデータキット間で、データの複製（レプリケーション）を実行します。

 **警告:**

このシナリオは、プロダクション用にインストールされたServiceCenterアプリケーションには当てはまりません。このシナリオは単なる用例に過ぎないことを念頭に置いて下さい。

このシナリオは、資産のリポジトリ、製品、サプライヤ、場所、従業員と部署に関するデータのマッピングのみに制限されています。

これらのデータキットは、ServiceCenterとAsset Managementアプリケーション間のデータの複製用に作成されてはいないため、データは非常に相違しています。

実際の企業では、ServiceCenterデータベース内に保存される従業員のリストは、Asset Managementデータベースの従業員のリストに一致するはずですが、

準備段階では、2データベースのデータを調整します。名前やキーの選択などを調和させます。

このシナリオは、まず第1にServiceCenterのデータをAsset Managementアプリケーションへインポートし、またその反対も実行します。

この作業は問題の一部を解決し、またServiceCenterとAsset Managementアプリケーション間で実際に複製（レプリケーション）を実行する際に適用する一部の規則を明らかにします。

以下の3つの規則の適用が可能になり次第、すぐに適用するようにします。

- 各リポジトリは1つのアプリケーションのみに管理されなければならない。
矛盾を回避するために、あるリポジトリのレコードは、常に同じアプリケーション（ServiceCenterまたはAsset Managementアプリケーション）で作成、変更、削除されなければなりません。これによって複製のプロセスは、あるリポジトリのレコードを別のアプリケーション内で更新するだけになります。例えば、資産のリポジトリは、AssetCenterで管理し、ヘルプデスクのチケットはServiceCenterで管理します。リポジトリは、データの内容がより正確で、データの処理方法がより自然であるアプリケーション内で管理されるべきです。リポジトリを1つのアプリケーションのみで管理すると重要な問題が生じる場合は、別のアプリケーションを小規模な更新時に使用します。しかし、新規レコードの作成やテーブルの識別子の変更などには、このアプリケーションを使用してはなりません。
- レコードを作成したアプリケーションの情報は保存されなければならない。
トラッキング情報を保存することにより、複製が適切に実行されたかどうかを確認できるのに加えて、レコードを作成したアプリケーションに応じて特定のマッピングを作成できるようになります。
- Asset Management アプリケーションとServiceCenter間のデータは調整されなければならない。

マッピングを簡略にするには、2アプリケーション内でデータが同じ方法で処理されなければなりません。例えば、Asset Managementアプリケーションで製品にデフォルトで関連付けられるカテゴリの一部は、ServiceCenterの

「model」に関連付けられるカテゴリとは全く違います。マッピングが適切に作成されるには、同じカテゴリが2アプリケーション内で定義されていなければなりません。

データの整合性チェックとデータの複製の相違点

データの整合性チェックとデータの複製との間には、大きな相違点があります。

整合性チェック

整合性チェックとは、別アプリケーションから受信するデータの統合であり、ソースアプリケーションから来るデータの方が、ターゲットアプリケーション内の既存のデータよりも新しいと見なされています。

このプロセスは、以下の質問とその答えに応じて実行されます。

「受信するデータは、ターゲットアプリケーション内に既存するか?」

- 答えが「いいえ」であると、データは挿入されます。
- 答えが「はい」であると、受信するデータの新規情報に応じて、既存のデータが更新されます。

ターゲットアプリケーションが空である場合は、このプロセスは整合性チェックではなく「インポート」になります。

複製（レプリケーション）

複製のプロセスでは、受信するデータは最新のデータであるとは見なされません。

受信するデータと既存データの最終変更日は、最終複製日と比較されます。どのデータが最も新しい情報であるかを決定するために、この比較が実行されます。

受信するデータと既存のデータが両方とも最終複製日以降に変更されていると、複製の矛盾が発生します。

この場合、どちらのデータがより新しいかを知るための確実な方法はありません。複製シナリオの作成者が、複製エンジンの操作方法を規定しなければなりません。

以下の規則が適用されます。

- A：受信データが拒否される
- B：受信データが最新データであると見なされ、選択される
- C：既存データが最新データであると見なされ、選択される

Connect-It内でのデータの複製の管理

ServiceCenterコネクタは、ドキュメントを取り込む際に、ServiceCenterデータベース内に直接レコードを作成しません。取り込まれたドキュメントから、入力イベントが作成され、キュー（event queue entry）内に追加されます。

この結果整合性チェックのプロセスは、複製のプロセスと同様に、入力イベントがServiceCenterサーバに処理される時に内部で管理されます。

ServiceCenterサーバ内の複製の問題を解決するのは、シナリオの実行責任者の役目になります。

Asset Managementコネクタを使用すると、Asset Managementアプリケーションのデータベースと直接相互作用することが可能になり、整合性のプロセスが（複製のプロセスと同様に）Connect-It内で完全に管理されるようになります。

以下の節では次の点について説明します。

- Asset Managementアプリケーションが複製の矛盾を検出する方法
- 矛盾が発生した場合に実行すべき操作
- データの複製を可能にする日付（日時）が比較される方法

複製の矛盾の検出

Asset Managementアプリケーションでは、取り込まれるドキュメントに【**変更日**】フィールド（Connect-It SQL名：dtLastModif）が含まれていると、複製のプロセスが起動します。

このプロセスは、このフィールドを含む全構造体と全コレクションに適用されます。

シナリオ内で、このフィールドはServiceCenterの【sysmodtime】フィールドにマップされなければなりません。

このシナリオでは、ドキュメントがServiceCenterコネクタに生成される場合は、ドキュメントが最終複製日以降に変更されたこと（またはドキュメントが最終複製日に存在しなかったこと）を意味します。この結果次の質問に対する答えが重要になります。

