

HP Connect-It Software

ソフトウェアバージョン : 3.80

コネクタ



法的制限事項

保証

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

本規定のいかなる部分も、他の保証を構成すると解釈されるものではありません。

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

本書に含まれる内容は、予告なく変更される場合があります。

限定保証条項

機密コンピュータソフトウェア。

所有、使用、または複製するには、HPからの有効なライセンスが必要です。

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

Copyrights

© Copyright 1994-2007 Hewlett-Packard Development Company, L.P.

商標

- Adobe®, Adobe Photoshop® and Acrobat® are trademarks of Adobe Systems Incorporated.
- Corel® and Corel logo® are trademarks or registered trademarks of Corel Corporation or Corel Corporation Limited.
- Java™ is a US trademark of Sun Microsystems, Inc.
- Linux is a U.S. registered trademark of Linus Torvalds
- Microsoft®, Windows®, Windows NT® and Windows® XP are U.S. registered trademarks of Microsoft Corporation.
- Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.
- UNIX® is a registered trademark of The Open Group.

目次

はじめに	13
コネクタの使用目的	13
コネクタの対象ユーザ	13
本書の使用方法	14
コネクタの互換性	14
1. コネクタの設定	15
コネクタの設定	15
Name and describe the connector	16
Select a connection type	18
Configuring the connection	19
Select files or folders	25
Advanced configuration (詳細モード)	27
Configure the reconnection parameters (詳細モード)	29
Determine server delay (詳細モード)	31
Configure schedule pointers (詳細モード)	33
Manage transactions (詳細モード)	35
Configure the cache (詳細モード)	36
Advanced options (詳細モード)	37
Define document types (詳細モード)	39
Configure the JVM	40
Parallelize consumption (詳細モード)	41

2. コネクタのルール（ディレクティブ）	43
生成用ルール	44
取り込み用ルール	46
照合更新	47
3. Hewlett-Packardコネクタ	59
Asset Managementコネクタ	59
Desktop Administration 6.xコネクタ	71
Enterprise Discoveryコネクタ	76
OpenView Desktop Inventory (PDI) コネクタ	79
OpenView Network Inventory 8.xコネクタ (PDI)	84
OpenView Network Inventoryコネクタ	89
OpenView Operationsコネクタ	91
BizDocコネクタ	95
SCAutoリスニングコネクタ	99
Decision Centerコネクタ	104
ServiceCenterコネクタ	107
ServiceCenter Webサービスコネクタ	127
OpenView Service Desk (送信)	130
OpenView Service Desk (受信)	133
OpenView Inventory Manager 4.2iシナリオ	137
OpenView Service Events v4.2iコネクタ	139
OpenView Service Events v4.2iコネクタ	141
Management Portalコネクタ	143
4. アプリケーションコネクタ	147
Action Request Systemコネクタ	147
Lotus Notesコネクタ	154
MQSeriesコネクタ	163
NTセキュリティコネクタ	185
Mercuryコネクタ	189
Tivoli Enterprise Consoleコネクタ (送信)	189
Tivoli Enterprise Consoleコネクタ (受信)	194
5. プロトコルコネクタ	205
データベースコネクタ	205
XMLリスニングコネクタ	214
Eメールコネクタ	225
LDAPコネクタ	239
コマンドラインコネクタ	252
区切りテキストコネクタ	256

JMSコネクタ	280
Webサービスコネクタ	284
XMLコネクタ	289

6. インベントリコネクタ 315

Altirisコネクタ	315
Asset Insightコネクタ	318
LANDesk for Inventory 8コネクタ	323
LANDesk Software Distributionコネクタ	326
System Management System (SMS) コネクタ	328
iInventoryコネクタ	332
Tivoli Inventoryコネクタ (バージョン4.0)	336
Tivoli CM for Software Distribution 4.2コネクタ	339
Tivoli CM Inventory 4.2コネクタ	339
Tivoli CM for Software Distribution Status 4.2コネクタ	344
TS.Census 2コネクタ	344
TS.Census 3コネクタ	348
CA Unicenter AMO 3コネクタ	351
CA Unicenter AMO 4コネクタ	354
Winpark Actimaコネクタ	357

7. ERPコネクタ 361

SAPコネクタ	361
SAP BAPIコネクタ	363
SAP IDocコネクタ	366
SAP ALEコネクタ	369

8. 付属シナリオ 373

Altirisシナリオ	374
Action Request Systemシナリオ	374
Asset Insightシナリオ	375
Asset Managementシナリオ	376
Eメールシナリオ	377
Enterprise Discoveryシナリオ	379
iInventoryシナリオ	385
LANDeskシナリオ	386
OpenView Configuration Managementシナリオ	388
Mercuryシナリオ	395
MQSeriesシナリオ	396
LDAPシナリオ	398
Network Node Managerシナリオ	399
NTセキュリティシナリオ	403

OpenView Network Discoveryシナリオ	405
OpenView Desktop Inventoryシナリオ	406
Decision Centerシナリオ	409
SAP BAPIシナリオ	409
SAP BAPIシナリオ	412
SMSシナリオ	414
ServiceCenterシナリオ	416
ServiceCenter Webサービスシナリオ	425
TS Censusシナリオ	427
Tivoliシナリオ	429
Tivoli Enterprise Data Warehouseシナリオ	432
Unicenter AMOシナリオ	432
Webサービスシナリオ	433
Winpark Actimaシナリオ	433

A. AQLクエリ 435

はじめに	435
AQLクエリの記述に関する推奨事項	438
並べ替えとインデックス	446
クエリエディタ	448
AQLのシンタックス	453
AQLの関数	465

インデックス 469

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	44
2.2. 取り込み用ルール - Reconciliation タブ	48
2.3. 取り込み用ルール - Advanced reconciliation タブ	53
2.4. 照合更新 - コレクションのフィルタ	55
2.5. 照合更新 - ターゲットコレクションの構成要素の削除	55
2.6. 取り込み用ルール - Reconciliation scripts タブ	56
3.1. Asset Manager コネクタ - 発行ドキュメントタイプ	60
3.2. PDI 8 - AM マッピング	69
3.3. PDI コネクタ - 生成用ドキュメントタイプ	83
3.4. PDI コネクタ - 生成用ドキュメントタイプ	88
3.5. ServiceCenter Web サービスコネクタ - 取り込み - 生成されたドキュメントタイプ	128
4.1. MQSeries コネクタ - 相関IDによるMQSeries メッセージの識別	182
4.2. NT セキュリティコネクタ - 発行ドキュメントタイプ	188
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	235
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	236
5.3. 区切りテキストコネクタ - テキストファイル内のデータの記録法	256
5.4. 区切りテキストコネクタ - データ処理	273
5.5. 区切りテキストコネクタ - 取り込み用ルール	279
5.6. Web サービスコネクタ - 一般的な原理	285
5.7. XML コネクタ - の取り込み用ルール	311
6.1. Asset Insight コネクタ - 生成ルールの例	321
6.2. Asset Insight コネクタ - AI_CONFIG_Tドキュメントタイプ	322

6.3. LANDesk for Inventory 8コネクタ - COMPUTER_SYSTEMドキュメントタイプ	325
6.4. SMSコネクタ - 生成用ルールの例	332
6.5. Tivoli CM - Inventory 4.2コネクタ - COMPUTER_SYSTEMドキュメントタイプ	342
6.6. Tivoli CM Inventory 4.2コネクタ - 生成用ルール	343
6.7. TS.Census 2コネクタの生成用ルール	347
6.8. TS.Census 3コネクタの生成用ルール	351
7.1. SAPコネクタ - 取り込み - 生成用ドキュメントタイプ	364
A.1. クエリエディタの活用法 - compositionモード	449

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 - 詳細設定オプション	38
3.1. Asset Managementコネクタの設定	63
3.2. Desktop Administration 6.xコネクタの設定	73
3.3. Enterprise Discoveryコネクタの設定	77
3.4. PDIコネクタの設定	79
3.5. PDIコネクタの設定	84
3.6. OpenView Network Inventoryコネクタの設定	89
3.7. node-->amcomputer	94
3.8. BizDocコネクタ（読み取り）の設定	95
3.9. BizDocコネクタ（書き込み）の設定	97
3.10. SCAutoリスニングコネクタの設定	101
3.11. Decision Centerコネクタの設定	105
3.12. ServiceCenterコネクタの設定	110
3.13. 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	117
3.14. ServiceCenter Webサービスコネクタの設定	128
3.15. OpenView Service Deskコネクタ（送信）の設定	131
3.16. OpenView Service Deskコネクタ（受信）の設定	133
3.17. OpenView Inventory Managerコネクタの設定	138
3.18. OpenView Service Eventsコネクタの設定	139
3.19. OpenView Usage Managerコネクタの設定	142
3.20. Management Portalコネクタの設定	144
4.1. Action Request Systemコネクタの設定	149
4.2. Lotus Notesコネクタの設定	155

4.3. MQSeriesコネクタ（読み取り）の設定	164
4.4. MQSeriesコネクタ（書き込み）の設定	173
4.5. NTセキュリティコネクタの設定	186
4.6. NTセキュリティコネクタ - 生成用ルールの例	188
4.7. Tivoli Enterprise Consoleアダプタ（送信）の設定	190
4.8. TECコネクタの設定	195
5.1. データベースコネクタの設定	211
5.2. XMLリスニングコネクタの設定	216
5.3. Eメールコネクタ（受信）の設定	226
5.4. Eメールコネクタ（送信）の設定	231
5.5. Eメールコネクタ（受信）が生成するドキュメントタイプ	234
5.6. Eメールコネクタ（送信）が取り込むドキュメントタイプ	237
5.7. LDAPコネクタの設定	241
5.8. LDAPフィルタ	248
5.9. マップテーブルファイル（ASCII値の文字）	249
5.10. コマンドラインコネクタの設定	253
5.11. 区切りテキストコネクタ（読み取り）の設定	257
5.12. 区切りテキストコネクタ（書き込み）の設定	265
5.13. サポートされるプロトコル - Webサービスコネクタ	286
5.14. Webサービスコネクタの設定	286
5.15. Webサービスコネクタ - 取り込みドキュメントタイプ	289
5.16. XMLコネクタ（読み取り）の設定	290
5.17. XMLコネクタ（書き込み）の設定	301
6.1. Altirisコネクタの設定	316
6.2. Asset Insightコネクタの設定	319
6.3. LANDesk for Inventory 8コネクタの設定	323
6.4. LANDesk Software Distributionコネクタの設定	327
6.5. SMS 2.xコネクタとSMS 2003コネクタの設定	329
6.6. iInventoryコネクタの設定	333
6.7. Tivoli Inventory Management 4.0コネクタの設定	336
6.8. Tivoli Inventory Management 4.2コネクタの設定	340
6.9. TS.Census 2コネクタの設定	345
6.10. TS.Census 3コネクタの設定	348
6.11. Unicenter AMO 3コネクタの設定	352
6.12. CA Unicenter AMO 4コネクタの設定	355
6.13. Winpark Actimaコネクタの設定	358
7.1. ABAPデータ型とConnect-Itデータ型の対応関係	364
7.2. SAP BAPIコネクタの設定	365
7.3. SAP IDocコネクタの設定	367
A.1. 表記法	454
A.2. 論理演算子	457
A.3. 比較演算子	458
A.4. AQL - 集合型のAQL関数	465
A.5. AQL - 文字列型の関数	466
A.6. AQL - 日付型関数	466

A.7. AQL - 日付型関数の例	467
A.8. AQL - 数値型の関数	468
A.9. AQL - テスト型関数	468

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Scenario Builder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Scenario Builder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Scenario Builderには付属していません。Hewlett-Packard Development Company, L.P.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

コネクタの対象は、企業内で現在使用されている多様なアプリケーションをHewlett-Packardアプリケーションで統合するユーザです。

本書の使用方法

コネクタの章

以下の章

- Hewlett-Packardコネクタ [献 59]
- アプリケーションコネクタ [献 147]
- プロトコルコネクタ [献 205]
- インベントリコネクタ [献 315]
- ERPコネクタ [献 361]

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、Scenario Builder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。

コネクタの互換性

本マニュアルには、コネクタがサポートするソフトウェアのバージョン番号に関する記載がありません。

互換性に関する情報については、HP-Peregrineソフトウェアサポート (www.hp.com/managementsoftware/peregrine_support/) にある互換対応表をご確認ください。

このサイトにアクセスするには、有効なログインとパスワードが必要になります。

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。

- Name and describe the connector
- 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
- 発行用、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多量のオプションを指定する


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、本章の節ですべてのコネクタに共通な設定ウィザードのページをまとめます。「[Hewlett-Packardコネクタ \[献 59\]](#)」、「[アプリケーションコネクタ \[献 147\]](#)」、「[プロトコルコネクタ \[献 205\]](#)」、「[インベントリコネクタ \[献 315\]](#)」と「[??? \[献 ?\]](#)」の各章では、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。一例を挙げると、「[Hewlett-Packardコネクタ \[献 59\]](#)」章の「[Asset Managementコネクタ \[献 59\]](#)」節、「[Asset Managementコネクタの設定 \[献 63\]](#)」節などです。

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。

コネクタの設定

Connect-Itコネクタの設定はウィザードで実行できます。

connector's configuration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*Scenario diagram*内に位置しない場合
 - 1 Toolbox内のコネクタをダブルクリックします。
 - 2 Toolboxから*Scenario diagram*へコネクタをドラッグします。
- コネクタが*Scenario diagram*内に位置する場合
 - 1 コネクタを選択してから**Tools/Configure**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**Configure connector**を選択します。
 - 4  をクリックします。

詳細モード

設定ウィザードに表示されるページ数は、このオプションによって変化します。

このオプションを有効にするには、 アイコンをクリックしてください。

以下のページが詳細モードで表示されます。

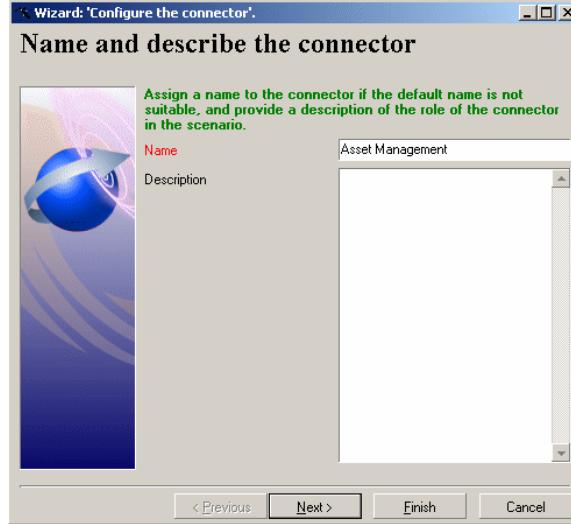
- Advanced configuration
- Configure the reconnection parameters
- Determine server delay
- Configure schedule pointers
- Manage transactions
- Configure the cache
- Define document types
- Parallelize consumption

Name and describe the connector

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



Name

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、Toolbox内のコネクタ名です。同種のコネクタがScenario diagram内に既にある場合は、Connect-Itがコネクタのデフォルト名に数字を加え、*NameOfConnector*、*NameOfConnector1*、*NameOfConnector2*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、Scenario diagram内のコネクタ上に表示されます。

警告:

1シナリオで同一コネクタを複数回使用する場合、それぞれに異なる名前を割り当てる必要があります。たとえば、2つのAssetCenterデータベース間でデータを転送するシナリオでは、各Asset Managementコネクタに異なる名前を割り当てます。

Description

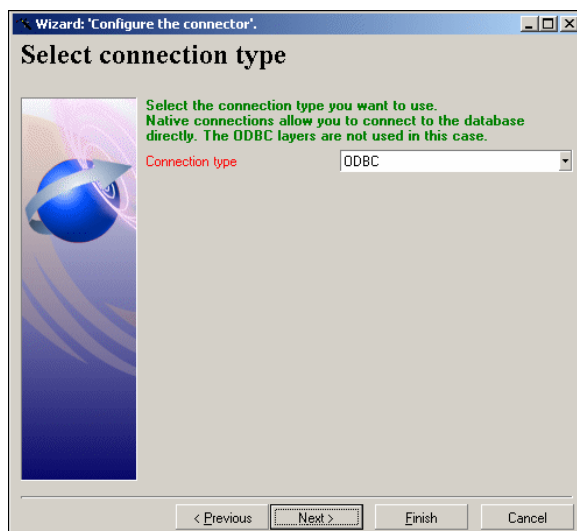
コネクタを説明するテキストを入力します（例：データソースAの読み取り、データソースAへの書き込み、など）。

これは必須フィールドではありません。

Select a connection type

このページでは、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:

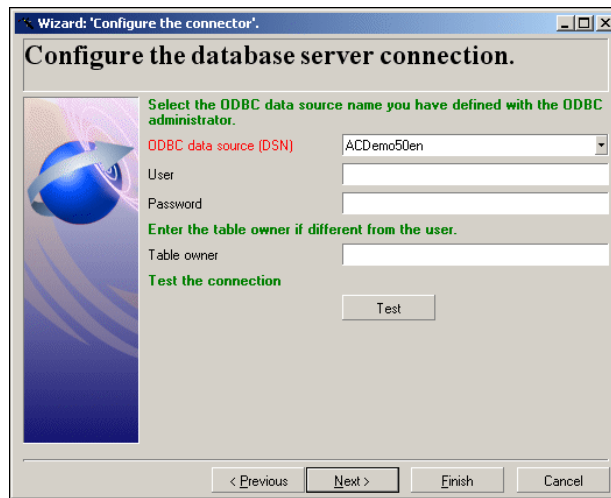
MySQLネイティブ接続を使用するには、MySQLのクライアントのダイナミックライブラリ（DLL）を、Connect-Itインストール先フォルダのbinフォルダ内（またはオペレーティングシステムのsystem32フォルダ内）にコピーする必要があります。

Configuring the connection

このページでは接続を設定できます。

前のページ [Select a connection type [献 18]] 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
- ユーザ
- パスワード
- テーブルの所有者

ODBC data source

ドロップダウンリストで値を選択して、ODBC接続の名前を指定します。ドロップダウンリストには、コンピュータ上で利用できるODBC接続が含まれます。このリストは変更不可能であるため、コネクタを作成する前にODBCデータソースをODBC Administratorで設定しておく必要があります。

User

ODBCデータソースにアクセスするユーザ名を入力します。

Password

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。

Table owner

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。

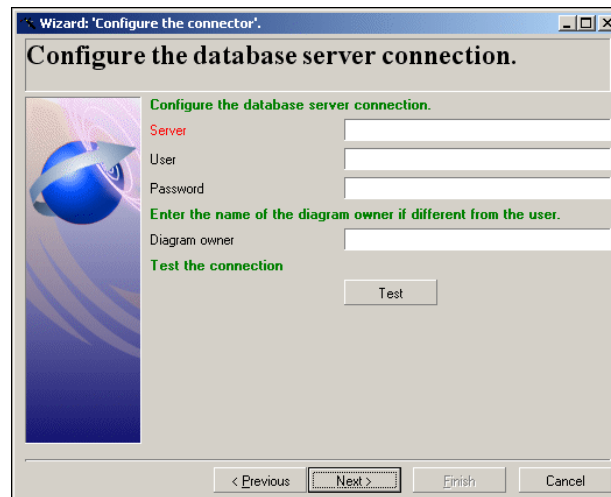
Test

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。
- **Test**をクリックします。
Test the connection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。
- **Close**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接続



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
- ユーザ
- パスワード
- テーブルの所有者

Data server

Oracleサーバの名前を指定します。

User

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。

Password

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。

Table owner

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。

Test

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。
- **Test**をクリックします。

Test the connection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。

- **Close**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接続

Wizard: "Configure the connector".

Configure the database server connection.

Configure the database server connection.

Server

User

Password

Database

Enter the table owner if different from the user.

Table owner

Test the connection

Test

< Previous Next > Finish Cancel

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
- ユーザ
- パスワード
- サーバデータベース
- テーブルの所有者

The database server

Sybaseデータベースのサーバの名前を指定します。

User

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。

Password

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。

Server database

接続先のデータベースの名前を入力します。

Table owner

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。

Test

このボタンを押すと接続をテストできます。

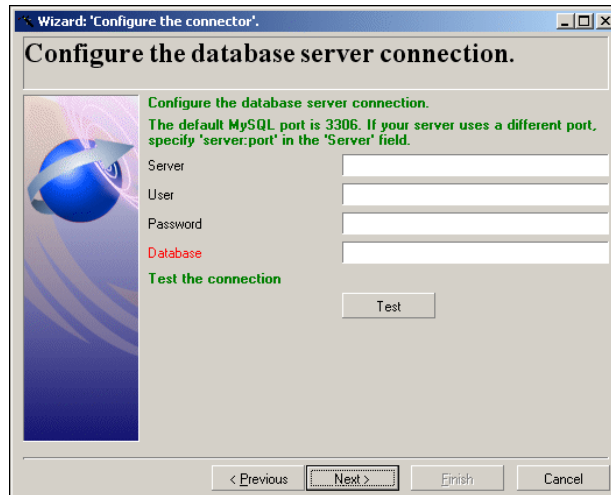
接続をテストするには：

- 接続用パラメータを入力します。
- **Test**をクリックします。

Test the connection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。

- **Close**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 - MySQL接続



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
- ユーザ
- パスワード
- サーバデータベース

Database server

MySQLデータベースのサーバの名前を指定します。

User

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。

Password

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。

Server database

接続先のデータベースの名前を入力します。

Test

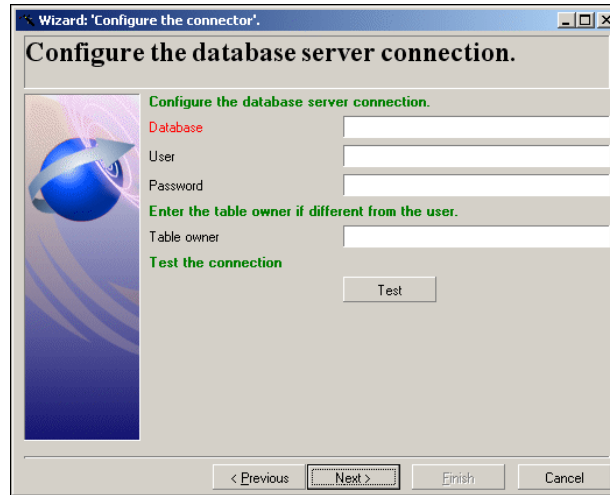
このボタンを押すと接続をテストできます。

接続をテストするには：

- 接続用パラメータを入力します。

- **Test**をクリックします。
Test the connection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。
- **Close**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



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
- ユーザ
- パスワード
- サーバデータベース
- テーブルの所有者

The database server

DB2データベースのサーバの名前を指定します。

User

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。

Password

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。

Table owner

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。

Test

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：

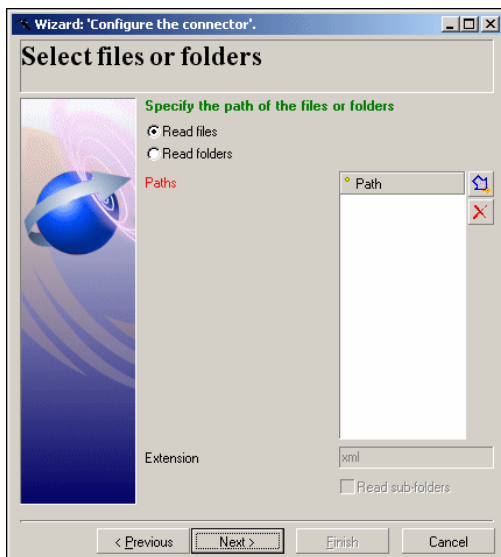
- 接続用パラメータを入力します。
- **Test**をクリックします。

Test the connection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。

- **Close**をクリックして設定ウィザードへ戻ります。

Select files or folders

ウィザードのこのページで、読み込み対象とするファイルとフォルダの場所を指定できます。



Read foldersオプションを選択する場合、以下の項目を指定する必要があります。

- フォルダのパス
- ファイルの拡張子名

このフィールドのデフォルト値はコネクタによって異なり、.xml、.gz、.xsfがあります。

拡張子名は任意です。セミコロン区切りのリストを作成することで、複数の拡張子を定義できます。

Read sub-folders オプションも選択する必要があります。

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に関連付けられているフォルダにあるファイルやフォルダのパスを指定する場合は、このドライブ文字は指定しないでください。この文字に関連付けられているフォルダのみを指定してください。

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」の節の「ドキュメント要素のパス」を参照してください。

Read files

読み込み対象とする複数のファイルを定義するには、以下の手順を実行します。

- 1  をクリックします。
- 2 表示されたフィールドで、 をクリックしてから適切なファイルを選択します。
- 3 この操作を必要な回数繰り返します。

このオプションを選択すると、ウィザードで、コンピュータやネットワーク上に存在するXMLファイルに対するパスの入力が求められます。

Read folders

このオプションを選択すると、ウィザードで、以下のフィールドへの入力が求められます。



■ Folder name

ファイルを含むフォルダのパスを入力します。

■ Extension

コネクタが読み取るファイルの拡張子を入力します。.xmlなどです。

読み込み対象とする複数のフォルダを定義するには、以下の手順を実行します。

- 1  をクリックします。
- 2 表示されたフィールドで、 をクリックしてから適切なファイルを選択します。
- 3 この操作を必要な回数繰り返します。

Read the sub-folders

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。

Advanced configuration（詳細モード）

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。

SQL92 supported

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。

アーカイバル

コネクタが処理後のアクションを使用する場合、移動済みのファイルをアーカイブできます。**Move it to folder**オプション選択時にアーカイバルが有効になります。これは移動が失敗しても成功しても実行されます。

アーカイバルは、特定期間にわたって、または特定の保持するバージョン数にわたって行われます。

Keep archiveオプション選択時にアーカイバルは有効有効になります。

各アーカイブ済みファイルには、定義済みの設定に従い、サフィックスが添付されます。

アーカイブ設定

アーカイバルを設定するには、以下に挙げる1項目を定義します。

- ファイルを保持する時間の限度（時間、日、週、月、年）。
この期間は**Date**フレームで定義します。
- 保持するファイルの最大バージョン数。
Number of versions of a file to keepフィールドに入力します。
- 無限アーカイバル。
Keep allオプションを選択します。

日時を使用したアーカイバル

日時を使用したアーカイブを行う場合、各ファイル名に日時が添付されます。使用する日時フォーマットを指定する必要があります。

日時フォーマットの例を以下に挙げます。

- yyyy-MM-dd-HH-mm-ss
- yyyyMMddHHmmss
- dd-mm-yyyy hh:nn:ss
- ddmmyyyyhhnnss

ex_2003-09-02-16-20-05などです。

注意:

サフィックスが行われた日時のみが考慮されます。ファイル自体の作成日は考慮されません。

番号を使用したアーカイバル

日時ではなく番号を使用したアーカイブを行う場合、各ファイルには番号が添付されます。

ファイル名の番号の増え方を指定する必要があります。この増分は、**Numbering** フレームで定義します。

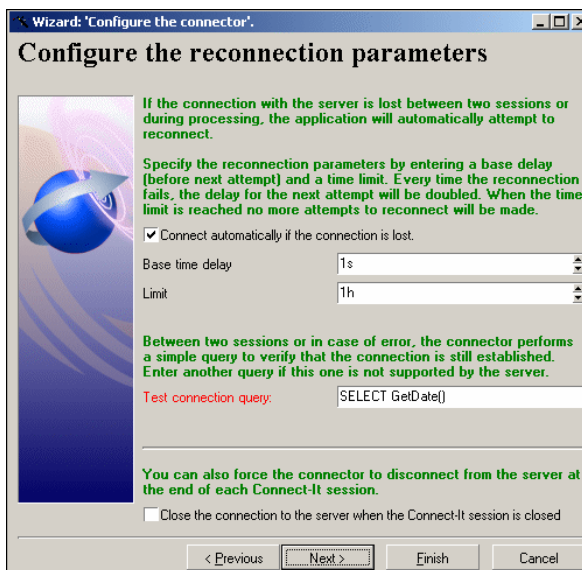
増分は以下のいずれかです。

- **Ascending sort** : 各ファイルキーブが、キューの最後に追加されます。
たとえば、ex_1、ex_2、ex_3というファイルがあれば、最新のファイルはex_3です。
- **Descending sort** : 各ファイルキーブが、キューの最初に追加されます。
たとえば、ex_1、ex_2、ex_3というファイルがあれば、最新のファイルはex_1です。

ファイル名のサフィックスとして使用する桁数と使用する区切り文字は設定できます。

Configure the reconnection parameters (詳細モード)

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。



Connect automatically if the connection is lost

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。

■ **Base time delay**

base time delay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。

■ **Time limit**

このtime limit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と表示されなければなりません。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。

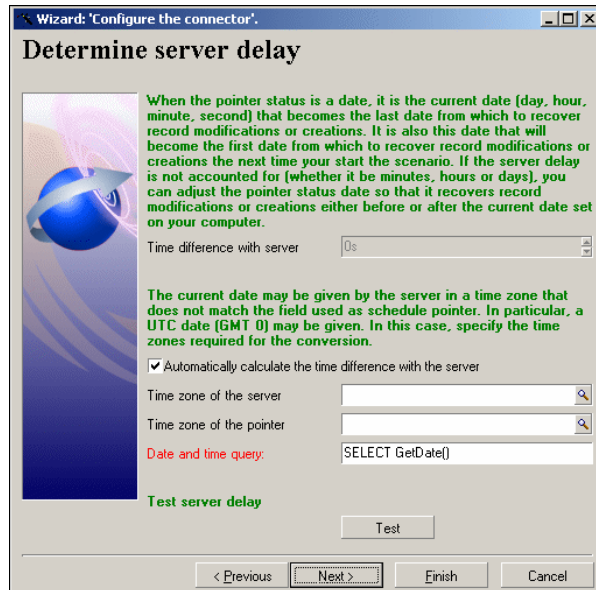
接続が切断されない場合は、このオプションを無効にして、警告メッセージがコネクタの開始時に表示されないようにします。例えば、ExcelファイルへのODBC接続の場合などです。

Closing the connection to the server when the Connect-It session is closed

Connect-Itサポート技術者による推奨がある場合を除いて、このオプションは選択しないでください。

Determine server delay (詳細モード)

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。



Automatically calculate the time difference with the server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。

Automatically calculate the time difference with the serverオプションが選択されていない

Automatically calculate the time difference with the server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。

Date and time query

このフィールドで、スケジューリングポイントがローカルで使用する時刻を補正するための、サーバの時刻を返すのに使用するクエリを指定できます。

クエリが実行される条件を以下に挙げます。

- コネクタが開いた場合
- セッションが開いた場合に、サーバとローカル時刻間の遅延を更新するため


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、**Time zone of the server**フィールドと（スケジュールの）**Time zone of the pointer**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。
- これらのタイムスタンプをフィルタする操作が、Connect-Itサーバの「日付」パラメータで定義されているタイムゾーンに対応する場合、外部アプリケーションは指定されたタイムゾーンでの現在のタイムスタンプを返します。

タイムゾーンを指定する

- 1 **Automatically calculate the time difference with the server**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。
- 3 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、**Apply daylight savings time**オプションを選択します。

- 4 **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

Test time difference with serverの右にある**Test**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。

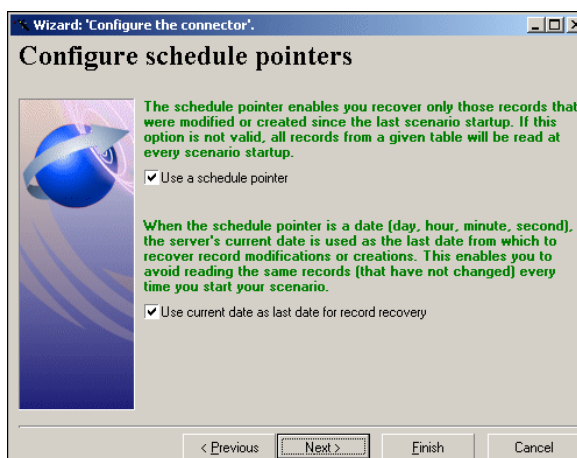
- 外部アプリケーションとの接続
- 外部アプリケーションの現在の日付と時刻が取得され、表示されます。
- タイムゾーンが指定されている場合は、タイムゾーンと時間調整をした後の現在の日付と時刻が表示されます。

タイムゾーンのパラメータをテストする

- 1 外部アプリケーション内のレコードを変更します。
- 2 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。
- 3 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、**Test**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。
- 4 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。

Configure schedule pointers (詳細モード)

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タを設定できます。



シナリオがスケジュールモードで起動していると、生成ドキュメントタイプのポインタ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。

Use a schedule pointer

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。

Use current date as last date for record recovery

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。

Use current date as last date for record recovery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データののみを処理します。

- 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可能である場合、サーバ遅延が調整されます）
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。

Use current date as last date for record recovery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データののみを処理します。

- 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。

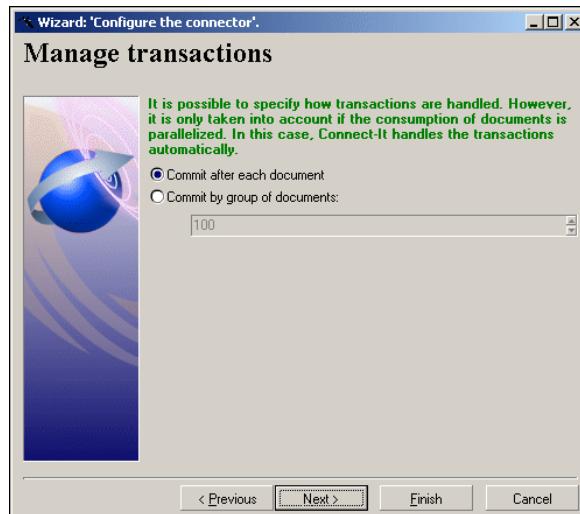
 **注意:**

スケジュールポイントの日付では、ローカルコンピュータ向けに定義されているフォーマットを使用します。このフォーマットは、現在のシナリオのフォルダにある[シナリオ名].iniファイルに保存されます。

スケジュールポイントの詳細については、『ユーザガイド』、「Putting a scenario into production (シナリオの本番投入)」の章の「Reconciliation keys (照合更新キー)」の節を参照してください。

Manage transactions (詳細モード)

このページでは、ドキュメント転送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。

- **Commit after each document**
各ドキュメントの処理後にCommit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。
- **Commit by group of documents**