「ドキュメントに含まれるフィールドは、最終複製日以降にAsset Managementアプリケーション内で変更されたか？」

- 変更されていた場合、複製の矛盾が発生します。
- 変更されていなかった場合（Asset Managementアプリケーションのデータベース内に、データがまだ存在していない場合などがこれに当たります）、データは更新または挿入されます。

矛盾が発生した場合に実行すべき操作

複製の矛盾が検出された場合、矛盾は生成されるドキュメント内の構造体またはコレクションのフィールドに関係します。

矛盾に関連するレコードでは、以下の3つの操作が可能です。

- 拒否
この場合、【 dtLastModif 】フィールドに関するフィールドのエラーが発生します。Asset Managementアプリケーションのデータは変更されません。
- フィールドに対する警告
この場合、【 dtLastModif 】フィールドに関するフィールドの警告が発生します。Asset Managementアプリケーションのデータは更新されます。
- 上書き
この場合、Asset Managementアプリケーション内のデータは、ServiceCenterから来るデータにより、警告なしで上書きされます。

Asset Managementコネクタの取り込み用ルールタブで、これらのオプションのうち1つを選択できます。

日付の比較

複製プロセスで日付（日時）の比較を実行するには、日付が正確である必要があります。日付の正確さは以下の要因に左右されます。

- タイムゾーン
- サーバとの時間差

タイムゾーン

GMT +1タイムゾーンに基づいた日付と、GMT -7タイムゾーンに基づいた日付を、直接比較することは不可能です。まず変換作業が必要になります。

Connect-Itでは、GMTタイムゾーンに基づいた日付が、常に格納され使用されます。GMT日付をConnect-Itへ提供するのは、各コネクタの役割です。

また日付は、Connect-Itを使用しているコンピュータでの設定に応じて表示されます。

サーバとの時間差

全サーバの時間が完全に一致するのが理想的です。しかし実際には、一部のサーバの時間は他のサーバより進んだり遅れたりしています。

Connect-Itでは、コネクタが提供する日時のデータは、ある一定の応答時間に応じて調整されます。応答時間は負の時間になり得ます。

この調節以外では、サーバの応答時間はConnect-It内で隠されています。

`scac /sc3ac40 /scac.scn`



このシナリオは、[scac/sc3ac36/scac.scn](#) [p.314]シナリオと同じ機能を果たします。
この場合複製は、ServiceCenterバージョン3.0と、Asset Managementアプリケーションのバージョン4.0間で実行されます。

scac/sc3ac41/scac.scn



このシナリオは、[scac/sc3ac36/scac.scn](#) [p.314]シナリオと同じ機能を果たします。
この場合複製は、ServiceCenterバージョン3.0と、Asset Managementアプリケーションのバージョン4.1間で実行されます。

scac/sc4ac40/scac.scn



このシナリオは、[scac/sc3ac36/scac.scn](#) [p.314]シナリオと同じ機能を果たします。
この場合複製は、ServiceCenterバージョン4.0と、Asset Managementアプリケーションのバージョン4.0間で実行されます。

scac/sc4ac41/scac.scn



このシナリオは、[scac/sc3ac36/scac.scn](#) [p.314]シナリオと同じ機能を果たします。
この場合複製は、ServiceCenterバージョン4.0と、Asset Managementアプリケーションのバージョン4.1間で実行されます。

scauto/scacfg/scacfg.scn



マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
Output Event Type	InputEventTypes
Input Event Type	OutputEventTypes
EventTypes	SCAutoConfiguration

scauto /sc-ac /sca-ac.scn

このシナリオでは、SCAutoリスニングコネクタ用に設定ファイル (scautoconfiguration.xml) を作成できます。このファイルは、ServiceCenterデータベースのイベントの定義を含みます。

accscups /dataprop

このフォルダにまとめられたシナリオは、ServiceCenter4データベースのデータをAssetCenter 4.1データベースへ移行します。接頭数字は、「accscups /dataprop」フォルダのシナリオが、どの順番で実行されるかを意味しています。

accscups /dataprop /01-SCcompany2ACamCompany.scn



このシナリオは、ServiceCenterの「company」ファイルのデータと、Asset Managementアプリケーションの [サプライヤ] テーブルのレコード間の統合を実行します。

accscups /dataprop /02-SClocation2ACamLocation.scn



このシナリオは、ServiceCenterの「location」ファイルのデータと、Asset Managementアプリケーションの [場所] テーブルのレコード間の統合を実行します。

このシナリオを実行した後、Asset Managementアプリケーションの [場所] のテーブルの [関連所在地] フィールドを確認する必要があります。親所在地と関連所在地のリンクが破損している場合は、リンクを手動で作成する必要があります。

accscups /dataprop /03-SCdept2ACamEmplDept.scn



このシナリオは、ServiceCenterの「department」ファイルのデータと、Asset Managementアプリケーションの [部署と従業員] テーブルのレコード間の統合を実行します。

accscups /dataprop /04-SCcontacts2ACamEmplDept.scn



このシナリオは、ServiceCenterの「contacts」ファイルのデータと、Asset Managementアプリケーションの [部署と従業員] テーブルのレコード間の統合を実行します。

acscups/dataprop/05-SCdevtype2ACamNature.scn



このシナリオは、ServiceCenterの「device types」ファイルのデータと、Asset Managementアプリケーションの [属性] テーブルのレコード間の統合を実行します。

acscups/dataprop/06-SCocmlcat2ACamNature.scn



このシナリオは、ServiceCenterの「categories」ファイルのデータと、Asset Managementアプリケーションの [属性] テーブルのレコード間の統合を実行します。

acscups/dataprop/07-SCmodel2ACamModel.scn



このシナリオは、ServiceCenterの「model」ファイルのデータと、Asset Managementアプリケーションの [モデル] テーブルのレコード間の統合を実行します。

acscups/dataprop/08-SCdevtype2ACamModel.scn



このシナリオは、ServiceCenterの「device type」ファイルのデータと、Asset Managementアプリケーションの [モデル] テーブルのレコード間の統合を実行します。

acscups/dataprop/09-SCdevice2AACamPortfolioAndamComputer.scn



このシナリオは、ServiceCenterの「device data」ファイルのデータと、Asset Managementアプリケーションの [ポートフォリオ品目] と [コンピュータ] テーブルのレコード間の統合を実行します。

acscups/dataprop/10-SClaptop2ACamPortfolioAndamComputer.scn



このシナリオは、ServiceCenterの「laptop」ファイルのデータと、Asset Managementアプリケーションの[ポートフォリオ品目]と[コンピュータ]テーブルのレコード間の統合を実行します。