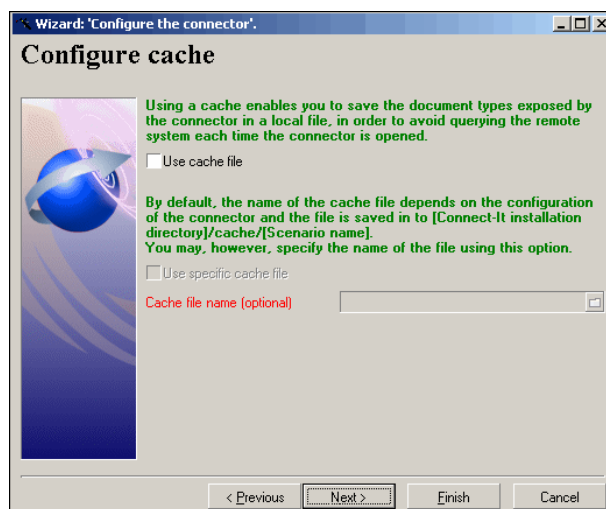
指定された数のドキュメントが処理された後、**Commit**コマンドが実行され、ドキュメントの処理時に発信された挿入クエリ、更新クエリまたは削除クエリを有効にします。ドキュメント数のデフォルト値は、**100**です。

 **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**Commit**を使用することをお勧めします。

Configure the cache (詳細モード)

このページでは、コネクタが発行する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。

- **Use cache file**

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、別の名前が付きません。一般的にファイルにはサーバ名が使用されます。

- **Use specific cache file**

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、コネクタを最初に開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 Scenario diagram内でコネクタを選択します。
- 2 右クリックします。
- 3 **Cache/Synchronize**ショートカットメニューを選択します。
(**Tool/Cache/Synchronize the cache**)

Advanced options (詳細モード)

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 **Name**の列にオプションの名前を入力します。
- 3 **Value**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 **X** 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

以下の表に、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を挙げます。

警告:

HP Software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 - 詳細設定オプション

Name	Default value	Description	Data type	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	ブーリアン： <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	ブーリアン： <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 <i>commit</i> の自動モード (1) または手動モード (0) を指定する	ブーリアン： <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	ブーリアン： <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

Name	Default value	Description	Data type	例
ExtendedFetch	1	拡張取り出しの使用	ブーリアン： ■ 0 = False ■ 1 = True	以下の場合0を入力する ■ Pilote ODBC Novell
AddIndexDesc	1	インデックスの自動記述を許可する	ブーリアン： ■ 0 = False ■ 1 = True	以下の場合0を入力する ■ 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 ■ Pilote ODBC Novell
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 ■ 0 FORWARD_ONLY ■ 1 KEYSET_DRIVEN ■ 2 KEYSET_DRIVEN ■ 3 STATIC	以下の場合0を入力する ■ DB2 NT
IsODBC3Compliant	1			以下の場合1を入力する ■ Oracleネイティブ

Define document types (詳細モード)

このページで、コネクタが発行するドキュメントタイプの拡張ファイルを定義できます。

このファイルの内容を以下に挙げます。

- データベースの異なるテーブル間の結合の定義。
- スケジュールポイントとして使用するフィールド。

元のファイルは変更できません。拡張ファイルの編集の詳細については、『ユーザガイド』、「*Configuration editor (設定エディタ)*」の章を参照してください。

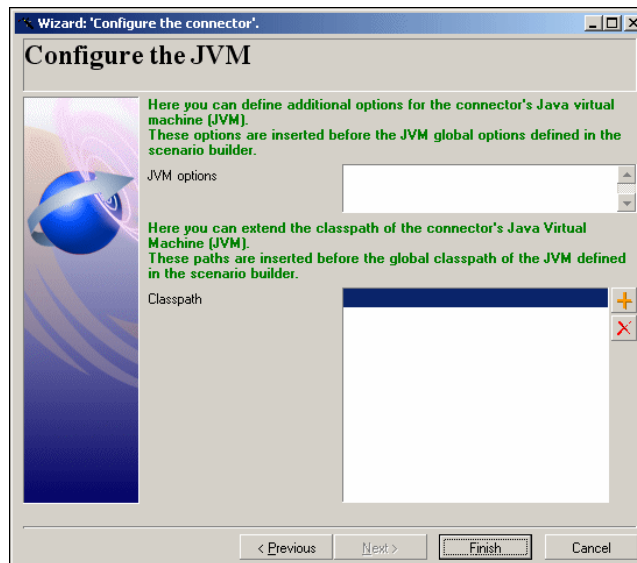


警告:

一部の拡張ファイルは設定エディタでは変更できません。この場合、データベースコネクタ（Connect-Itのインストールフォルダにある config\database\config\database.cfgファイル）の拡張ファイルを使用する必要があります。

Configure the JVM

コネクタ設定ウィザードにあるこのページは、JMVを使用するすべてのコネクタで利用できます。



Connect-Itで、アプリケーション全体に対してJVMオプションとクラスパスを指定するには、**Java/Configure the JVM**メニューを使用します。Configure the JVMページで、グローバルレベルで定義したパラメータを個別指定し、コネクタの追加JVMパラメータを定義できます。

設定フィールドを以下に挙げます。

- **JVM options** : このフィールドを使用して、**Java/Configure the JVM**メニューで定義したのと同様の追加オプションを指定します。追加オプションが定義されると、コネクタは自動的に再展開されます。



注意:

このJVMオプションの値は、シナリオ設定をエクスポートすると公開されます (**-export**パラメータ)。

- **ClassPath** : このフィールドで、要素をクラスパスに追加できます。ここで定義したクラスパス要素はJVM一般設定で定義したクラスパス要素より優先します。



注意:

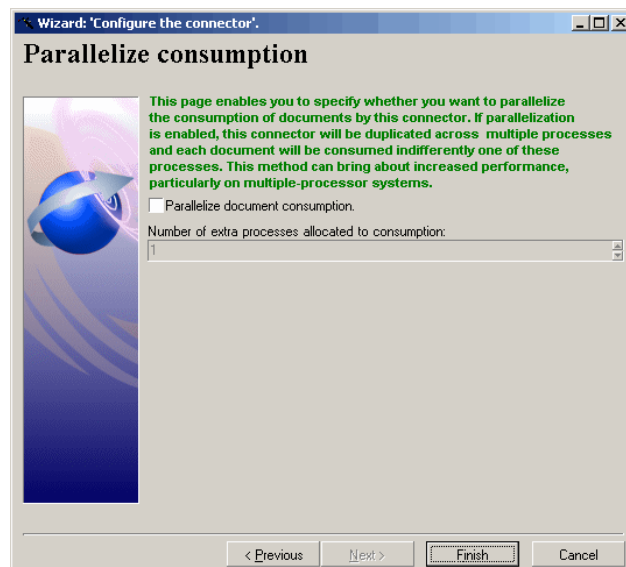
このプロパティは、シナリオ設定をエクスポートすると公開されます (**-export**パラメータ)。

Parallelize consumption (詳細モード)

並列化は、ドキュメントタイプを取り込むすべてのコネクタに適用されます。並列化には、ドキュメントの取り込みを並列で処理するためのコネクタの複製、複数プロセスの作成が含まれます。

このページでは、

- このオプションを有効にします
- 並列実行するプロセス数を定義します



Parallelize the consumption of the documents

このオプションを選択して、取り込みドキュメントタイプの並列処理を有効にします。

Number of additional processes allocated to consumption

このフィールドには、同時に実行し、ドキュメントの取り込みに割り当てられるプロセス数を入力します。

2 コネクタのルール（ディレクティブ）

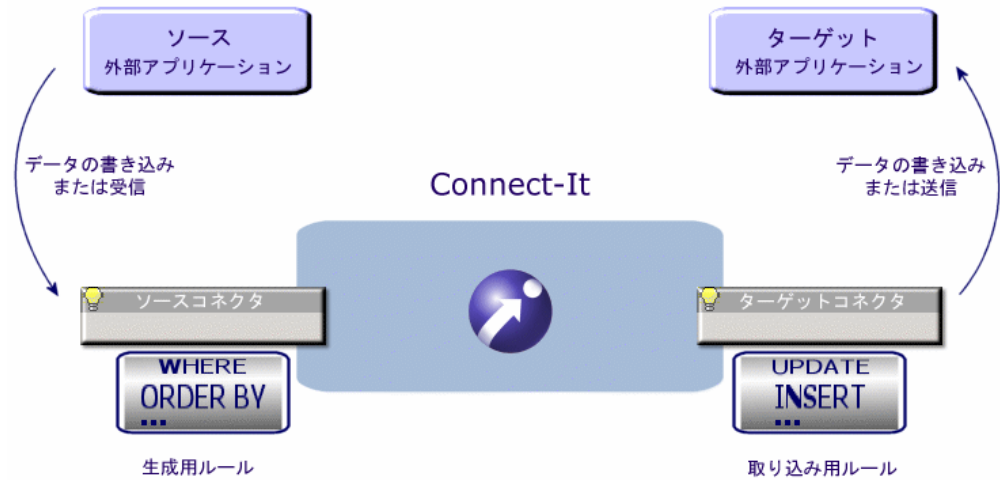
ルール（ディレクティブ）とはユーザが定義するコネクタ用の指示内容のことです。コネクタは、ドキュメントの生成または取り込み時にこの指示に従います。

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照合更新に相当し、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。

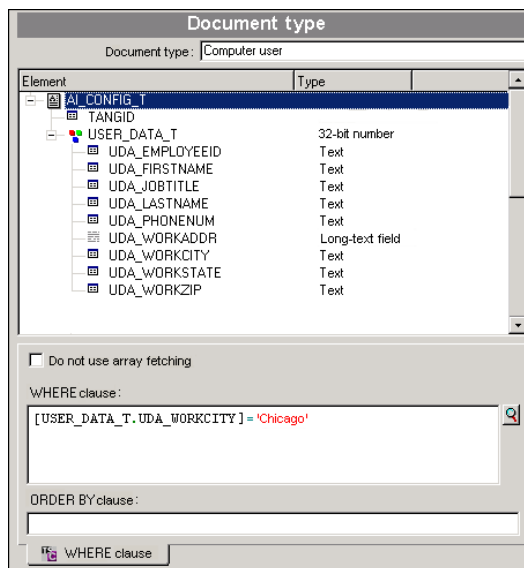
生成用ルールの定義

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 (**Tools/Open** または **F4** キー)。
- 4 **Document types** タブを選択します。
- 5  をクリックするか、または **Produced document types** 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句を使用して、ソースデータベースにあるレコードをフィルタします。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ことができます。これには、以下の句を使用します。

```
[レコードの作成日を含むフィールド] >= '1/01/2002'
```

リンクをフィルタするには、以下のシンタックスを使用します。

```
[OperatingSystem.OperatingSystem_ID] = 1000001107
```

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 [献 59]

データベースコネクタ [献 205]

LANDesk for Inventory 8コネクタ [献 323]

Tivoli Inventoryコネクタ (バージョン4.0) [献 336]

System Management System (SMS) コネクタ [献 328]

CA Unicenter AMO 3コネクタ [献 351]

Tivoli CM Inventory 4.2コネクタ [献 339]

sysdateフィールド

AQLで記述するWHERE句では、**sysdate**フィールドを**getdate()**関数に置換する必要があります。

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。

Do not use array fetching

Blobまたは**memo**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。**array fetching**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の定義

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。

- 取り込みドキュメントタイプ編集ウィンドウ内
- マッピング編集ウィンドウ内


取り込み編集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、取り込みドキュメントタイプ編集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 (**Tools/Open**または**F4**キー)。
- 4 **Document types**タブを選択します。
- 5 をクリックするか、または**Consumed document type**の枠内で取り込みドキュメントタイプを編集します。
取り込み編集ウィンドウが表示されます。ここで取り込みドキュメントタイプを作成または編集します。
- 6 ウィンドウに取り込み用ルールを入力します。

マッピング編集ウィンドウ内で作成する取り込み用ルール

マッピング編集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 **Mappings**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。
マッピング編集ウィンドウが開きます。
- 5 **Mappings**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。

照合更新

データベース型のコネクタでは、照合更新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照合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)

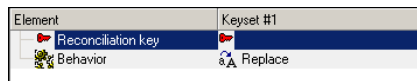
取り込み用ルールは、**Reconciliation**タブと**Advanced reconciliation**タブ内で入力します。

Reconciliationタブの使用方法

Reconciliation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、照合更新のタイプを指定する
- 2 照合更新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 - **Reconciliation**タブ



『ユーザガイド』、「Putting a scenario into production（シナリオの本番投入）」の章の「Creating a service（サービスの作成）」の節も参照してください。

照合更新のタイプを選択する

照合更新のタイプでは、実行するアクションを選択できます。

- Update or insert
- Update
- Insert only
- Delete

照合更新のタイプを指定する

- 1 ルートノード、構造体、またはコレクションをMappingsウィンドウの作業枠内で選択します。
- 2 **Reconciliation type**のドロップダウンリストからオプションを選択します。

照合更新キーを選択する

複雑な要素用に照合更新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照合更新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は、照合更新キーで識別されます。

照合更新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**フィールドは照合更新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**serial number**フィールドは、照合更新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

照合更新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、照合更新がより迅速に実行されます。

照合更新キーを選択する

1 要素を作業枠内で選択します。

2 **Reconciliation key**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：pdi\pdi8ac44\pdia.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**フィールドが照合更新キーとして選択されています。

構造体が照合更新キーとして選択されている場合

Asset Managementコネクタが発行する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照合更新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照合更新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**Identifier**フィールドを含んでおり (*not shown on-screen*)、この識別子が他のテーブルへのリンクを可能にします。構造体を照合更新キーとして指定するということは、この**Identifier**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。



上の図は、資産のドキュメントタイプが作成するレコードを示しています。

- **Assets** ルートノードの **Name** フィールド
- **Product** 構造体の **Identifier** フィールド
- **Product** 構造体の **Model** フィールド

親のIDを照合更新キーとして使用する

Reconciliation type	Update or insert	Update or insert
Reject in case of error	<input type="checkbox"/>	<input type="checkbox"/>
Weight of key set	100	100
Action in case of a replication conflict	Generate a reject	Generate a reject
Use the reconciliation cache	<input type="checkbox"/>	<input type="checkbox"/>

1 コレクションを選択します。

2 **Use the parent Id as the reconciliation key** 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 **lparentId** フィールドによりリンクされています。

Use the parent Id as the reconciliation key 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照合更新キーとして選択されています。



Use the parent Id as the reconciliation key オプション - 例

この例では、**Use the parent Id as the reconciliation key** 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 Use the parent Id as the reconciliation keyを選択した場合、

Use the parent Id as the reconciliation key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。

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

2 Use the parent Id as the reconciliation key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。**Parent Id**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					
Name	資産タグ	識別子	親識別子	Name	資産タグ	
コンピュータB	COMPB001	CPBBB1	- ❶ -	❶CPBBB1	プリンタX	PRNTX001
コンピュータA	COMPA001	CPAAA1				

親の変更を有効にする方法

Use the parent ID as the reconciliation keyオプションを使用すると、レコードの挿入や更新の前に親の存在が確認されます。親要素が変更されている場合、このレコードへのクエリは失敗し、エラーがConnect-Itログに保存されます。

親レコードとその依存関係（コンピュータに接続されているモニタなど）間の変更に合わせてレコードを更新するには、**Enable parent change**オプションを使用します。

このオプションはクエリをAssetCenterデータベースに送信し、親IDではなくユーザが定義した照合更新キーを使用します。レコードが検出されれば、照合更新エンジンは親IDを子レコードに追加し、依存関係を作成します。

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 **Reject the parent node if rejection** 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 (**amPortfolio**) 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。

複数の照合更新キーセットの使用

Connect-Itでは、複数の照合更新キーセットを使用できます。各キーには重みが割り当てられます。

これは、選択した照合更新キーではレコードの照合更新を実行できない場合に便利です。例：照合更新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。

ある照合更新キーが2番目のセットに所属すると指定すると、1番目の照合更新キーセットで照合更新に失敗した場合に、Connect-Itはこのキーを使用します。

照合更新キーの設定についての詳細は、『ユーザガイド』の「シナリオの本番投入」の章「照合更新キー」の節を参照してください。

Case-sensitive reconciliation

このオプションにより、Connect-Itの照合更新は以下のようになります。

- **Case-sensitive reconciliation** 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、照合更新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照合更新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。

- **Case-sensitive reconciliation** 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、照合更新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照合更新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、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のみ表示されます。ODBC接続を使用するコネクタの場合、このオプションはデフォルトで選択されます。

照合更新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、以下のオプションから選択します。

- **Replace**
- **Add**

デフォルトでは、**Replace**オプションが選択されています。**memo**フィールドでは**Append**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。

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つのオプションを利用できます。

- Generate a reject
- Log warning if new entry or element type modified
- Overwrite

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 [献 373]」の章の「ServiceCenterシナリオ [献 416]」の以下の節を参照してください。

- データの照合更新とデータの複製の相違点 [献 419]
- Connect-It内でのデータの複製の管理 [献 420]

Advanced Reconciliationタブを使用する

advanced reconciliation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、コレクション用の照合更新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 - Advanced reconciliation タブ



コレクションの照合更新を実行する

Activate collection reconciliation 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

Action on members not found in the source document field 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 *Status* 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 *Status* 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：

```
Status = "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、**Reconciliation filter** 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。**Action on members not found in the source document** 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、**Collection reconciliation filter** フィールドにフィルタ句を作成します。

図 2.4. 照合更新 - コレクションのフィルタ



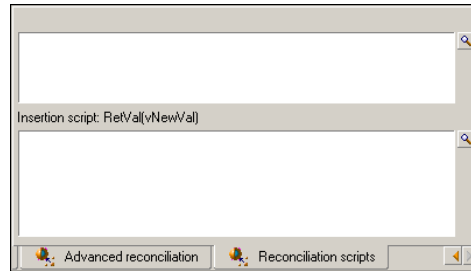
図 2.5. 照合更新 - ターゲットコレクションの構成要素の削除



Reconciliation scriptsタブの使用方法（Asset Managementコネクタの場合のみ）

Asset Managementコネクタが取り込みモードである場合、**Reconciliation scripts**タブが利用できます。

図 2.6. 取り込み用ルール - **Reconciliation scripts**タブ



このタブで、生成ドキュメントと取り込みドキュメントの2つのドキュメントタイプの照合更新に必要なスクリプトを入力できます。

これらのスクリプトで、ターゲットコネクタのドキュメントタイプに対し、以下のアクションを実行できます。

- ターゲットデータベースに値が存在する場合の更新
- ターゲットデータベースに値がない場合の値の挿入

使用シンタックス

更新スクリプトと挿入スクリプトのシンタックスを以下に挙げます。

- **vNewVal** : ソースドキュメントの値
- **vOldVal** : ターゲットデータベースの値
- **vOldId** : 変更対象レコードのIDの現在値

大括弧[]とBASIC関数をスクリプトで使用できます。

例 :

```
// 従来の値を保持
RetVal = vOldVal
If vNewVal >= vOldVal and [dtLastModif] > '1980-11-12 10:11:12' Then
pifSetPendingDocument('Document not finished')
DbExecAql('INSERT INTO .....')
// 新しい値を取得
RetVal = vNewVal
End If
```

照合更新スクリプトの動作方法

照合更新スクリプトは、マッピングの管理方法を改善するための追加手順です。

これらのスクリプトは、コネクタが生成するドキュメントタイプや他のコネクタが取り込むドキュメントタイプに含まれるデータに関係します。ターゲットデータベースのフィールドにデータが含まれるかどうかに応じて、スクリプトでマップされたフィールドの値の変更（更新）や、作成（挿入）を行えます。

例：

ソースコネクタは、以下のフィールドを含むドキュメントタイプAを作成します。

- Name
- Id
- Comment

IDフィールドが照合更新キーです。

Asset ManagementターゲットコネクタはドキュメントタイプAを取り込み、同じフィールドを持つターゲットデータベースに接続します。

ドキュメントの値を以下に挙げます。

- Name = Test
- Id = 1
- Comment = Transfer

ターゲットデータベースの値を以下に挙げます。

- Name = Currency
- Id = 1
- Comment = (空)

Nameフィールドの更新スクリプトを以下に挙げます。