[acscups/dataprop/11-SCworkstation2ACamPortfolioAndamComputer.scn](#)



このシナリオは、ServiceCenterの「workstation」ファイルのデータと、Asset Managementアプリケーションの[ポートフォリオ品目]と[コンピュータ]テーブルのレコード間の統合を実行します。

[acscups/dataprop/12-SCpc2ACamAndamComputer.scn](#)



このシナリオは、ServiceCenterの「pc」ファイルのデータと、Asset Managementアプリケーションの[コンピュータ]テーブルのレコード間の統合を実行します。

[acscups/dataprop/13-SCserver2ACamAndamComputer.scn](#)



このシナリオは、ServiceCenterの「server」ファイルのデータと、Asset Managementアプリケーションの[コンピュータ]テーブルのレコード間の統合を実行します。

[acscups/dataprop/14-SCphone2ACamPortfolioAndamPhone.scn](#)



このシナリオは、ServiceCenterの「server」ファイルのデータと、Asset Managementアプリケーションの[電話機]テーブルのレコード間の統合を実行します。

SMSシナリオ

本節では、SMS 1.xとSMS 2.xコネクタを使用する付属シナリオについて簡潔に説明します。

[sms/sms1ac35/smsac.scn](#)



このシナリオは、SMS 1.xデータベースの情報（IT資産）を、AssetCenter 3.5データベースの [資産] のテーブルへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
MachinesSrc	amAssetDst

sms /sms1ac35 /smsactl.scn



このシナリオは、SMS 1.xデータベースの情報（IT資産）を、AssetCenter 3.5データベースの [資産] のテーブルへ移行します。

このシナリオは、[sms /sms1ac35 /smsac.scn \[p. 324\]](#)シナリオの簡易バージョンです。マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
MachinesSrc	amAssetDst

sms /sms2ac35 /smsac.scn



このシナリオは、SMS 2.xデータベースの情報（IT資産）を、AssetCenter 3.5データベースの [資産] のテーブルへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
System_DATASrc	amAssetDst

sms /sms2ac35 /smsactl.scn



このシナリオは、SMS 2.xデータベースの情報（IT資産）を、AssetCenter 3.5データベースの [資産] のテーブルへ移行します。

このシナリオは、`sms /sms2ac35 /smsac.scn` [p. 325]シナリオの簡易バージョンです。
 マップされるソースドキュメントタイプとターゲットドキュメントタイプは、
 以下の表の通りです。

ソース	ターゲット
System_DATASrc	amAssetDst

sms /sms2ac41 /smsac.scn



このシナリオは、SMSデータベースの情報（IT資産）を、AssetCenter 4.1データベースの [コンピュータ]（amComputer）のテーブルへ移行します。

ソース	ターゲット
System_DATASrc	amComputerDst

Tivoli Inventory Managementシナリオ

本節では、Tivoli Inventory Managementコネクタを使用するシナリオについて簡潔に説明します。

tim /tim3ac35 /timac.scn



このシナリオは、Tivoli Inventory Management 3.1と3.6のデータベースを、AssetCenter 3.6データベースへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア
- 周辺機器

tim /tim4ac36 /timac.scn



このシナリオは、Tivoli Inventory Management 4.0のデータベースを、AssetCenter 3.6データベースへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア

tim /tim4ac41 /timac.scn



このシナリオは、Tivoli Inventory Management 4.0のデータベースを、AssetCenter 4.1データベースへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア

tim /tim4sc4 /timsc.scn



このシナリオは、Tivoli Inventory Management 4.0のデータベースを、ServiceCenter 4.0アプリケーションへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア

tim /tcm42ac41 /tcmac.scn



 **注意:**

バージョン4.2から、Tivoli Inventory Managementの名前はTivoli Configuration Managerへ変更されました。

このシナリオは、Tivoli Configuration Manager 4.2のデータベースを、AssetCenter 4.1データベースへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア

ゲートウェイ3.xシナリオ

本節では、ゲートウェイ3.xコネクタを使用する付属シナリオについて簡潔に説明します。

uagw /gw3ac36 /gwac.scn



このシナリオは、ゲートウェイ3.xxから来るデータを、ピボットドキュメントタイプを経由してAssetCenterアプリケーションへ移行します。

移行されるデータは以下の通りです。

- IT資産内の全コンピュータ
- コンピュータの技術的な情報（任意管理項目）
- コンピュータの場所
- コンピュータが対応する製品
- インストールされたソフトウェア
- コンピュータのユーザ

Webサービスシナリオ

本節では、Webサービスコネクタを使用する付属シナリオについて簡潔に説明します。

ws/wsac41/currency.scn



このシナリオは、Asset ManagementアプリケーションのamCurRateテーブルで既に定義された通貨間の為替レートを更新します。

このシナリオで使用されているWebサービスのWSDLアドレスは、<http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl>です。