```
RetVal=vOldVal
```

Commentフィールドの挿入スクリプトを以下に挙げます。

```
RetVal=vNewVal
```

照合更新スクリプトの実行後、Asset Managementが取り込んだドキュメントには、以下の情報が含まれるようになります。

- Name = Currency
- Id = 1
- Comment= Transfer

照合更新と照合更新に関する詳細については、『ユーザガイド』、「*Putting a scenario into production* (シナリオの本番投入)」の章の「*Reconciliation* (照合更新)」の節を参照してください。

3 Hewlett-Packardコネクタ

HP Softwareコネクタは、Hewlett-Packardアプリケーションのデータを処理するコネクタです。

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コネクタについて

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。

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。

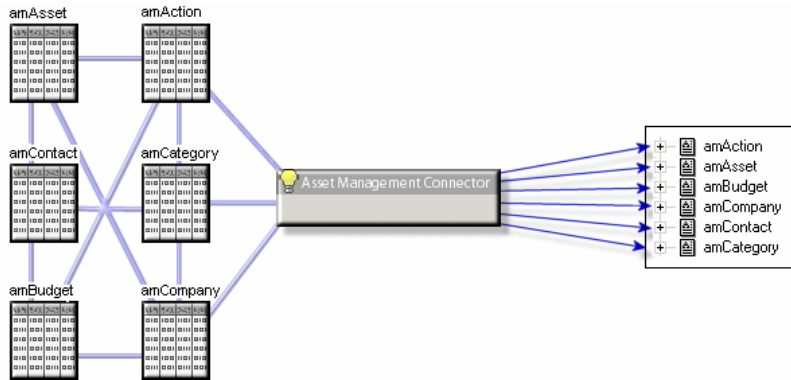
Asset Managementアプリケーションクライアントは、Connect-Itがインストールされているコンピュータにインストールされなければなりません。



注意:

例えば、AssetCenterデータベースにあるそれぞれのテーブルは、シナリオビルダ内でAsset Managementコネクタで発行された単一ドキュメントタイプに対応します。

図 3.1. Asset Managerコネクタ - 発行ドキュメントタイプ



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 (AssetCenterバージョン4.4)

AssetCenterバージョン4.4を使用している場合、接続を記述するamdb.iniファイルを編集し、**UseSql92Join=1** パラメータを追加する必要があります。

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

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が複数のamdb.iniファイル内で宣言されています。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コネクタが発行する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。

照合更新時に、Asset Managementコネクタが発行する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。

バッチの管理

バッチで管理される要素には特別なマッピングを使用します。モニタのマッピングは、**AddOn**コレクションで実行され、その属性は**MON**です。特定のコンピュータに接続されている特定のモニタが、単一レコードに追加されないようにするため、この属性は、モニタのバッチとして管理されます（管理条件は'Asset tag'です）。AssetManagementコネクタの付属シナリオの管理では、**AddOn**コレクションのCode要素は、モニタのSerialNumberフィールドと合わせてマップされ、結合メカニズムは適用されません。 \n

UNIXインストール

AssetCenterデータベースをUNIXで使用する場合、AssetCenter 『インストールとアップグレード』マニュアルの「*Installing and configuring in UNIX (UNIXでのインストールと設定)*」を参照してください。

注意:

環境変数の宣言および接続ファイル (amdb.ini) の作成に関する手順を完全に遵守していることを確認します。

AssetCenterデータベースを正しくインストールしたら、以下の手順を実行します。

- 1 Windowsでシナリオを開発し、Asset ManagementコネクタをUNIX設定に適応します。
- 2 接続の宣言とシナリオの作成を終えたら、WindowsフォルダのConnectIt/*scenario*フォルダに保存します。
- 3 保存したシナリオを、UNIXのConnectIt/*scenario*にコピーします。
- 4 サービス（デーモン）を作成し、シナリオを稼動状態にします。

UNIXサービスの作成に関する詳細については、『ユーザガイド』、「*Putting a scenario into production (シナリオの本番投入)*」の章の「*Creating a service (サービスの作成)*」の節を参照してください。

AssetCenterのマイグレーションに関する注意事項

Connect-Itは、データ転送のみではなくデータ構造の変更も行い、特別なAssetCenterマイグレーションツールと併用します。Connect-Itのみを使用して、古いフォーマットのAssetCenter本番データベースを変換しないでください。

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。
- WHERE句とORDER BY句で使用するシンタックスに、フィールド名の大括弧[]を含めることはできません。

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 [献 15]」章の「コネクタの設定 [献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.1. Asset Managementコネクタの設定

Name and describe the connector [献 16]
Configuring the connection [献 19]
Advanced configuration (詳細モード) [献 64]
Configure the reconnection parameters (詳細モード) [献 29]
Determine server delay (詳細モード) [献 31]
Configure schedule pointers (詳細モード) [献 33]
Manage transactions (詳細モード) [献 35]
Configure the cache (詳細モード) [献 36]
Advanced options (詳細モード) [献 65]
Define document types (詳細モード) [献 39]
Parallelize consumption (詳細モード) [献 41]

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「コネクタの設定 [献 15]」の節を参照してください。

Configuring the connection

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configuring the connection [献 19]」の節を参照してください。



警告:

このページが表示されるのは、オプション**Display advanced configuration wizards**を選択した場合のみです (EditのOptionsメニュー)。

Dynamic library

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の完全パスを指定できます。たとえば、AssetCenter 4.4の場合、C:\Program Files\HP Software\AssetCenter 4.40 en\bin\aamapi44.dllです。デフォルトで、コネクタは自動的に使用するDLLを見つけます。

Expose the primary key of the tables

このオプションを選択すると、Asset Managementコネクタが発行するドキュメントタイプには、データベース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、コネクタが発行する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。

Expose the foreign keys of the tables

このオプションを選択すると、Asset Managementコネクタが発行するドキュメントタイプに、データベースのテーブルの外部キーに対応するフィールドが含まれます。

Executing reconciliation scripts

このページの使用方法については、「コネクタの設定 [献 15]」章の「取り込み用ルール - Reconciliation scriptsタブ [献 56]」の節を参照してください。

Executing an initial import

このオプションで、データベースが空である場合や、挿入対象の項目がまだ存在しない場合などに、初期インポートの照合更新スクリプトでSELECTクエリを使用しないで済ませることができます。これにより処理時間が短縮されます。

SELECTの実行対象外を以下に挙げます。

- ルート要素
- **Follow the link**オプションが有効であり、親ドキュメントが存在しない場合の1対1リンク
- 親要素が挿入されてすぐの場合の1対Nリンク

Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the reconnection parameters (詳細モード) [献 29]」の節を参照してください。

Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Determine server delay (詳細モード) [献 31]」の節を参照してください。

Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure schedule pointers (詳細モード) [献 33]」の節を参照してください。

Manage transactions (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Manage transactions (詳細モード) [献 35]」の節を参照してください。

Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the cache (詳細モード) [献 36]」の節を参照してください。

Advanced options (詳細モード)

このページでは、取り込みドキュメントタイプの一部の要素の詳細を指定できません。詳細オプションでは、使用されている完全名が標準名でない場合のみ、各ドキュメントの完全名構造体 (**FullName**) を定義できます。**FullName** フィールドは、アプリケーションが生成する一意なフィールドです。

従業員テーブル (**amEmplDept**) では、オプションは以下のように入力します。

Name	Value
amEmplDept.FullNameDef	lParentId,Name,bDepartment,[Name],[FirstName], [IdNo]

オプションの値フィールドで定義する構造体を以下に挙げます。

- **lParentId** : 階層フィールド
- **Name** : 階層の作成に使用する階層フィールド
- **bDepartment** : レコードが完全名にリンクされている最終レコードであるかどうかを判断できる、テストフィールド
- **[Name]**、**[FirstName]**、**[IdNo]** : 完全名の最終レコードの構造体です。フィールドは、大括弧[...]に含める必要があります。

Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Define document types (詳細モード) [献 39]」の節を参照してください。

Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Parallelize consumption (詳細モード) [献 41]」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 (ディレクトティブ) [献 43]」章の「生成用ルール [献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。これらの句の詳細については「コネクタのルール (ディレクトティブ) [献 43]」章の「生成用ルール [献 44]」節、「WHERE句とORDER BY句 [献 45]」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 [献 435]」の章を参照してください。

AssetManagementコネクタで利用可能なオプションの詳細については、「照合更新 [献 47]」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

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)

sysdateフィールド

AQLで記述するWHERE句では、**sysdate**フィールドを**getdate()**関数に置換する必要があります。

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。

例：最近1日以内に更新された全レコードを取得するには、**DaysDiff(GetDate(), dtLastModif) <= 1**とします

外結合

生成ドキュメントタイプに非排他リンク (**LEFT OUTER JOIN** SQL関数に対応)が含まれる場合、**Make an outer join**が利用可能になります。

このオプションで、リンクが非排他であるテーブルの全レコードを取得できます。

注意:

このオプションは、AssetCenter 4.3以上でのみ機能します。

例として、従業員に割り当てられている全コンピュータを取得するコンピュータテーブルと従業員テーブル間のマッピングでの動作を以下に挙げます。

- オプション選択時：従業員に割り当てられているコンピュータのみ取得します
- オプション未選択時：割り当てられていないコンピュータも含めたすべてのコンピュータを取得します

Follow link

このオプションは、任意の選択した構造体の**Reconciliation**タブで有効にできます。

このオプションが有用となる2つの場合を以下に挙げます。

- オーバーフローテーブルの場合
これらのテーブルの特異な点は、参照テーブルに1対1リンクでリンクされていることです。このため、照合更新キーの使用はオプションです。参照テーブルの項目のリンクをたどることで、このリンク先テーブルからの情報を取得できます。
- 信頼できる照合更新キーを定義できない場合（従業員IDではなく、名前と姓で照合更新を実行する場合など）

Follow linkは、ポートフォリオ品目テーブル（amPortfolio）に対し特に有効であり、特定要素の照合更新キーを定義することなく、このテーブルと資産テーブル（amAsset）間のリンクをたどることができます。

 **注意:**

Update or insert modeでこのオプションを使用することを推奨します。

コレクション-コレクション間マッピング

- N対Nリンクをマッピング（コレクション対コレクションマッピング）する際、**Use the parent ID as a reconciliation key**オプションがデフォルトで選択されます。このタイプのマッピングでエラーを回避するには、このオプションを選択解除します。
- コレクション-コレクション間マッピングの詳細な照合更新中、**Minimize the number of requests when reconciling collections**オプションにより以下の動作が行われます。
 - **SELECT**タイプクエリがコレクションを構成する要素のセットに対して実行されます。
このオプションが有効ではない場合、クエリはコレクションを構成する各要素に対して実行されます。

▶ コレクションの照合更新を実行する [献 53].

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） [献 43]」章の「取り込み用ルール [献 46]」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、**Reconciliation**と**Advanced reconciliation**タブ内で照合更新用のパラメータを入力します。

これら照合更新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） [献 43]」章の「取り込み用ルール [献 46]」節内「照合更新 [献 47]」を参照してください。

AssetManagementコネクタが生成するドキュメントタイプ

AssetManagementコネクタが生成するドキュメントタイプにより、AssetCenterテーブルが公開されます。

特に、生成ドキュメントタイプは、AssetCenterバージョン4.4向けに開発された正規化機能を記述します。

この機能は、OpenView Desktop Inventory 8コネクタ用に提供されているシナリオにあります。

図 3.2. PDI 8 - AMマッピング

Element	Mapping
TcpIpAddress	[hardwaredata.hwNetworkData.hwNetworkCards.hwNetwork...
TcpIpDomain	[hardwaredata.hwNetworkData.hwNetworkTcpip.hwIPDome...
TcpIpHostName	[hardwaredata.hwNetworkData.hwNetworkTcpip.hwIPHostN...
VideoCard	[hardwaredata.hwVideoData.hwDisplayGraphicsAdapters.hw...
Work.group	[hardwaredata.hwNetworkData.hwNetworkNames.hwWorkg...
Portfolio	
Asset	
User	' Use a Follow link on this field since we can not reconcile
AddOn	
dtInvent	PifUserFmtStrToVar([hardwaredata.hwSystemData.hwScanC...
Folder	[applicationdata.application.maindir]
InventModel	
BrandName	[applicationdata.application.publisher]
InventoryKey	RetVal = "PDI" & [applicationdata.application.versionid]
ModelName	[applicationdata.application.name]
Version	[applicationdata.application.version]
ModelTmp	
BarCode	RetVal = "sysUNKNOWN_SOFT"
Model	
SoftInstall	
AddOn#1	If [hardwaredata.hwVideoData.hwDisplayMonitors.hwDisplay...
AddOn#2	' Information on memory device can be present in hardware...
AddOn#3	' Information on memory device can be present in hardware...
Document	

AssetCenterバージョン4.4から、ソフトウェアのインストールモデルとその名前とを、外部アプリケーションで得られたインベントリの結果とを関連付けることが可能になりました。たとえば、この機能により、（インベントリツールによる）ソフトウェアモデルの名前と、AssetCenterでの（カタログなどによる）内部名とを関連付けることができます。これは、**amInventModel**テーブルで定義され、外部アプリケーションによる作成した照合更新キーと、ソフトウェアの一意の識別子（PDI|1234など）に依存します。このキーには、ソフトウェア名とそのバージョンも含まれます。

シナリオpdi/pdi8ac44/pdiac.scnは、この新しい正規化機能を使用します。ソフトウェアモデルは**AddOn**コレクションにある**InventModel**リンクにマップされます。各不明インベントリモデルは、ソフトウェアのインストールテーブル（amSoftInstall）に記録され、一時モデルsysUNKNOWN_SOFTにリンクされるため、正規化提案が作成されます。正規化提案は、ウィザードを使用して処理します。このようにして一時モデルを、AssetCenterにある既存の最終的モデルとを関連付けます。正規化処理が完了したら、更新はコンピュータにリンクされているすべてのインベントリされたアプリケーションに伝播します。

注意:

最終モデルが**sysUNKNOWN_SOFT**をリセットしないよう、更新を実行する際、**PifIgnoreSubDocumentReconc**関数を含む照合更新スクリプトを作成する必要があります。

照合更新キャッシュ

照合更新キャッシュにより、データベースクエリの回数が削減されます。キャッシュはメモリに格納されます。

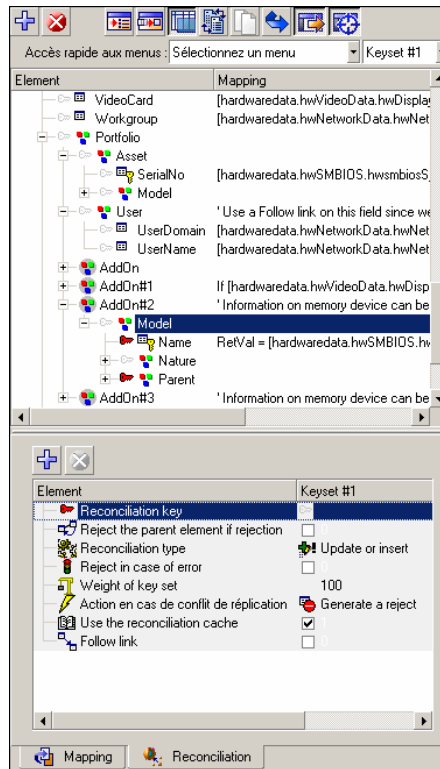
照合更新キャッシュが有用となる状況を以下に挙げます。

- マッピングに定数が使用されている場合
- モデル (amModel) テーブルや属性 (amnature) テーブルなど、比較的小さいテーブルに関するマッピングの場合
- マッピングによってテーブルが更新されない場合

レコードが照合更新キャッシュに見つかり、ドキュメントの内容が変更されていないことが推測され、結果としてこのレコードは更新されません。

ドキュメントの最大数に達するとキャッシュは削除されます。この最大数はコネクタオプションで定義します (**Edit/ Options/ Connector/ Maximum number of documents in the reconciliation cache**メニュー)。

照合更新キャッシュは、OpenView Desktop Inventoryなどのインベントリツールと併用すると効力を発揮します。



シナリオpdiac.scnの**inventorySrc-amComputerDst**マッピングで、照合更新キャッシュは**モデル**構造に使用されます。AssetCenterモデル自体は変更しないままで、関連情報（この場合は資産）が変更されることを仮定しています。

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を以下に挙げます。

- Action Request Systemシナリオ [献 374]
- Asset Insightシナリオ [献 375]
- Asset Managementシナリオ [献 376]
- Enterprise Discoveryシナリオ [献 379]
- Eメールシナリオ [献 377]
- iInventoryシナリオ [献 385]
- OpenView Configuration Managementシナリオ [献 388]
- LANDeskシナリオ [献 386]
- LDAPシナリオ [献 398]
- OpenView Network Discoveryシナリオ [献 405]
- MQSeriesシナリオ [献 396]
- NTセキュリティシナリオ [献 403]
- Decision Centerシナリオ [献 409]
- OpenView Desktop Inventoryシナリオ [献 406]
- SAP BAPIシナリオ [献 409]
- SAP BAPIシナリオ [献 412]
- SMSシナリオ [献 414]
- ServiceCenterシナリオ [献 416]
- TS Censusシナリオ [献 427]
- Tivoliシナリオ [献 429]
- Unicenter AMOシナリオ [献 432]
- Webサービスシナリオ [献 433]
- Winpark Actimaシナリオ [献 433]

Desktop Administration 6.x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、Desktop Administration 6.xコネクタについて説明します。

Desktop Administration 6.xコネクタについて

Desktop Administration 6.xで、Desktop Administration 6.xデータベースに含まれるデータを処理できます。

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。

Desktop Administration 6.xアプリケーションクライアントを、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する必要があります。

Desktop Administration 6.xの設定に関する注意事項

amdb.iniファイル (Desktop Administration 4.0.1より前のバージョン)

このファイルは、Desktop Administration 6.x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、Desktop Administration 6.x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.

Desktop Administration 6.xコネクタの設定を行う際、Connect-Itはこのファイルを読み込み、利用可能な接続のリストを提示します。使用しているコンピュータがDesktop Administration 6.xアプリケーションにアクセスできない場合、Desktop Administration 6.xコネクタは使用できません。

amdb.iniファイル (6.0.1以降のDesktop Administration)

バージョン6.0.1以降のDesktop Administrationでは、コンピュータで使用可能な接続が複数のamdb.iniファイル内で宣言されています。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

Desktop Administration 6.xデータベースの必須フィールド

Desktop Administration 6.xアプリケーションでは、フィールドやリンクをデフォルトで必須とすることができ、Desktop Administration 6.xアプリケーションの管理者がカスタマイズできます。

照合更新時に、Desktop Administration 6.xコネクタが発行する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。

API : DLL

Desktop Administration 6.xアプリケーションの完全インストールまたはカスタムインストールを実行する際、APIパッケージを選択すると、Desktop Administration 6.xコネクタを容易に使用できるようになります。

Desktop Administration 6.xの既知の制約点

Desktop Administration 6.x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。

Desktop Administration 6.xコネクタとその他のコネクタ間のシナリオで、この基準を満たさない日付（マッピングスクリプト中で使用されている日付）が使用されていると、シナリオの実行中にエラーメッセージが表示されます。

Desktop Administration 6.xコネクタの設定

ウィザードでコネクタを設定できます。コネクタの詳細については、「コネクタの設定 [献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.2. Desktop Administration 6.xコネクタの設定

 Name and describe the connector [献 16]
 Select a connection type [献 18]
 Configuring the connection [献 19]
 Advanced configuration (詳細モード) [献 74]
 Configure the reconnection parameters (詳細モード) [献 29]
 Determine server delay (詳細モード) [献 31]
 Configure schedule pointers (詳細モード) [献 33]
 Manage transactions (詳細モード) [献 35]
 Configure the cache (詳細モード) [献 36]
 Define document types (詳細モード) [献 39]
 Parallelize consumption (詳細モード) [献 41]

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「Name and describe the connector [献 16]」の節を参照してください。

Select a connection type

このページの使用方法については、「コネクタの設定 [献 15]」章の「Select a connection type [献 18]」の節を参照してください。

Configuring the connection

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Configuring the connection \[献 19\]](#)」の節を参照してください。

Advanced configuration (詳細モード)



警告:

このページが表示されるのは、オプション**Display advanced configuration wizards**を選択した場合のみです（**Edit**の**Options**メニュー）。

Dynamic library

このオプションで、**Desktop Administration 6.x**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。

Expose the primary key of the tables

このオプションを使用すると、**Desktop Administration 6.x**コネクタが発行するドキュメントタイプに、データベーステーブルの主キーに対応するフィールドが含まれるようになります。

Expose the foreign keys of the tables

このオプションを選択すると、発行ドキュメントタイプには、データベーステーブルの外部キーに対応するフィールドが含まれます。

Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Configure the reconnection parameters \(詳細モード\) \[献 29\]](#)」の節を参照してください。

Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Determine server delay \(詳細モード\) \[献 31\]](#)」の節を参照してください。

Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Configure schedule pointers \(詳細モード\) \[献 33\]](#)」の節を参照してください。

Manage transactions (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Manage transactions \(詳細モード\) \[献 35\]](#)」の節を参照してください。

Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the cache (詳細モード) [献 36]」の節を参照してください。

Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Advanced options (詳細モード) [献 37]」の節を参照してください。

Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Define document types (詳細モード) [献 39]」の節を参照してください。

Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Parallelize consumption (詳細モード) [献 41]」の節を参照してください。

Desktop Administration 6.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 (ディレクトティブ) [献 43]」章の「生成用ルール [献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。これらの句の詳細については「コネクタのルール (ディレクトティブ) [献 43]」章の「生成用ルール [献 44]」節、「**WHERE**句と**ORDER BY**句 [献 45]」を参照してください。

AQLシンタックスの詳細については、付録「**AQL**クエリ [献 435]」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)

オペランドのタイプ	例
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で記述するWHERE句では、**sysdate**フィールドを**getdate()**関数に置換する必要があります。

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。

Desktop Administration 6.x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） [献 43]」章の「取り込み用ルール [献 46]」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、**Reconciliation**と**Advanced reconciliation**タブ内で照合更新用のパラメータを入力します。

これら照合更新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） [献 43]」章の「取り込み用ルール [献 46]」節内「照合更新 [献 47]」を参照してください。

付属シナリオ - Desktop Administration 6.xコネクタ

このコネクタ用のシナリオはありません。

Enterprise Discoveryコネクタ

ベースコネクタ

■ 生成（ソース）

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

Enterprise Discoveryツールのカスタマイズ方法と使用方法の補足情報については、別マニュアルを参照してください。

Enterprise Discoveryコネクタで、Network Discoveryデータベースを処理できます。

Enterprise Discoveryデータベースには、企業全体のITポートフォリオに関する情報が含まれています。データベースに格納されている各コンピュータに対し










て、付属シナリオはこれらのレコードに関するデータをAssetCenterやServiceCenterのデータベースに転送します。

Enterprise 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 [献 15]」の章の「コネクタの設定 [献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.3. Enterprise Discoveryコネクタの設定

 Name and describe the connector [献 16]
 Select a connection type [献 18]
 Configuring the connection [献 19]
 Advanced configuration (詳細モード) [献 27]
 Configure the reconnection parameters (詳細モード) [献 29]
 Determine server delay (詳細モード) [献 31]
 Configure schedule pointers (詳細モード) [献 33]
 Configure the cache (詳細モード) [献 36]
 Define document types (詳細モード) [献 39]

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「Name and describe the connector [献 16]」の節を参照してください。

Select a connection type

このページの使用方法については、「コネクタの設定 [献 15]」章の「Select a connection type [献 18]」の節を参照してください。

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。

Configuring the connection

- **Server**フィールド：サーバへの接続に使用するポートを指定します。

Connect-ItがEnterprise Discoveryサーバ以外のワークステーション上にインストールされている場合に、このフィールドにEnterprise DiscoveryサーバのDNS名やIPアドレスを入力します。DNS名やIPアドレスはコロンの前に配置します。以下に例を挙げます。

server.company.com:8108

127.0.0.1:8108

- **User**フィールド：ログインを指定します。すべての権限を持つ`admin`ログインを使用することを推奨します。使用するログインは、必須の読み取りおよび書き込み権限を持つ必要があります。ログインプロファイルはEnterprise Discovery Webユーザインタフェースで定義します。
- **Password**フィールド：このログイン名のパスワードを指定します。
- **Database**フィールド：使用するデータベース名を指定します。デフォルト名は`Aggregate`です。

Advanced configuration（詳細モード）

このページの使用方法については、「コネクタの設定 [献 15]」章の「Advanced configuration（詳細モード） [献 27]」の節を参照してください。

Configure the reconnection parameters（詳細モード）

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the reconnection parameters（詳細モード） [献 29]」の節を参照してください。

Determine server delay（詳細モード）

このページの使用方法については、「コネクタの設定 [献 15]」章の「Determine server delay（詳細モード） [献 31]」の節を参照してください。



注意:

クエリで、アプリケーションに含まれ、編集不可能な現在の日時を取得できません。

Configure schedule pointers（詳細モード）

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure schedule pointers（詳細モード） [献 33]」の節を参照してください。

Configure the cache（詳細モード）

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the cache（詳細モード） [献 36]」の節を参照してください。

Define document types（詳細モード）

使用しているEnterprise Discoveryのバージョンに応じて、拡張ファイルを入力します。

このページの使用方法については、「コネクタの設定 [献 15]」章の「Define document types（詳細モード） [献 39]」の節を参照してください。

発行ドキュメントタイプ

Enterprise Discoveryコネクタが発行するドキュメントタイプを以下に挙げます。

- **Devices**ドキュメントタイプ。このドキュメントタイプは、Enterprise Discoveryデータベースに含まれているインベントリ情報を発行します。
- **DeviceStructure**ドキュメントタイプ。このドキュメントタイプは、筐体やモジュールなど、個別の追跡が必要となるネットワーク要素のサブコンポーネントに関するインベントリ情報を発行します。これらのサブコンポーネントは、シリアル番号を持ち、AssetCenterで**固有資産タグ**によって管理される必要があります。
 - ▶ Connect-Itマニュアル、『AssetCenterデータベース統合ソリューション』、「AssetCenter Software - 一般概念」

付属シナリオ - Enterprise Discoveryコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- Enterprise Discoveryシナリオ [献 379]

OpenView Desktop Inventory (PDI) コネクタ

ベースコネクタ

- 生成 (ソース)

OpenView Desktop Inventory 7.xコネクタについて

PDIコネクタは、OpenView Desktop Inventoryのインベントリで取得されたXMLファイルを処理します。これらのファイルは、Connect-Itデータキット付属のGZファイルで圧縮できます（[Connect-Itインストール先フォルダ]\datakit\pdi*.xml.gz）。

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 [献 15]」章の「コネクタの設定 [献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.4. PDIコネクタの設定

 Name and describe the connector [献 16]

 Selecting a connection protocol [献 80]

FTP server [献 80]	Local or network file(s) [献 81]
Select files or folders [献 81]	
Define post-processing actions [献 82]	
Choose a DTD/XSD [献 83]	
Configure the JVM [献 83]	

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、設定ウィザードでファイルやフォルダのパスを指定しなくてはいけない場合があります。この場合、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。

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「Name and describe the connector [献 16]」の節を参照してください。

Selecting a connection protocol

以下のオプションがあります。

- FTP server
- Local or network file(s)

FTP server

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 server connection

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。

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

- **Connect in passive mode**
このオプションを選択して、パッシブモードでサーバに接続します。
- **Test the connection**
このボタンで、接続の有効性をテストできます。

 **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、

Select files or folders

このページの使用方法については、「コネクタの設定 [献 15]」章の「Select files or folders [献 25]」の節を参照してください。

Define post-processing actions

PDIコネクタがファイルを読み取ったあとの操作として、Connect-Itには以下の3つのオプションがあります。

- **Leave it in the folder**
- **Delete it from the folder**
- **Move it to another folder**
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。

Local or network file(s)

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

Choose a file or folder

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。以下のオプションから選択できます。

- **Read a file**
- **Read files in a folder**

Read a file

このオプションを選択すると、ウィザードで、コンピュータやネットワーク上に存在するテキストファイルに対するパスの入力が求められます。

Read files in a folder

このオプションを選択すると、ウィザードで、以下のフィールドへの入力が求められます。

- **Folder name**

ファイルを含むフォルダのパスを入力します。

- **Extension**

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、**OpenView 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。

Read the sub-folders

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。

Select files or folders

このページの使用方法については、「コネクタの設定 [献 15]」章の「**Select files or folders** [献 25]」の節を参照してください。

Define post-processing actions

コネクタがファイルを読み取ったあとの操作として、**Connect-It** には以下の3つのオプションがあります。

- **Leave it in the folder**

- **Delete it from the folder**

- **Move it to another folder**

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。

処理後のアクションの使用

処理後のアクションを使用するには：

- ソースコネクタが生成したドキュメントタイプの *UrlFileInfo.Path* 構造体と *SuccessReport* ドキュメントタイプの *UrlFileInfo.Path* 構造体との間にマッピングを作成します。

Choose a DTD/XSD

OpenView Desktop Inventory スキャンに対応するXMLファイル进行处理するには、コネクタはDTDまたはXSDを使用する必要があります。

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。

Configure the JVM

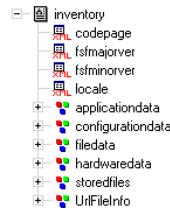
このページが表示されるのは、設定ウィザードの **Choose a DTD/XSD** [献 87] ページで XSD を選択している場合のみです。

このページの使用方法については、「コネクタの設定 [献 15]」章の「**Configure the cache** (詳細モード) [献 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 3.3. PDIコネクタ - 生成用ドキュメントタイプ



OpenView Desktop Inventoryコネクタのスケジュール

AssetCenter Server では、Desiktop Inventory が取得した.xml.gz ファイルの取得をスケジュールできます。

シナリオのスケジュールリングについては、AssetCenter 『管理』の「AssetCenter Server」の章、「AssetCenter Server でモニタするモジュールを設定する」を参照してください。

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を以下に挙げます。

- OpenView Desktop Inventoryシナリオ [献 406]

OpenView Network Inventory 8.xコネクタ (PDI)

ベースコネクタ

- 生成 (ソース)
-

OpenView Network Inventory 8.xコネクタについて

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 [献 15]」章の「コネクタの設定 [献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.5. PDIコネクタの設定

■ Name and describe the connector [献 16]
■ Selecting a connection protocol [献 80]
■ FTP server [献 80] ■ Local or network file(s) [献 81]
■ Select files or folders [献 82]
■ Define post-processing actions [献 82]
■ Choose a DTD/XSD [献 83]
■ Configure the JVM [献 88]



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、設定ウィザードでファイルやフォルダのパスを指定しなくてはいけない場合があります。この場合、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。

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「Name and describe the connector [献 16]」の節を参照してください。

Selecting a connection protocol

以下のオプションがあります。

- FTP server
- Local or network file(s)

FTP server

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 server connection

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。

- Server
FTPサーバ名を指定します。
- Login
サイトへのアクセスを許可するログインを入力します。
- Password
ログイン用のパスワードを入力します。
- Connect in passive mode
このオプションを選択して、パッシブモードでサーバに接続します。
- Test the connection
このボタンで、接続の有効性をテストできます。



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、

Select files or folders

このページの使用方法については、「コネクタの設定 [献 15]」章の「Select files or folders [献 25]」の節を参照してください。

Define post-processing actions

PDIコネクタがファイルを読み取ったあとの操作として、Connect-Itには以下の3つのオプションがあります。

- Leave it in the folder
 - Delete it from the folder
 - Move it to another folder
- 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。

Local or network file(s)

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

Choose a file or folder

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。以下のオプションから選択できます。

- **Read a file**
- **Read files in a folder**

Read a file

このオプションを選択すると、ウィザードで、コンピュータやネットワーク上に存在するテキストファイルに対するパスの入力が求められます。

Read files in a folder

このオプションを選択すると、ウィザードで、以下のフィールドへの入力が求められます。

- **Folder name**

ファイルを含むフォルダのパスを入力します。

- **Extension**

コネクタが読み込む必要があるファイル名の拡張子を入力します。デフォルトではこのフィールドの値は、`gz;xsf`です。このGZファイルは、OpenView Network Inventory スキャンに対応するXMLファイルを含む圧縮ファイルです。

Read the sub-folders

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。

Select files or folders

このページの使用方法については、「コネクタの設定 [献 15]」章の「Select files or folders [献 25]」の節を参照してください。

Define post-processing actions

コネクタがファイルを読み取ったあとの操作として、Connect-It には以下の3つのオプションがあります。

- **Leave it in the folder**
- **Delete it from the folder**
- **Move it to another folder**

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。

処理後のアクションの使用

処理後のアクションを使用するには：

- ソースコネクタが生成したドキュメントタイプの *UrlFileInfo.Path* 構造体と *SuccessReport* ドキュメントタイプの *UrlFileInfo.Path* 構造体との間にマッピングを作成します。

Choose a DTD/XSD

OpenView Network Inventory スキャンに対応するXMLファイルを処理するには、コネクタはDTDまたはXSDを使用する必要があります。

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。

Configure the JVM

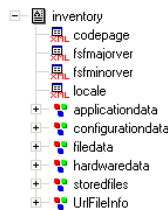
このページが表示されるのは、設定ウィザードのChoose a DTD/XSD [献 87]ページでXSDを選択している場合のみです。

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the JVM [献 40]」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 3.4. PDIコネクタ - 生成用ドキュメントタイプ



OpenView Network Inventoryコネクタのスケジュール

AssetCenter Serverでは、Desiktop Inventoryが取得した.xml.gzファイルの取得をスケジュールできます。

シナリオのスケジュールリングについては、AssetCenter『管理』の「AssetCenter Server」の章、「AssetCenter Serverでモニタするモジュールを設定する」を参照してください。

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を以下に挙げます。

- OpenView Desktop Inventoryシナリオ [献 406]

OpenView Network Inventoryコネクタ

ベースコネクタ

■ 生成 (ソース)

この節では、Network Discoveryコネクタについて説明します。

Network Discoveryコネクタで、Network Discoveryデータベースを処理できます。

Network Discoveryデータベースには、会社のコンピュータの集合に関する情報が含まれます。シナリオは、このデータベースにあるコンピュータ関連の情報をAssetCenterデータベースやServiceCenterデータベースに転送します。

OpenView Network Inventory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 [献 15]」章の「コネクタの設定 [献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.6. OpenView Network Inventoryコネクタの設定

 Name and describe the connector [献 16]

 Select a connection type [献 18]

 Configuring the connection [献 19]

 Advanced configuration (詳細モード) [献 27]

 Configure the reconnection parameters (詳細モード) [献 29]

 Determine server delay (詳細モード) [献 31]

 Configure schedule pointers (詳細モード) [献 33]

 Configure the cache (詳細モード) [献 36]

 Advanced options (詳細モード) [献 37]

 Define document types (詳細モード) [献 39]

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「Name and describe the connector [献 16]」の節を参照してください。

Select a connection type

このページの使用方法については、「コネクタの設定 [献 15]」章の「Select a connection type [献 18]」の節を参照してください。

Configuring the connection

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configuring the connection [献 19]」の節を参照してください。

Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Advanced configuration (詳細モード) [献 27]」の節を参照してください。

Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the reconnection parameters (詳細モード) [献 29]」の節を参照してください。

Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Determine server delay (詳細モード) [献 31]」の節を参照してください。

注意:

クエリで、アプリケーションに含まれ、編集不可能な現在の日時を取得できません。

Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure schedule pointers (詳細モード) [献 33]」の節を参照してください。

注意:

PNDサーバの現在の日付は、デフォルトで上限として使用されます。

Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the cache (詳細モード) [献 36]」の節を参照してください。

Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Advanced options (詳細モード) [献 37]」の節を参照してください。

Define document types (詳細モード)

OpenView Network Inventoryのバージョンに合わせて拡張ファイルに入力します。

- バージョン5.1のConnect-Itフォルダにあるconfig/pnd/pnddb51.cfgファイルを指定します
- バージョン5.2のConnect-Itフォルダにあるconfig/pnd/pnddb52.cfgファイルを指定します

このページの使用方法については、「コネクタの設定 [献 15]」章の「Define document types (詳細モード) [献 39]」の節を参照してください。

発行ドキュメントタイプ

OpenView Network Inventoryコネクタは単一ドキュメントタイプ**Devices**を発行します。このドキュメントタイプは、OpenView Network Inventoryデータベースに含まれるインベントリ情報を発行します。

付属シナリオ - OpenView Network Inventoryコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- OpenView Network Discoveryシナリオ [献 405]

OpenView Operationsコネクタ

OpenView Operations UnixコネクタはConnect-Itにのみ用意されています。しかし、区切りテキストコネクタで動作するOpenView Operation Windowsのシナリオが利用できます。

ベースコネクタ

- 読み取り


このコネクタは、HP Software UNIXアプリケーションの以下の主要テーブルを公開します。

- OPC_NODES
- OPC_ACT_MESSAGES
- OPC_HIST_MESSAGES

OpenView Operations Unixコネクタの設定

 Name and describe the connector [献 92]

 Select connection type [献 92]

 Configure the database server connection [献 92]

 Advanced configuration [献 92]

 [Configure the reconnection parameters \[献 92\]](#)

 [Determine the time difference \[献 92\]](#)

 [Configure schedule pointers \[献 92\]](#)

 [Configure the cache \[献 92\]](#)

 [Advanced options \[献 93\]](#)

 [Define document types \[献 93\]](#)

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「Name and describe the connector [献 16]」の節を参照してください。

Select connection type

OracleクライアントはConnect-Itアプリケーションサーバ上にインストールされている必要があります。

このページの使用方法については、「Select a connection type [献 18]」章の「Select a connection type [献 18]」の節を参照してください。

Configure the database server connection

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configuring the connection [献 19]」の節を参照してください。

Advanced configuration

このページの使用方法については、「コネクタの設定 [献 15]」章の「Advanced configuration (詳細モード) [献 27]」の節を参照してください。

Configure the reconnection parameters

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the reconnection parameters (詳細モード) [献 29]」の節を参照してください。

Determine the time difference

このページの使用方法については、「コネクタの設定 [献 15]」章の「Determine server delay (詳細モード) [献 31]」の節を参照してください。

Configure schedule pointers

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure schedule pointers (詳細モード) [献 33]」の節を参照してください。

Configure the cache

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the cache (詳細モード) [献 36]」の節を参照してください。

Advanced options

このページの使用方法については、「コネクタの設定 [献 15]」章の「Advanced options（詳細モード） [献 37]」の節を参照してください。

Define document types

このページの使用方法については、「コネクタの設定 [献 15]」章の「Define document types（詳細モード） [献 39]」の節を参照してください。

発行ドキュメントタイプ

発行ドキュメントタイプを以下に挙げます。

- Node : OPC_NODESテーブルの解釈
- ActiveMessage : OPC_ACT_MESSAGESテーブルの解釈
- MessageHistory : OPC_HIST_MESSAGESテーブルの解釈

付属シナリオ

- ovou82ac50/ovac.scn
このシナリオで、HP Software Operationsに格納されているサーバ（ノード）関連情報を取得し、それをAssetCenterに挿入できます。
- ovow75ac50/ovowac.scn
このシナリオは、
 - WMI経由でノードのリストをエクスポートします
 - AssetCenterにノードのリストを挿入します
- ovou82sc62/ovsc.scn
このシナリオで、HP Software Operationsに格納されているサーバ（ノード）関連情報を取得し、それをServiceCenterに挿入できます。
- ovow75sc62/ovowsc.scn
このシナリオは、
 - WMI経由でノードのリストをエクスポートします
 - ServiceCenterにノードのリストを挿入します

重要項目:

正しく機能するには、シナリオはOVO Windowsサーバ上に配置する必要があります。

このシナリオは、次のように機能します。Visual Basicがサーバ（ノード）のリストをWMI経由で取得します。サーバのリストを取得したら、それはタブ区切りテキストファイルとして保存されます。

スクリプトが受け付けるのが1パラメータのみの場合：エクスポート対象のデータを含むファイル名。

WMI経由で区切りテキストコネクタが取得するノードフィールドを以下に挙げます。

WMIフィールド	区切りテキストファイルの列	コメント
キャプション	キャプション	ノードの説明
PrimaryNodeName	PrimaryNodeName	ホスト名（ドメイン名でサフィックス可能）
OSType	OSType	OSTypeはWMIの番号を返します。対応するテキスト地はスクリプトを使用して決定されます
OSVersion	OSVersion	OSバージョン/ビルド番号
CommunicationPath	CommunicationPath	通常、ホスト名+ドメイン、またはIPアドレス
Name	Name	OVO Windowsでの識別子
SystemType	SystemType	SystemTypeはWMIの番号を返します。対応するテキスト地はスクリプトを使用して決定されます

スクリプトが生成するファイルの解析

スクリプトが生成するファイルは、テキスト区切りファイルタイプです。このファイルは、データをアプリケーションに挿入するため、Connect-Itシナリオによって解析されます。

.dscファイルはファイルの解析用に用意されています。区切りテキストコネクタが公開するドキュメントは、**node**と呼ばれます。

表 3.7. node-->amcomputer

ターゲット	マッピング/ソース
Name	[PrimaryNodeName]
ComputerDesc	[Caption]
OperatingSystem	[OSType]
OSBuildNumber	[OSVersion]
TcpIpAddress	[CommunicationPath]
TcpIpHostName	[PrimaryNodeName]
Portfolio	
Folder	[Name]
Portfolio.Asset	
Portfolio.Asset.Model	
Name	
Portfolio.Asset.Model.Nature	
"CPU" Code	



注意:

太字のフィールド (Name、Code) は照合更新キーを持ちます。

BizDoc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)

BizDocコネクタで、HP Software Webクライアントアプリケーションサーバとの対話を可能にします。

BizDocの既知の制約点

エラーは、ローカルにはではなく、サーバレベルでログファイルに保存されます。

BizDocコネクタ (読み取り) 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 [献 15]」章の「コネクタの設定 [献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.8. BizDocコネクタ (読み取り) の設定

Name and describe the connector [献 16]
Choosing a processing mode [献 95]
Configure BizDoc server connection [献 96]
Choose a DTD/XSD [献 97]
Determine server delay (詳細モード) [献 97]
Configure schedule pointers (詳細モード) [献 33]

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「Name and describe the connector [献 16]」の節を参照してください。

Choosing a processing mode

このページでは、コネクタを読み取りモードまたは書き込みモードのどちらで使用するのかを指定できます。

Configure BizDoc server connection

このページでは、以下のHTTP接続パラメータを指定する必要があります。

- Address
[プロトコル]:/[アドレス]:[ポート]/[パス]のように、タイプの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。
- Login
サイトへのアクセスを許可するログインを入力します。
- Password
ログイン用のパスワードを入力します。

Secured connection (HTTPS protocol)

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 [protocol] の部分がHTTPSプロトコルでない場合でも、**Secured connection (HTTPS protocol)** 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

Address	Secured connection (HTTPS protocol)	Connection mode
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

Script

このフィールドは自動的に入力されます。

.cgiスクリプトは、BizDoc管理者が用意します。

Paging options

Paging sizeおよび**Start of first page**オプションで、BizDocサーバにクエリする際のサブクエリのサイズを定義できます。

Do not verify the server identity

セキュアな接続でのプロキシサーバをサポートするには、このオプションを使用します。

プロキシサーバがインストールされている場合、サーバ証明書に示されているアドレスは、プロキシサーバのアドレスとは異なります。このオプションを無効にして、サーバ識別の自動検証を無効にしてください。

Choose a DTD/XSD

ファイル进行处理するには、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、(*.xsdを使用して) **DTD/XSD** フィールドで使用するDTDやXSDを指定できます。



注意:

DTDはサーバ管理者が用意します。

Determine server delay (詳細モード)

Script フィールドで、使用する.cgiスクリプトを指定し、サーバにローカル時刻を知らせます。

このページの使用方法については、「コネクタの設定 [献 15]」章の「**Determine server delay (詳細モード)** [献 31]」の節を参照してください。

Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「**Configure schedule pointers (詳細モード)** [献 33]」の節を参照してください。






BizDocコネクタ (書き込み) 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 [献 15]」章の「コネクタの設定 [献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.9. BizDocコネクタ (書き込み) の設定

 Name and describe the connector [献 16]

-  [Choosing a processing mode \[献 98\]](#)
 -  [Configure BizDoc server connection \[献 98\]](#)
 -  [Choose a DTD/XSD \[献 98\]](#)
 -  [Determine server delay \(詳細モード\) \[献 31\]](#)
 -  [Parallelize the consumption of the documents \[献 42\]](#)
-

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「Name and describe the connector [献 16]」の節を参照してください。

Choosing a processing mode

このページでは、コネクタを読み取りモードまたは書き込みモードのどちらで使用するのかを指定できます。

Configure BizDoc server connection

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。


- **Address**
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、タイプアドレスを入力します。
- **Login**
サイトへのアクセスを許可するログインを入力します。
- **Password**
ログイン用のパスワードを入力します。

Secured connection (HTTPS protocol)

このオプションの使用方法については、「Secured connection (HTTPS protocol) [献 291]」の節を参照してください。

Choose a DTD/XSD

Select a DTD/XSD ページでは、(*.xsdを使用して) **DTD/XSD** フィールドで使用するDTDやXSDを指定します。BizDocコネクタが発行するドキュメントタイプは、このXSDに対応します。

 をクリックすると、**File location** 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。

注意:

DTDはサーバ管理者が用意します。

Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Determine server delay (詳細モード) [献 31]」の節を参照してください。

Parallelize the consumption of the documents

このページの使用方法については、「コネクタの設定 [献 15]」章の「Parallelize the consumption of the documents [献 42]」の節を参照してください。

発行ドキュメントタイプ

BizDocコネクタは、すべてのドキュメントタイプに対する以下の構造体を発行します。

- **UrlFileInfo**

この構造体の主な使用目的は処理後のアクションです。

UrlFileInfo構造体には以下のフィールドが含まれます。

- **creationdate**

このフィールドは、ドキュメントの作成日に対応します

- **lastmodificationdate**

このフィールドは、ドキュメントの最終更新日に対応します

- **lastmodificationdate**

このフィールドは、ドキュメントの行数に対応します

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。SCAutoリスニングコネクタは、イベントを受信しクライアントの要求を処理できます。

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 [献 15]」章の「コネクタの設定 [献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

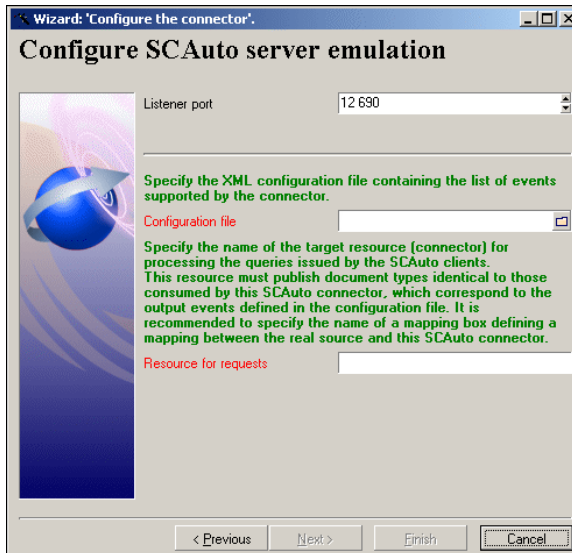
表 3.10. SCAutoリスニングコネクタの設定

 Name and describe the connector [献 16]
 Configure SCAuto server emulation [献 101]
 Configure the JVM [献 102]

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「Name and describe the connector [献 16]」の節を参照してください。

Configure SCAuto server emulation



Wizard: "Configure the connector".

Configure SCAuto server emulation

Listener port: 12690

Specify the XML configuration file containing the list of events supported by the connector.

Configuration file:

Specify the name of the target resource (connector) for processing the queries issued by the SCAuto clients. This resource must publish document types identical to those consumed by this SCAuto connector, which correspond to the output events defined in the configuration file. It is recommended to specify the name of a mapping box defining a mapping between the real source and this SCAuto connector.

Resource for requests:

< Previous Next > Finish Cancel

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が発行する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 Scenario Builder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**)。
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。

Configure the JVM

このページが表示されるのは、設定ウィザードのChoose a DTD/XSD [献 87]ページでXSDを選択している場合のみです。

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the JVM [献 40]」の節を参照してください。

追加情報

SCAAutoリスニングコネクタ設定ファイルを変更する

SCAAuto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*

SCAAuto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、SCAAutoアダプタが要求する要求に対応します。

- *OutputEventTypes*

SCAAuto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、SCAAutoアダプタが生成するイベントに対応します。

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せん。

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の表し方には数通りの方法があります。一つは、単純フィールドのコレクションとして表す方法です (これはscacfg.scnシナリオでのデフォルト動作です)。別な方法では、値がパラグラフである単純Memoタイプフィールドとして表します (各配列要素の値は1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る必要があります。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" "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`end of line` および復帰`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を以下に挙げます。

- `scauto\scacfg\scacfg.scn` [献 424]
- `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。

Decision Centerコネクタ

ベースコネクタ

- 生成
- 取り込み

この節では、Decision Centerコネクタについて説明します。

既知の制約点

コネクタを、同一バージョンレベルのDLLファイルを使用するODBCドライバとODBC Administratorと併せて使用することを強く推奨します。

Decision Center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 [献 15]」章の「コネクタの設定 [献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.11. Decision Centerコネクタの設定

Name and describe the connector [献 16]
Select a connection type [献 18]
Configuring the connection [献 19]
Advanced configuration (詳細モード) [献 27]
Configure the reconnection parameters (詳細モード) [献 29]
Determine server delay (詳細モード) [献 31]
Configure schedule pointers (詳細モード) [献 33]
Manage transactions (詳細モード) [献 35]
Configure the cache (詳細モード) [献 36]
Define document types (詳細モード) [献 39]
Parallelize consumption (詳細モード) [献 41]

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Configuring the connection \[献 19\]](#)」の節を参照してください。

Select a connection type

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Advanced configuration \(詳細モード\) \[献 27\]](#)」の節を参照してください。

Configuring the connection

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Configuring the connection \[献 19\]](#)」の節を参照してください。

Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Advanced configuration \(詳細モード\) \[献 27\]](#)」の節を参照してください。

Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Configure the reconnection parameters \(詳細モード\) \[献 29\]](#)」の節を参照してください。

Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「[Determine server delay \(詳細モード\) \[献 31\]](#)」の節を参照してください。

Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure schedule pointers (詳細モード) [献 33]」の節を参照してください。

Manage transactions (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Manage transactions (詳細モード) [献 35]」の節を参照してください。

Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Configure the cache (詳細モード) [献 36]」の節を参照してください。

Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Define document types (詳細モード) [献 39]」の節を参照してください。

Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Parallelize consumption (詳細モード) [献 41]」の節を参照してください。

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) [献 43]」章の「生成用ルール [献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。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) [献 43]」章の「生成用ルール [献 44]」節、「WHERE句とORDER BY句 [献 45]」を参照してください。

sysdateフィールド

AQLで記述するWHERE句では、**sysdate**フィールドを**getdate()**関数に置換する必要があります。

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。

AQLシンタックスの詳細については、付録「AQLクエリ [献 435]」を参照してください。

NULL

「数値」型のフィールドが未入力（値がNULL）の場合は、値は"0"に設定されます。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。"Location=0"または"lLocaId=0"などのようにします。

Decision Center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） [献 43]」章の「取り込み用ルール [献 46]」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、**Reconciliation**と**Advanced reconciliation**タブ内で照合更新用のパラメータを入力します。

これら照合更新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） [献 43]」章の「取り込み用ルール [献 46]」節内「照合更新 [献 47]」を参照してください。

発行ドキュメントタイプ

Decision Centerコネクタが発行するドキュメントタイプは、Decision Centerデータベースのテーブルに対応します。

付属シナリオ - Decision Centerコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- Decision Centerシナリオ [献 409]

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コネクタについて

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。

同一シナリオ内で、同一サーバや異なるバージョンのサーバに接続する、複数のServiceCenterコネクタを使用できます。

ServiceCenterの設定に関する注意事項

入カイベントの作成

ServiceCenterコネクタはドキュメントを直接的に取り込むことができません。データの整合性の理由により、ServiceCenterは、レコードの直接的な更新を許可しないためです。代わりに、対応する入カイベントをServiceCenterで作成する必要があります。ドキュメントを取り込む際には、このイベントがイベントキューに配置されます。次にServiceCenterの内部マッピングにより、このイベントの要素がServiceCenterのフィールドにマップされます。

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カ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カ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 入カ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の各付属シナリオのUNLファイルは入手可能です。*Hewlett-Packard Development Company, L.P.*技術サポートまでお問い合わせください。

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。

入カイベントの同期をとる

ServiceCenter内の入カ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。

非同期処理モードを選択すると、ServiceCenterで正しくデータが処理されていることの即時検証が行われなくなります。ドキュメントの処理が保留中であることを示す警告メッセージが発行されます。エラーを確認するには、ServiceCenterにある入力キューを確認します。

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せん。

注意:

非同期モードで作業するとパフォーマンスが改善します（50%以上の改善）。パフォーマンスレベルが適切であれば、同期モードを使用することを推奨します。

同期モードの処理では、警告がドキュメントログに記録されます。このような警告を無効にするには、「非同期モードのイベントへの警告 [献 124]」の節を参照してください。

結合ファイル

ServiceCenterでは、結合ファイルにより、複数テーブルのフィールドを単一画面に表示できます。

たとえば、**Device**および**Computer**テーブルには、**joincomputer**結合ファイルにあるフィールドが含まれます。

このような結合ファイルは設定ウィザード内で宣言され、生成ドキュメントタイプで表示可能です。

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.msg*を「sc.ini」ファイルに書き込みます。
- 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。

sccl32.dll

Connect-Itをアップデートする際、c:\windowsやc:\windows\system32フォルダにあるsccl32.dllファイルを削除してください。

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 [献 15]」章の「コネクタの設定 [献 15]」の節を参照してください。

このコネクタに関する一部のオプションを有効にするには、**Edit/ Options/ Connector**メニューを使用します。詳細については、『ユーザガイド』、「シナリオビルダ」、「コネクタ」を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.12. ServiceCenterコネクタの設定

 Name and describe the connector [献 16]
 Define the connection parameters [献 110]
 Advanced configuration (詳細モード) [献 111]
 Configure the reconnection parameters (詳細モード) [献 29]
 Determine server delay (詳細モード) [献 31]
 Configure schedule pointers (詳細モード) [献 33]
 Configure the cache (詳細モード) [献 36]
 Define document types (詳細モード) [献 39]
 Parallelize consumption (詳細モード) [献 41]

Name and describe the connector

このページの使用方法については、「コネクタの設定 [献 15]」章の「Name and describe the connector [献 16]」の節を参照してください。

Define the connection parameters

Server name : 以下のように名前を入力します。

例 :

[Server name].[ServiceCenterクライアントサーバのポート番号]

SC.hp.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。

Login

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。

Password

ログイン用のパスワードを入力します。

Test

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。

- **Test**をクリックします。
Test the connection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。
- **Close**をクリックして設定ウィザードへ戻ります。

Advanced configuration (詳細モード)

このページでは以下のパラメータを設定できます。

- **List of virtual tables** : このフィールドでは、**joindefs**テーブルで定義されているリンクテーブルを選択して、読み取り専用モードで表示できます。デフォルトでは、*join*で始まるすべてのリンクテーブルが表示されます。セミコロン (;) 区切りの値のリストを入力します。ワイルドカード文字 (*, \$, ?)が使用できます。

注意:

イベントが、構造体コレクションの定義にあるリンクテーブルを参照すると、コネクタの設定ウィザードで指定されていない場合でも、そのテーブルは自動的に読み込まれ（読み込み用に公開され）ます。

- 入力イベントの日時タイプフィールドのフォーマット。入力イベントのフォーマットがサーバと異なる場合にこのオプションを選択してください。
- 出力イベントの日時タイプフィールドのフォーマット。出力イベントのフォーマットがサーバと異なる場合にこのオプションを選択してください。

このページの使用方法については、「コネクタの設定 [献 15]」章の「**Advanced configuration (詳細モード)** [献 27]」の節を参照してください。

Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「**Configure the reconnection parameters (詳細モード)** [献 29]」の節を参照してください。

Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「**Determine server delay (詳細モード)** [献 31]」の節を参照してください。

Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「**Configure schedule pointers (詳細モード)** [献 33]」の節を参照してください。

Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「**Configure the cache (詳細モード)** [献 36]」の節を参照してください。

Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Define document types (詳細モード) [献 39]」の節を参照してください。

Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [献 15]」章の「Parallelize consumption (詳細モード) [献 41]」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

数値

文字

日付

例

1 (シングルクォーテーションマークなし)

"a" (ダブルクォーテーションマーク)

'2002-01-10' (国際標準)

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

取り込みドキュメントのルール

コネクタが取り込むドキュメントタイプの照合更新キーを定義できます。

照合更新キーが定義されていない場合、コネクタのデフォルトの動作が使用されます（イベントによる処理）。

注意:

照合更新キーは第1レベルの要素に対してのみ定義できます。構造体やコレクションに含まれるそれ未満のレベルの要素に照合更新キーを持たせることはできません。

▶ [Advanced Reconciliation](#) タブを使用する [献 53].

動作

照合更新キーを保持するドキュメントタイプを基にイベントが生成されます。

ServiceCenterアプリケーションは、ユーザによる変更が整合性ルールを満たすことを保証します。通常、照合更新キーを保持するフィールドが照合更新手順で使用されます。

通常の場合、イベントの参照キーとして使用されるフィールドは存在します。このフィールドはServiceCenterで定義され、必須です。このフィールドに値が存在しない場合、照合更新キーを構成するフィールドの結合を基にした乱数値が作成されます。



注意:

logical.nameフィールドはイベントの処理に必要です。

照合更新キーの動作

以下の場合が考えられます。

- 必要なイベントフィールドの値が入力されている場合：
 - レコードが存在しない場合：イベント（Insert/UpdateまたはUpdate onlyモード）がトリガされます。
 - レコードが存在する場合：イベント（Insert/UpdateまたはUpdate onlyモード）がトリガされます。
 - レコードが存在してその値が変更されていない場合：イベントはトリガされません。
- 必要なイベントフィールドの値が入力されていない場合：
 - レコードが存在しない場合：イベント（Insert/UpdateまたはUpdate onlyモード）がトリガされます。
 - レコードが存在する場合：イベント（Insert/UpdateまたはUpdate onlyモード）がトリガされます。
 - レコードが存在してその値が変更されていない場合：イベントはトリガされません。
- 必要なイベントフィールドの値が入力されているものの、データベース中の値と一致しない場合：
 - レコードが存在しない場合：イベント（Insert/UpdateまたはUpdate onlyモード）がトリガされます。
キーのあるレコードは更新されません。
 - レコードが存在する場合：イベント（Insert/UpdateまたはUpdate onlyモード）がトリガされます。
キーのあるレコードは更新されません。
 - レコードが存在してその値が変更されていない場合：イベントはトリガされません。
キーのあるレコードは更新されません。

データベースのフィールドの値がレコードの値と同一でないことを示す警告メッセージがConnect-Itログに保存されます。レコードは更新されますが、フィールドの値は更新されません。

照合更新スクリプトの動作

イベントの動作はServiceCenterで定義します。

通常、イベントはレコードの更新と挿入を行えますが、レコードの削除は行えません。レコードを削除するための特定のイベントが存在します。選択したテーブル

ルに応じて照合更新を行う方法です。照合更新にレコードの削除が伴う場合、ServiceCenterで定義された対応イベントが使用されます。

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。

scdb.cfgファイル

ServiceCenterデータベースの記述はscdb.cfgファイルに含まれます。ここでは、このファイルを変更する際、使用するServiceCenterデータベースに対応するファイルを変更する必要があります。

- scdb3.cfg：ServiceCenter 3データベースの場合
- scdb4.cfg：ServiceCenter 4データベースの場合
- scdb5.cfg：ServiceCenter 5データベースの場合
- scdb51.cfg：ServiceCenter 5.1データベースの場合
- scdb6.cfg：ServiceCenter 6データベースの場合

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

この例を実行するには、HP Softwareのサポートから以下の3つのファイル入手する必要があります。

- 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 **Options/Import/Load**メニューを選択します。

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。
 - **Utility programs**タブを選択します。
 - **Event services**をクリックします。
 - **Queues**タブを選択します。
 - **Output event**をクリックします。
 - **PMO**タイプのイベントを作成します。
 - **Add**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で、イベントの削除が可能になります。

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 3.13. 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 ドキュメントタイプ 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照合更新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するようにするには、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り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**コレクションに追加するテンプレートを変更できます。