Peregrine Desktop Inventoryシナリオ

本節では、Peregrine Desktop Inventoryコネクタを使用する付属シナリオについて簡潔に説明します。

pdi /pdiac35 /pdiac.scn



このシナリオは、Peregrine Desktop Inventoryで取得したデータを使って、AssetCenter 3.5データベースの [コンピュータ] テーブル (amComputer) のコンピュータに関する情報を更新します。

pdi /pdiac36 /pdiac.scn



このシナリオは、Asset Management 3.6データベースの [コンピュータ] テーブル (amComputer) のコンピュータに関する情報を、Peregrine Desktop Inventoryで取得されたデータを用いて更新します。

pdi /pdiac41 /pdiac.scn



このシナリオは、Asset Management 4.1データベースの [コンピュータ] テーブル (amComputer) のコンピュータに関する情報を、Peregrine Desktop Inventoryで取得されたデータを用いて更新します。

pdi /pdiac42 /pdiac.scn



このシナリオは、Asset Management 4.2データベースの [コンピュータ] テーブル (amComputer) のコンピュータに関する情報を、Peregrine Desktop Inventoryで取得されたデータを用いて更新します。

A | AQLクエリ

付録

本章では、AQLでクエリを書く方法について説明します。

はじめに

本節では、AQL言語と、クエリが必要になる状況について説明します。

AQL

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

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言語の利点

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

データベースの種類に無関係

AssetCenterがサポートする様々なデータベースエンジンは、すべて異なるバージョンのSQLを採用しており、互換性がありません。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。

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

```
SELECT FIRST_ROWS IModelId, FullName FROM amModel
```

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

```
SELECT /*+ FIRST_ROWS INDEX_ASC(M1 Model_IModelId) */ M1.IModelId, M1.FullName FROM amModel M1
```

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

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

IBM DB2では、次のコードが生成されます。

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

AssetCenterデータベースへのアクセスを簡略化

AQLは、リンクと結合の管理を簡略化します。そのため、AQLでクエリを記述すると、直接SQLを使う場合に比べ、速くデータベースにアクセスできます。さらに、AQLを使うと任意管理項目へのアクセスが簡単になり、関連付けられているテーブルのフィールドとして直接任意管理項目を使うことができます。AQLによって、特殊フィールドも容易に活用できます。

AQLの特徴（SQLと比較して）

AQLはDDL（Data Definition Language）ステートメントをサポートしません。AQLには、結合、任意管理項目、特殊フィールドの処理を簡単にする拡張機能があります。

警告:

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

AssetCenterのクエリ

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

次の場合にクエリを使うことができます。

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

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

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

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

警告:

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

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 \[p. 350\]](#)」と「[AQLの関数 \[p. 363\]](#)」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。

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

AQL結合について

定義

結合とは、複数のデータテーブルを1つのクエリに統合することです。

AQL結合

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

AQLリンクは次のように表現されます。

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

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

例

次のAQLで記述したクエリは、モデルごとに以下のデータを返します。

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

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

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

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

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

注意:

OracleとDB2以外のシステムでは、結合数は1つに限られています。

Microsoft SQL Server 7とMSSQL 2000では、「amdb.ini」ファイルを変更して、クエリ実行に関連する問題を防ぎます。次の手順に従って、接続の詳細画面で、「amdb.ini」ファイルを変更します。

```
useSQL92Join=1
```

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、ポートフォリオ品目の資産タグごとに、そのユーザおよび責任者の名前を返します。

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

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

主キーが「0」のレコードを使う理由

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

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

例

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

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

生成されたSQLコードでDBMSの外部結合を使った場合は、Sybase SQL Server用に生成されたSQLコードは次のようになります。

```
SELECT a.AssetTag, l.name FROM amPortfolio a, amEmpDept e, amLocation l WHERE a.UserId *= e.EmpDeptId AND e.LocalId *= l.LocalId
```

このコードは複数の外部結合を次々に使うので、Sybase SQL Serverではサポートされません。

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

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

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

結果

- 特に集計関数を使う場合などは、記述したクエリで前述のようなレコードを処理することが重要になります。

例

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

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

- DBMSレベルの外部結合を生成する必要はほとんどありません。

 注意:

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

NULLの使い方

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

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

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

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

```
WHERE <text field> = NULL
```

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

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

未入力の「日付」または「日付+時刻」型のフィールドについては、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

Self

Selfは、適用先テーブル名の表記（特殊）文字列に相当する式です。

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

例

部署と従業員のテーブル名の表記文字列が次のように構成されているとします。

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

AQLクエリは次のようになります。

```
SELECT self FROM amEmpIDept
```

これは、次のクエリに相当します。

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

CurrentUser

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

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

式として使う

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

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

リンクとして使う

CurrentUserは、先ず全テーブルを検索し、次に【部署と従業員】テーブル内にある現在のユーザのレコードに検索を絞るリンク、と見なされます。

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

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

The screenshot shows a configuration window for a message. The 'Message' tab is selected. The 'Recipient' field is populated with the expression '[CurrentUser.Email]'. The 'Subject' field contains '確認メッセージ' (Confirmation Message). The 'Message' field contains the text 'アクションがトリガされました。' (Action triggered).

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:

システムリストデータの値を確認するには、AssetCenter Database Administrator を使用する方法と、データベース構造を記述する「Database.txt」ファイルを参照する方法があります。

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

階層構造のテーブル

すべての階層構造のテーブルには、次のフィールドが含まれています。

- [FullName] フィールド
- [sLvl] フィールド

[FullName] フィールド

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

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

例

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

```
FullName = '/PC118/DD054/CR012/'
```

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

```
FullName = '/東京/府中支社/府中ビル/5階/'
```

[sLvl] フィールド

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

ルートはレベル0になります。



次のクエリは、「アジア地域営業部」のレコードとその付属部署を選択します。

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

AQLの簡単な表記法

ここでは、AQLの記述を簡略化するための表記法を紹介します。

外部キー

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

例えば次のクエリ

```
WHERE location = 0
```

は次のクエリと同等に処理されます。

```
WHERE ILocald = 0
```

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

表記文字列

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

例

部署と従業員のテーブル名の表記文字列が次のように構成されているとします。

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

AQLクエリは次のようになります。

```
SELECT user FROM amPortfolio
```

これは、次のクエリに相当します。

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

これ自体は次と同等です。

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

任意管理項目

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

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

```
SELECT fv_WorkUnit FROM amEmplDept
```

特殊フィールド

AQLでは、テーブルに関連付けられている特殊フィールドを活用できます。

特殊フィールドのSQL名の前に「cf_」を付けて記述してください。

並べ替えとインデックス

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

- クエリで指定したインデックスをAssetCenterでそのまま使うモード。このモードでは、検索結果がそのまま順次表示されます。
- クエリで指定したインデックスをAssetCenterで使わないモード。この場合は、DBMSがデータの保存方法を決定します。

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 IModelId, Brand FROM amModel ORDER BY Brand
```

- **インデックスを使用しないアクセス**：データベースエンジンは、クエリで指定した【Brand】（メーカー）のインデックスを使わずに全テーブルを走査します。クエリの条件を満たすすべてのデータ項目を検索してから「メーカー」を基準にして並べ替え、ユーザに送信します。結果は一定の時間が経過しないと表示されません。
- **インデックスを使用するアクセス**：データベースエンジンは、【Brand】のインデックスを使って検出した結果をそのままの順番で即時表示します。このため、最初のデータ項目はすぐに表示されますが、全体を処理する時間は長くなります。

インデックスを使う方法

インデックスを使う方法は、クエリを作成する方法によって変わります。

[リストの設定] メニューを使う

AssetCenterのリストごとにデータのアクセスタイプを設定できます。メインリストおよびタブページ内のリストで設定できます。手順は次の通りです。

- 1 設定するリストに移動します。
- 2 右クリックします。
- 3 ポップアップメニューから [リストの設定] を選択します。

- 4 **【列/並べ替え】**タブページで**【インデックスの強制使用】**チェックボックスをオンにし、クエリで指定されたインデックスを使って実行したクエリ結果を、そのままの順番で即時に表示するよう指定します。別のアクセスタイプを選択する場合は、このチェックボックスをオフにします。

AQLを使う

AQLで直接クエリを記述する場合は、FIRST_ROWS句でインデックスを使うことを指定できます。

例

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

 **注意:**

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

並べ替え順

並べ替え順は次の条件に左右されます。

- データベースエンジン
- インデックス使用の有無

Oracle

インデックスを使う場合

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

インデックスを使わない場合

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

例

並べ替え

元のリスト	A B C D a b NULL NULL
インデックスを使ったリスト	A B C D a b
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、クエリの実設計と結果をSQL言語でプレビューすることができます。クエリエディタは、特にデータベース管理者やクエリに精通したパワーユーザを対象としています。

動作

クエリエディタでは、次のいずれかの方法でクエリを設計できます。

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

グラフィカルユーザインタフェースを使う場合も直接AQLで記述する場合も（多くの場合、2つの方法を組み合わせて使います）、作成したクエリをSQL言語で

リアルタイムで表示することができます。ただし、クエリを直接SQLで記述することはできません。

図 A.1. クエリエディタの活用法



パワーユーザや管理者は、クエリエディタを使ってAQLクエリの作成、変更、削除を実行できます。作成者や他のユーザは、作成されたクエリを状況に応じて使います。

クエリエディタにアクセスする

クエリエディタには次の方法でアクセスできます。

- **【ツール/クエリ】**メニューを使う。このメニューを使ってクエリを作成すると、他のユーザもそのクエリを自由に使用できるようになります。クエリは、次の方法で実行できます。
 - **【ツール/クエリ】**メニューを選択すると表示されるクエリの詳細画面で直接実行
 - クエリのメインテーブルに表示される専用メニューから、クエリフィルタを選択して実行
- アクセス制限、クエリフィルタ、リストの設定など、AssetCenterのクエリを呼び出す機能を使う。
- AssetCenterのExportモジュールなどの外部プログラムを使う。

次のクエリエディタでは、内容に応じてクエリを簡略化できます。

例：次のようなクエリがあるとします。

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

簡易版のクエリエディタ（シンプルフィルタ、クエリフィルタなど）では、クエリのWHERE句だけを定義するだけで済みます。クエリのその他のコンポーネント（クエリを実行するテーブル、フィールドなど）は暗黙的に指定します。例えば、クエリフィルタの場合、テーブルはフィルタを適用するテーブルであり、フィールドと並べ替え条件は**【リストの設定】**ポップアップメニューで定

義した列 / 並べ替えの条件です。[**ツール / クエリ**] メニューからクエリエディタにアクセスした場合も同様です。

例えば、次のような明示的に記述したクエリがあるとします。

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

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

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

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

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

クエリエディタを使ってクエリを作成する

クエリエディタを使ってクエリを作成するには、[**ツール / クエリ**] メニューを選択します。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 [作成] をクリックすると、クエリの詳細を指定するタブページを使うことができます。

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

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

複数のフィルタ条件を定義できます。


フィルタ条件を定義するには、次の手順に従います。

- 1 [フィールド1] に開始テーブルのフィールド、定数、または式を指定します (比較演算子を使う場合は [フィールド2] フィールドにも同様に指定します)。
- 2  ボタンを使ってこのクエリを画面下部のウィンドウに転送し、フィルタ条件を確認します。
- 3 [変更] をクリックし、クエリの作成を確定します。

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

- 1 前述のように、最初のフィルタ条件を作成します。
- 2 その他のフィルタ条件を定義し、[AND] ボタンまたは [OR] ボタンで条件を連結します。
- 3 [変更] をクリックし、クエリの作成を確定します。

注意:


選択した条件を変更するには、 ボタンをクリックしてウィンドウの内容を削除するか、またはAQLコードを直接変更します。

注意:

グラフィックツールを使わずに、[フィルタ (WHERE句)] タブページの下部のウィンドウに直接AQLでクエリを入力することもできます。

手順3：クエリの実行をプレビューする

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

- 1 クエリの詳細画面の [プレビュー] タブページに移動します。
- 2  アイコンをクリックします。クエリによって検索されたレコードが下のウィンドウに一覧表示されます。クエリ条件に一致するレコード数は、ウィンドウの右下に表示されます。

注意:


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

クエリで使うフィールド

クエリのフィルタ条件を定義する場合、次のフィールドを指定できます。


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

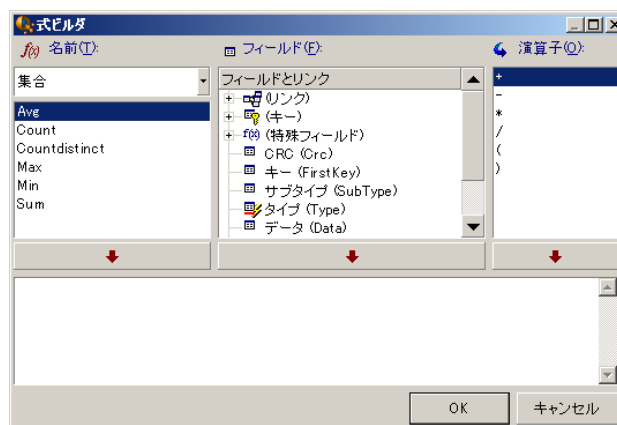
式を記述する

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


式を記述するには、次の方法があります。

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


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



式ビルダは、次の3つの列で構成されます。

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

【名前】、**【フィールド】**、**【演算子】**を式に挿入するには、


- 1 関数、フィールド、演算子のいずれかを選択します。
- 2  をクリックします。

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


定数

定数 x は、選択基準に割り当てる固定の値です。例えば、メーカーが「3Com」のすべてのモデルを検索する場合は、モデルのテーブルの**【Brand.Name】**リンクフィールドに定数値「3Com」を割り当てます。

定数を選択するには、次の手順に従います。

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

注意:

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

AQL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 A.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。
その他	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 [p. 357]]
 [WHERE句 [p. 358]]
 [GROUP BY句 [p. 359]]
 [HAVING句 [p. 359]]
 [ORDER BY句 [p. 360]]

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 [p. 357]]
 [WHERE句 [p. 358]]
 [GROUP BY句 [p. 359]]
 [HAVING句 [p. 359]])

)

警告:

サブクエリは括弧で囲む必要があります。

使用例