```

{ 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、マネージャでイベントキューを閲覧する際に **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NTAX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、レコードの照合更新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す。

```
{ 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 (**event pmo**) 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 **Edit/ Options**を選択します。
- 2 **Connector**項目の**ServiceCenter**ノードを開きます。
- 3 **Display a warning for events used in asynchronous mode**オプションを**No**に設定します。
- 4 **OK**をクリックします。

同期処理対非同期処理 - ServiceCenterコネクタ

ServiceCenterコネクタが取り込む各ドキュメントは、ターゲットアプリケーションに送信されるクエリに対応します。

ドキュメントの取り込みに要する時間を改善する方法として、以下の方法から選択できます。

- 同期データ処理です。
前のドキュメントがターゲットアプリケーションで処理されると、各取り込み済みドキュメントが送信されます。
- 非同期データ処理です。
前のドキュメントがターゲットアプリケーションで処理されていない場合でも、各取り込み済みドキュメントが送信されます。

テキストフィールドの解釈

ServiceCenterデータベースでは、長いテキストフィールドは配列の形式で保存されます。**Connect-It**はこれらの配列を解釈し、テキストの行を単一の長いテキストフィールドにまとめます。

local.softwareコンピュータテーブルの値のリストなどのように、これらの配列をコレクションとして解釈する場合、**Connect-It \config\sc**アプリケーションフォルダにあるsc.cfgファイルを変更する必要があります。

- 1 sc.cfgファイルを編集します
- 2 **NotMemo**セクションを検索します

各セクションの最初の行には、処理例外の適用先となるテーブル名が含まれています。このテーブル名には、同一フィールドを含むその他のテーブルも考慮に入れるよう、アスタリスク文字(*)でプレフィックスされています。

例：

```
{ NotMemo
{ *deviceworkstation
'boot.files'
}
{ *pc*
'controlling.software'
'local.software'
'remote.software'
}
}
```

この場合、**PC**テーブルには、コレクションとして処理される3フィールドが含まれています。

- 3 解釈対象としないセクションとフィールドを追加し、保存します。



注意:

使用するシンタックス：フィールドとテーブル名には大文字小文字の区別があります。フィールドやテーブル名にピリオドが含まれている場合、名前全体を単一引用符'(...)'に含める必要があります。

日時フィールドの解釈

ServiceCenterデータベースでは、日時フィールドは配列の形式で保存されます。**Connect-It**はこれらの配列を解釈し、コレクションの形式の日時フィールドにします。

Connect-Itは、日付を含むフィールドと期間を含むフィールドとを区別できないため、日時フィールドは問題となります。

日時フィールドを期間として解釈するには、以下に挙げる例のようにscdbN.cfgファイル（ここで、NはServiceCenterのバージョン番号）を変更する必要があります。

- 1 テーブルやイベントが正しく宣言されていることを確認します。

```
// List of tables/events having a processing to add a link or process blob or
// have duration field
#define LINK_TABLES ocmo, eventregister, contacts, problem, enclapplication,
application, probsummary, incidents, ocml, rmlin, clocks
#define LINK_DEVICES device, deviceparent, computer, joincomputer, displaydevice,
joindisplaydevice, furnishings, joinfurnishings, handhelds, joinhandhelds,
mainframe, joinmainframe, networkcomponents, joinnetworkcomponents, officeelectronics,
joinofficeelectronics, softwarelicense, joinsoftwarelicense, storage, joinstorage,
telecom, jointelecom
#define EVENT_ATTACHMENT ICMcomputer, ICMapplication, ICMdisplaydevice,
ICMfurnishings, ICMhandhelds, ICMmainframe, ICMnetworkcomponents,
ICMofficeelectronics, ICMsoftwarelicense, ICMstorage, ICMtelecom, pmo, pmu, pmc, smin
```

- 2 テーブルやイベントを例外として宣言する場合、以下のスクリプトを追加します。

```
{ STRUCT incidents
MODEIN = 0 // Do not display the table in consumption. Put this line only for a table
MODEOUT = 0 // Do not display an event in production. Put this line only for an event and if this event is not available in production.

{ ATTRIBUTE AllFields
}
}
```

- 3 期間フィールドをDURATIONプロパティで倍長整数として宣言します。

```
{ STRUCT incidents
MODEIN = 0 // Do not display the table in consumption. Put this line only for a table
MODEOUT = 0 // Do not display an event in production. Put this line only for an event and if this event is not available in production.