```
SELECT Self FROM amAsset WHERE mPrice >= (SELECT Max(mPrice)/2 FROM a
mAsset)
```

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 [p. 357]]

[WHERE句 [p. 358]]

[GROUP BY句 [p. 359]]

[HAVING句 [p. 359]]

[UNION | UNION ALL | INTERSECTS | MINUS]

SELECT <選択リスト>

[FROM句 [p. 357]]

[WHERE句 [p. 358]]

[WHERE句 [p. 358]]

[GROUP BY句 [p. 359]]

[HAVING句 [p. 359]]...

[ORDER BY句 [p. 360]]

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.]]<フィールド>

ポートフォリオ品目 (SQL名 : AmPortfolio) テーブルの例

Model
User.Name
User.Location.Name

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


```
<table.[link...]field>
```

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


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

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

後半の2つの表記は、FROM句が使えない場合に便利です。

例えば、AssetCenterでクエリを記述する場合は、WHERE句しか使えません。クエリの開始テーブルは暗黙的に指定します（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#
#2004-01-01 01:00:03#
```

式

式は、次の要素で構成されます。

- 定数
- フィールド
- 関数
- サブクエリ

これらの要素に演算子や括弧を組み合わせると、複雑な式を作成できます。

比較の式のシンタックスは次の通りです。

<式> <比較演算子> <式>

論理式のシンタックスは次の通りです。

<比較式> <AND | OR> <比較式>

括弧を使って複数の論理式を組み合わせることができます。

演算子

論理演算子

論理演算子を使うと、2つの式を連結できます。

表 A.2. 論理演算子

演算子	説明
AND	論理積演算
OR	論理和演算

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

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

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

```
seAssignment >=3
```

比較演算子

比較演算子は、2つの式を比較する時に使います。

表 A.3. 比較演算子

演算子	説明
=	等しい
<>	等しくない
=!	
>	より大きい
<	より小さい
>=	以上
=<	以下
=*	右外部結合。この演算子は、AQLでリンクを処理する場合のみ使います。
*=	左外部結合。この演算子は、AQLでリンクを処理する場合のみ使います。

演算子	説明
LIKE	<p>「=」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。</p> <p>次のワイルドカード文字を使用できます。</p> <p>パーセント (%) : すべての文字列に置き換わります。</p> <p>アンダースコア (_) : 任意の1文字に置き換わります。</p> <p>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)。</p> <p>[abc...]は任意の並び文字 (間にスペースがない) を定義します。</p> <p>[a-c]は、文字列値の範囲を定義します。</p> <p>DB2では、LIKE X演算子のXにSQL列名が含まれる場合は、この演算子を使うことはできません。この演算子で使用できるのは定数のみです。例えば、次のようなクエリはDB2で機能しません。</p> <p>SELECT COL1, COL2 FROM TABLE1 WHERE COL1 LIKE COL2</p>
NOT LIKE	
IS NULL	フィールド値がNULLかどうかを調べます。
IS NOT NULL	AssetCenterでは、空のテキストフィールドと、値が入力されていない日付または日付+時刻型フィールドがNULLと見なされます。

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ポートフォリオ品目のリストを取得できます。

```
SELECT IModelId, Model.Brand FROM amPortfolio WHERE Model.Brand = ANY (SELECT Model.Brand FROM amPortfolio WHERE 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ステートメントを指定します。

選択リストは、次のようにカンマで区切られた式で構成されます。

<式> [, <式> ...]

各式をエイリアスにリンクできます。例は以下の通りです。

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

選択リストは、出力クエリで出力する列名を指定する場合に特に便利です。

 **注意:**

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

FROM句

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

AQLでは、テーブル名のエイリアスを利用できます。

シンタックス

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

クエリの開始テーブル

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

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

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

```
SELECT AssetTag FROM amAsset
```

クエリ内のテーブル数

クエリで指定できるテーブル数は、使用しているDBMSによって異なります。

例

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

警告:

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

例

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

次のクエリはすべて同じように機能します。

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

WHERE句

AQLのWHERE句は、SQLのWHERE句と同様に機能します。

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

シンタックス

WHERE <検索条件>

検索条件の作成

多くの場合は、次の書式で条件を記述する必要があります。

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

場合によっては、次のように複雑なクエリを記述する必要があります。

```
<WHERE | HAVING> [NOT] EXISTS (<サブクエリ>)
<WHERE | HAVING> [NOT] <式> [NOT] IN (<値のリスト> | <サブクエリ>)
<WHERE | HAVING> [NOT] <式> <比較演算子> <ANY | ALL> (<サブクエリ>)
```

GROUP BY句

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

シンタックス

GROUP BY <集合を使わない式> [, <集合を使わない式>]...

注意事項

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

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

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

GROUP BY - 例

次のクエリは、データベース内のメーカーの総数を算出します。このクエリの場合、AssetCenterは、1つのメーカーに関連付けられている資産ごとに、1つのメーカーインスタンスを返します。

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

次のようにGROUP BY句を使うと、メーカーのリストと各メーカーの資産数が算出されます。

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

HAVING句

AQLのHAVING句は、SQLのHAVING句と同等に機能します。

シンタックス

HAVING <検索条件>

WHERE句との相違点

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

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

例

次の例に示すクエリでは、WHERE句とHAVING句が同等に機能します。

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

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

次のWHERE句を使ったクエリでも同じ結果が得られます。

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

HAVING句を使ったクエリの例

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

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

ORDER BY句

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

項目の並べ替え順には、次のタイプがあります。

- 昇順：ASC。デフォルトの並べ替え順です。
- 降順：DESC

シンタックス

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

INSERT句

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

シンタックス

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

この句はAssetCenter API AmDbExecAqlに含まれています。

AssetCenter APIの詳細については、マニュアル『プログラマーズリファレンス』の「関数の説明」の章を参照してください。

例

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

INSERT句を使用しないウィザードのコード

```
hrAlarm = AmCreateRecord("amDateAlarm")
  Err = AmSetFieldLongValue(hrAlarm, "bSecondLevel", 0)
  Err = AmSetFieldLongValue(hrAlarm, "dtTrig1", AmGetFieldLongValue(hrAsset, 2)-IDaysBefore*86400)
  Err = AmSetFieldLongValue(hrAlarm, "IAction1Id", IActionId)
  Err = AmSetFieldLongValue(hrAlarm, "IMonitObjId", IAstId)
  Err = AmSetFieldStringValue(hrAlarm, "MonitoredField", "dWarrEnd")
  Err = AmSetFieldStringValue(hrAlarm, "MonitoredTable", "amAsset")
  Err = AmSetFieldLongValue(hrAlarm, "sDaysBefore1", IDaysBefore)
  Err = AmInsertRecord(hrAlarm)
```

INSERT句を使用したウィザードのコード

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

UPDATE句

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

シンタックス

UPDATE <テーブル名> [テーブルのエイリアス] SET (<フィールド名> [, <フィールド名>...] [FROM句 [p. 357]] [WHERE句 [p. 358]]

例

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

UPDATE句を使用しないアクションのコード

```
hr = AmGetRecordFromMainId("amPOrder", [IPOrdId])
|Err = AmSetFieldLongValue(hr, "seStatus", "${IDS_POSTATUS_ORDERED}")
|Err = AmUpdateRecord(hr)
```

UPDATE句を使用したアクションのコード

```
|Err = AmDbExecAql("update amPOrder set seStatus = 21 where IPOrdId = " &
|IPOrdId)
```

DUPLICATE句

DUPLICATEは、データベースのテーブルのレコードを複製します。

この関数はAssetCenter特有です。

詳細については、マニュアル『はじめに』の「Connect-Itを初めて使用する」の章の「レコードの処理」/「レコードを複製する」を参照してください。

シンタックス

DUPLICATE <テーブル名> [テーブルのエイリアス] SET (<フィールド名> [, <フィールド名>...] [FROM句 [p. 357]] [WHERE句 [p. 358]]

DELETE句

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

シンタックス

DELETE [FROM句 [p. 357]] [WHERE句 [p. 358]]

AQLの関数

次のAQL関数は、クエリと計算式で使うことができます。

- 集合型のAQL関数
- 文字列型のAQL関数
- 日付型のAQL関数
- 数値型のAQL関数
- テスト型のAQL関数

注意:

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

集合型のAQL関数

表 A.4. 集合型のAQL関数

機能	説明
Avg(<列>)	数値型の列項目の平均を返します。列にレコードがない場合は「0」を返します。
Count(<列>)	列内の非NULL値をカウントします。
Countdistinct(<列>)	列内の異なる非NULL値をカウントします。
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 A.5. 文字列型のAQL関数

機能	説明
Ascii(<文字列>)	<文字列>の先頭の文字のASCIIの値を返します。
Char(<n>)	ASCIIコード「n」の文字を返します。
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)。
Upper(<文字列>)	<文字列>を大文字で返します。

日付型のAQL関数

表 A.6. 日付型のAQL関数

機能	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)。

機能	説明
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）
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 A.7. 日付型のAQL関数

説明	AssetCenterのクエリ言語
先週変更したすべてのレコード	AddDays(dtLastModif,7)>=Getdate()

説明	AssetCenterのクエリ言語
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 A.8. 数値型のAQL関数

機能	説明
Abs(<数値>)	「数値」の絶対値を返します。
Ceil(<数値>)	「数値」以上で一番小さい整数を返します。
Floor(<数値>)	「数値」以下で一番大きい整数を返します。
Mod(<a>,)	「a」を「b」で割った余りを返します(a = qb + r、ただしqは整数で 0 =< 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 A.9. テスト型のAQL関数

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

クエリの例

本節では、各例につきクエリ作成の1側面が説明されています。以下の例を参考にして独自のクエリを作成してください。

ここで紹介する例では、完全なシンタックスでクエリを記述しています。これらのクエリを実際にテストする場合は、AssetCenter Exportを使います。これらのクエリをデータベースのクエリフィルタなどで使用するには、シンタックスを変更する必要があります。

例えば、次のような明示的に記述したクエリがあるとします。

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

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

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

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

注意:

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

メインテーブルのフィールドを特定の値と比較する

例：メーカーが「Compaq」であるすべてのポートフォリオ品目

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

メインテーブルのリンクを別のリンクと比較する

例：親資産と場所が同じであるすべてのポートフォリオ品目

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

メインテーブルのリンクを特定の値と比較する

例：「大阪支社」に直接リンクしているすべての部署と従業員

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

メインテーブルにリンクしたテーブルのフィールドの値に応じて比較する

例：親資産と場所名が同じであるすべてのポートフォリオ品目

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

階層構造のテーブル

[FullName] フィールドを使う

例：名前が「大阪支社」である場所の、すべての関連場所

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

[FullName] フィールドと [sLvl] フィールドを使う

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

例：「大阪支社」場所に関連する従属場所で、階層のレベルが3より低いレコード

ツリー構造のルートの階層レベルは「0」です。

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/大阪支社/%') AND (sLevel < 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「府中支社」に位置するすべての従業員

```
SELECT Self FROM amEmpIDept WHERE (Title = '会計部長') AND (Location.Name = '府中支社')
```

数値、日付、テキスト型フィールドの比較

例：2003年1月1日から2003年12月31日までに実行したすべての作業指示

```
SELECT self FROM amWorkOrder WHERE (dtActualFixStart >= #2003-01-01 00:00:00#) AND (dtActualFixStart <= #2003-12-31 00:00:00#)
```

任意管理項目に関するクエリ

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

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

式でレコードを検索する

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

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

未入力のフィールドを検索する

例：電話番号のないすべての従業員。空の文字列を一重引用符で囲みます。

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

別の組み合わせの例

「コンピュータ」属性の従属モデルがないすべてのモデル

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

 注意:

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

エイリアスを使ったクエリ

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

開始テーブル：[部署と従業員] テーブル

クエリは次の通りです。

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

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

以下のように記述すると、

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

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

以下のように記述すると、

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

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

索引

目次

- キャッシュ
 - 設定, 35
- コネクタ
 - Action Request System, 53
 - Asset Management, 60
 - InfraTools Desktop Discovery, 67
 - InfraTools Management, 78
 - InfraTools Network Discovery, 82
 - Peregrine Desktop Inventory, 88
 - Peregrine Systems, 53
 - ServiceCenter, 98
 - インベントリ, 25
 - ゲートウェイ3.x, 92
 - コネクタに名前を付け、役割を説明する, 14
 - ルール (ディレクティブ), 39
 - 取り込み用ルール (ディレクティブ), 43
 - 生成用ルール, 40
 - 接続タイプの選択, 15
 - 設定, 14, 13
- コミット, 34
- サーバ
 - サーバとの時間差を指定する, 29
- スケジュールのポインタ, 32
- 使用, 32
- セッション, 28
- データベース記述ファイル, 24
- ドキュメント
 - ルール (ディレクティブ), 39
 - ドキュメントタイプ
 - 取り込み用ルール (ディレクティブ), 43
 - トランザクション, 34
 - ピボットドキュメント, 36
 - マッピング
 - 取り込み用ルール (ディレクティブ), 43
 - ルール (ディレクティブ), 39
 - WHERE句とORDER BY句, 41
 - 取り込み用ルールの定義, 43
 - 整合性チェック, 44
 - 生成用ルールの定義, 40
 - 外部アプリケーション, 13
 - 高度な設定, 24
 - 再接続, 27
 - 自動的, 28
 - 識別キー, 45
 - キーセット, 48
 - 処理後のアクション, 191, 91
 - 詳細オプション, 25
 - 整合性, 44
 - 整合性チェック, 50
 - キーセット, 48
 - コレクション, 50
 - 整合性チェックのタイプ, 44
 - 大文字小文字, 49
 - 動作, 49
 - 接続, 15
 - DB2, 23
 - MySQL, 21
 - ODBC, 17

- Oracle, 18
- Sybase, 20
- 選択, 16
- 接続テスト, 28
- 設定ウィザード, 14
 - サーバとの時間差を指定する, 29
 - 再接続用パラメータ, 27
 - 接続タイプの選択, 15
- 大文字小文字, 49
- 配列の取り出し, 42
- 複製 (レプリケーション)
 - 矛盾, 52

B

- Blob, 42

D

- DSC, 201
- DSE, 174

F

- FSF, 68

O

- ORDER BY, 41

S

- SQL92, 25

W

- WHERE, 41