{ LONG handle.time
DURATION = 1
}

{ ATTRIBUTE AllFields
EXCEPTION = handle.time
```

- ```
}
}
```
- 4 ServiceCenterコネクタが開いている場合、一旦閉じてから再度開きます。  
**handle.time**フィールドが日時フィールドではなく、32ビット倍長整数として表示されます。
  - 5 フィールドが既にマップされている場合、以下の手順を実行する必要があります。
    - 生成モード：ドキュメントタイプを編集し、フィールドを再度追加する前にフィールドを削除します。
    - 生成モード：マッピングを編集し、フィールドを再度追加する前にフィールドをマッピングから削除します。

## 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を以下に挙げます。

- Enterprise Discoveryシナリオ [ 献 379]
- OpenView Configuration Managementシナリオ [ 献 388]
- iInventoryシナリオ [ 献 385]
- LANDeskシナリオ [ 献 386]
- MQSeriesシナリオ [ 献 396]
- OpenView Desktop Inventoryシナリオ [ 献 406]
- OpenView Network Discoveryシナリオ [ 献 341]
- Decision Centerシナリオ [ 献 409]
- ServiceCenterシナリオ [ 献 416]
- Tivoliシナリオ [ 献 429]
- TS Censusシナリオ [ 献 427]

---

## ServiceCenter Webサービスコネクタ

ベースコネクタ

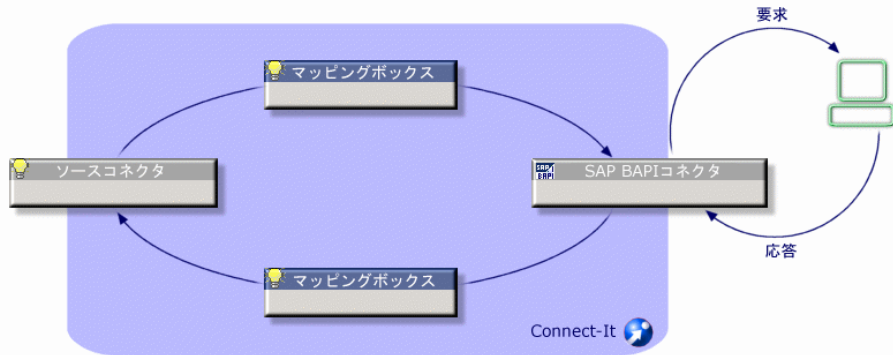
- 取り込み（ターゲット）
- 取り込み - 自発生成（要求 - 応答）

---

ServiceCenter Webサービスコネクタで、ServiceCenter Webサービスと対話できます。

統合シナリオでは、ServiceCenter Webサービスコネクタはドキュメントを取り込み、クエリの形式でWebサービスに送信します。その後、生成されたドキュメントに自動的に変換された応答を受信します。

## 図 3.5. ServiceCenter Webサービスコネクタ - 取り込み - 生成されたドキュメントタイプ



### ServiceCenter 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。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.14. ServiceCenter Webサービスコネクタの設定

|                                                 |
|-------------------------------------------------|
| ■ Name and describe the connector [ 献 16]       |
| ■ Define the connection parameters [ 献 128]     |
| ■ SSL configuration (詳細モード) [ 献 129]            |
| ■ Configuring the proxy server (詳細モード) [ 献 129] |
| ■ Configure the JVM [ 献 130]                    |
| ■ Parallelize consumption [ 献 130]              |

#### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

#### Define the connection parameters

このページを使用して、コネクタの接続パラメータをServiceCenter Webサービスサーバに指定します。

- **Server name** : コンピュータ:ポートのフォーマットでのServiceCenterサーバの名前



- コンテキストパス :

例 :

```
sc62server/ws
```

- Service name : ServiceCenter名 ITILサービス、インシデント管理、構成管理、問題管理、変更管理、サービスデスク、サービスレベル管理
- Login : ServiceCenterログイン
- Password : このログインのパスワード

### SSL configuration (詳細モード)

セキュアな接続を行う場合、**Use an SSL connection**を選択します。

このページでは以下の項目を定義できます。

- セキュア接続用Java証明書ファイル (**CA certificates file (truststore)**)  
デフォルトでConnect-Itは、JREのJava証明書ファイルに関する情報を、`/lib/security/jsse.cacerts`ファイルおよび`/lib/security/cacerts`ファイルから検索します。
- **Truststore password**
- **Truststore type** : ファイルのフォーマットが使用されます。デフォルトで、*JKS*です。
- 秘密キーファイル (**Certificates and client keys file (keystore)**)。秘密キーがデータベースに格納されている場合、このファイルが必要になります。
- 秘密キーファイルのパスワード (**Keystore password**)
- **Keystore type** : ファイルのフォーマットが使用されます。デフォルトで、*JKS*です。

**Verify the server's identity** オプションを有効にする前に、このオプションを有効にする必要があります。

### Configuring the proxy server (詳細モード)

このページで、使用する接続タイプを定義できます。

- **Direct connection**
- **HTTP**  
サーバの名前またはIPアドレス、および必要であればユーザ名とパスワードを入力します。
- **SOCKS**  
サーバの名前またはIPアドレス、および必要であればユーザ名とパスワードを入力します。

HTTPおよびSOCKSタイプ接続の場合、プロキシサーバをバイパスする状況を指定できません (**Do not use a proxy server for** フィールド)。使用する接続に応じて、HTTPまたはSOCKSタイプのアドレスを入力する必要があります。名前はスペースで区切る必要があります。

## Configure the JVM

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

## Parallelize consumption

このページの使用方法については、「コネクタの設定 [ 献 15]」章を参照してください。

## 取り込み用ルール

ドキュメントルートを選択すると、コネクタが取り込む各ドキュメントの**Ignore response failure status**オプションが利用できるようになります。

ServiceCenter Webサービスは次のように機能します。ServiceCenter Webサービスに送信された任意のクエリにより応答が生成されます。この応答には、成功か失敗かを示すステータスが含まれます。

- **Ignore response failure status**オプションが有効である場合、応答は常に生成されます。
- このオプションが有効でない場合、Webサービスからエラーが戻されると応答は生成されず、エラーは取り込まれたドキュメントタイプで保存されます。

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) [ 献 43]」章の「取り込み用ルール [ 献 46]」の節を参照してください。

## 付属シナリオ

このコネクタを使用する付属シナリオを以下に挙げます。

- ServiceCenter Webサービスシナリオ [ 献 425]

---

## OpenView Service Desk (送信)

ベースコネクタ

- 取り込み

---

OpenView Service Desk connector (送信) コネクタの使用目的を以下に挙げます。

- OpenView Service Deskデータベースからのデータの処理。

## OpenView Service Deskコネクタ（送信）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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.15. OpenView Service Deskコネクタ（送信）の設定

|                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------------------------|
|  Name and describe the connector [ 献 16]              |
|  Select a connection type [ 献 18]                     |
|  Configuring the connection [ 献 19]                   |
|  Advanced configuration（詳細モード） [ 献 27]                |
|  Configure the reconnection parameters（詳細モード） [ 献 29] |
|  Determine server delay（詳細モード） [ 献 31]                |
|  Configure schedule pointers（詳細モード） [ 献 33]           |
|  Configure the cache（詳細モード） [ 献 36]                   |
|  Advanced options（詳細モード） [ 献 37]                      |
|  Define document types（詳細モード） [ 献 39]                 |

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

### Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

### Advanced configuration（詳細モード）

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration（詳細モード） [ 献 27]」の節を参照してください。

- **SQL92 supported : AssetCenter Database** ODBCドライバを使用する場合、このオプションを無効にしてください。
- **Execute an initial import** : このオプションを選択すると、データベースにレコードがない場合や挿入する要素がまだ存在しない場合といった初期インポートの場合に、照合更新スクリプトでSELECTクエリを使用しないで済ませることができます。これによりレコードの処理時間を短縮します。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Manage transactions (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Manage transactions (詳細モード) [ 献 35]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## 付属シナリオ - OpenView Service Desk (送信) コネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- sd45ac44\acsd\acsd.scn
- sd45ac44\sdac\sdac.scn
- sd45ac50\acsd\acsd.scn
- sd45ac50\sdac\sdac.scn
- sd50ac44\acsd\acsd.scn
- sd50ac44\sdac\sdac.scn
- sd50ac50\acsd\acsd.scn
- sd50ac50\sdac\sdac.scn

これらのシナリオには、独立したマニュアルがあります。

『*SDAC\_Integration\_Userguide*』を参照してください。

---

## OpenView Service Desk（受信）

ベースコネクタ

■ 生成

OpenView Service Desk connector（受信）コネクタの使用目的を以下に挙げます。

- OpenView Service Deskアプリケーションの.xmlファイルの処理。

### OpenView Service Deskコネクタ（受信）の設定

ウィザードを使用してコネクタを設定できます。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 3.16. OpenView Service Deskコネクタ（受信）の設定

 Name and describe the connector [ 献 16]

 Select a connection protocol [ 献 133]

 Select files or folders [ 献 133]

 Behavior between two sessions [ 献 134]

 Choose a DTD/XSD [ 献 135]

---

#### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

#### Select a connection protocol

このページで、コネクタで使用される接続プロトコルを定義できます。

このコネクタで使用するプロトコルは、デフォルトで定義されています。

#### Select files or folders

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。

コネクタの動作を定義する前に、ファイルの書き込み先となるフォルダ（**Folder name** フィールド）を指定します。このフォルダのパスは、*UrlFileInfo.Path* 要素と組み合わせてマッピングで使用されます。

以下の動作から選択できます。

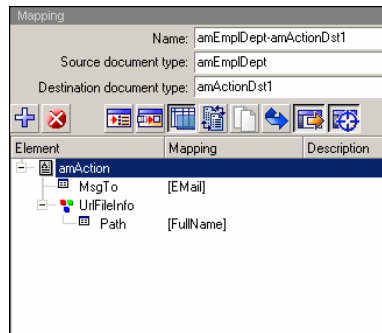
- Write documents to one single file

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。

- Write to a different file for each document

ファイルの書き込み先フォルダ名を指定します(**Folder name** フィールド)。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モード [ 献 310]」を参照してください)。

*UrlFileInfo.Path*要素をマッピングすることで、これらのファイルの名前を定義することもできます。例えば、AssetCenterの各従業員の名前を持つファイル名を作成するなどです。この場合、ウィザードページで定義するパスは、マッピングで定義するパスと連結します。



マッピングで定義したパスをデフォルトパスとして使用し、ウィザードで定義したパスを使用しないためには、/文字をマッピングで使用します。

 **注意:**

この動作により、コネクタ設定ウィザードで有効なパスを指定できなくなります。

*UrlFileInfo.Path*要素の指定されたパスが存在しない場合、以下の条件の下でパスが自動的に作成されます。

- パスは、設定ウィザードで指定されているパスの一部を使用します。
- パスは、指定されたツリーに2レベル以上は追加しません。設定ウィザードでa/b/cというパスが指定されている場合、a/b/c/d/filenameというパスのみ、マッピングで指定できます。

### Behavior between two sessions

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。

### Append to the same file

コネクタは、前回のセッションが中断した箇所から、再度開いたファイルにデータを書き込みはじめます。

### Overwrite the last file

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。


### Number the different files

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。

## Choose a DTD/XSD

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コネクタが発行するドキュメントタイプは、このDTDまたはXSDに一致します。

このコネクタにはDTDが用意されていて、デフォルトの値が入力されています。

をクリックすると、**File location**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。

### File location

以下の3つのオプションがあります。

#### ■ Local/Network

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

**Server**、**Login**、**Password**と**Path**フィールドにFTPパラメータを入力します。

**Path**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、**Server**フィールドにftp.mycompany.comを入力し、**Path**フィールドに/myfolder/DTD/mydtd.dtdを入力します。



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

**Server**、**Login**、**Password**と**Path**フィールドにHTTPパラメータを入力します。

**Path**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、**Server**フィールドにftp.mycompany.comを入力し、**Path**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。

### ルート要素の決定

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要素は複数のChapter要素に含まれ、更にChapter要素は1つの要素Bookに含まれます）。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。

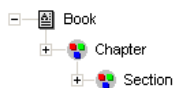
ルート要素（つまり、XMLコネクタに発行される使用可能なドキュメントタイプに当たります）を指定するためには、Connect-Itは以下の2つのオプションを提示します。

- Publish a document type for each root element found in the DTD/XSD（推奨オプション）
- Publish a document type for each root element selected by the user (values are separated by commas)

### Publish a document type for each root element found in the DTD/XSD（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。





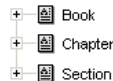
 注意:

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。

## Publish a document type for each root element selected by the user (values are separated by commas)

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、ユーザが選択したルート要素ごとに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。

## OpenView Inventory Manager 4.2iシナリオ

ベースコネクタ

■ 生成（ソース）

OpenView Inventory Managerコネクタの使用目的を以下に挙げます。

- Inventory Managerデータベースからのデータの処理。

## OpenView Inventory Manager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 [ 献 15]」の章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

**表 3.17. OpenView Inventory Managerコネクタの設定**

|                                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------|
|  Name and describe the connector [ 献 16]               |
|  Select a connection type [ 献 18]                      |
|  Configuring the connection [ 献 19]                    |
|  Advanced configuration (詳細モード) [ 献 27]                |
|  Configure the reconnection parameters (詳細モード) [ 献 29] |
|  Determine server delay (詳細モード) [ 献 31]                |
|  Configure schedule pointers (詳細モード) [ 献 33]           |
|  Configure the cache (詳細モード) [ 献 36]                   |
|  Advanced options (詳細モード) [ 献 37]                      |
|  Define document types (詳細モード) [ 献 39]                 |

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

### Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

### Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Manage transactions (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Manage transactions (詳細モード) [ 献 35]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## 付属シナリオ - OpenView Inventory Managerコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- OpenView Configuration Managementシナリオ [ 献 388]

---

## OpenView Service Events v4.2iコネクタ

ベースコネクタ

- 生成 (ソース)

---

OpenView Service Eventsコネクタの使用目的を以下に挙げます。

- Service Eventsデータベースからのデータの処理。

## OpenView Service Events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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードにある各ページを一覧します。

**表 3.18. OpenView Service Eventsコネクタの設定**

---

 Name and describe the connector [ 献 16]

 Select a connection type [ 献 18]

---

|   |                                                                       |
|---|-----------------------------------------------------------------------|
| ■ | <a href="#">Configuring the connection</a> [ 献 19]                    |
| ■ | <a href="#">Advanced configuration</a> (詳細モード) [ 献 27]                |
| ■ | <a href="#">Configure the reconnection parameters</a> (詳細モード) [ 献 29] |
| ■ | <a href="#">Determine server delay</a> (詳細モード) [ 献 31]                |
| ■ | <a href="#">Configure schedule pointers</a> (詳細モード) [ 献 33]           |
| ■ | <a href="#">Configure the cache</a> (詳細モード) [ 献 36]                   |
| ■ | <a href="#">Advanced options</a> (詳細モード) [ 献 37]                      |
| ■ | <a href="#">Define document types</a> (詳細モード) [ 献 39]                 |

## 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

## Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

## Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

## Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

## Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

## Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Manage transactions (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Manage transactions (詳細モード) [ 献 35]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

### Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Parallelize consumption (詳細モード) [ 献 41]」の節を参照してください。

## 付属シナリオ - OpenView Service Eventsコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- OpenView Configuration Managementシナリオ [ 献 388]

---

## OpenView Service Events v4.2iコネクタ

ベースコネクタ

- 生成 (ソース)

OpenView Usage Managerコネクタの使用目的を以下に挙げます。

- Usage Managerデータベースからのデータの処理。

## OpenView Usage Managerコネクタの設定

ウィザードを使用してコネクタを設定できます。「コネクタの設定 [ 献 15]」の章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードにある各ページを一覧します。

表 3.19. OpenView Usage Managerコネクタの設定

|                                                         |
|---------------------------------------------------------|
| ■ Name and describe the connector [ 献 16]               |
| ■ Select a connection type [ 献 18]                      |
| ■ Configuring the connection [ 献 19]                    |
| ■ Advanced configuration (詳細モード) [ 献 27]                |
| ■ Configure the reconnection parameters (詳細モード) [ 献 29] |
| ■ Determine server delay (詳細モード) [ 献 31]                |
| ■ Configure schedule pointers (詳細モード) [ 献 33]           |
| ■ Configure the cache (詳細モード) [ 献 36]                   |
| ■ Advanced options (詳細モード) [ 献 37]                      |
| ■ Define document types (詳細モード) [ 献 39]                 |

## 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

## Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

## Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

## Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

## Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Manage transactions (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Manage transactions (詳細モード) [ 献 35]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

### Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Parallelize consumption (詳細モード) [ 献 41]」の節を参照してください。

## 付属シナリオ - OpenView Usage Managerコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- OpenView Configuration Managementシナリオ [ 献 388]

---

## Management Portal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Management Portalコネクタについて説明します。

## Management Portalコネクタの設定

ウィザードを使用してコネクタを設定できます。「コネクタの設定 [ 献 15]」の章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードにある各ページを一覧します。

表 3.20. Management Portalコネクタの設定

|                                                          |
|----------------------------------------------------------|
| ☰ Name and describe the connector [ 献 144]               |
| ☰ Configure the connection to Management Portal [ 献 144] |
| ☰ Specify the configuration files (詳細モード) [ 献 144]       |
| ☰ Configure the reconnection parameters (詳細モード) [ 献 145] |
| ☰ Configuring the proxy server (詳細モード) [ 献 145]          |
| ☰ Configure the JVM [ 献 145]                             |
| ☰ Parallelize consumption [ 献 145]                       |

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

### Configure the connection to Management Portal

以下のフィールドに入力します。

- URL : サーバのWebサービスのアドレス。

例 :

```
http://RadiaServer:3466/proc/Radia
```

- Login : Radiaツールを使用して暗号化されたユーザ名です。

例 :

```
$ nvdkit.exe
% password encrypt login
{DES}XXXXXXXXX:X
% Base64_Encode
```

- Password : Radiaツールを使用してユーザ名と同じ手順に従って暗号化されたパスワードです。

### Specify the configuration files (詳細モード)

以下のフィールドに入力します。

- XML schema : Management Portal WebサービスXMLスキーマへのパス。
- Properties file : プロパティファイルへのパス。



## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

## Configuring the proxy server (詳細モード)

このページで、使用する接続タイプを定義できます。

- Direct connection
- HTTP

サーバの名前またはIPアドレス、および必要であればユーザ名とパスワードを入力します。

- SOCKS

サーバの名前またはIPアドレス、および必要であればユーザ名とパスワードを入力します。

HTTPおよびSOCKSタイプ接続の場合、プロキシサーバをバイパスする状況を指定できません (**Do not use a proxy server for** フィールド)。使用する接続に応じて、HTTPまたはSOCKSタイプのアドレスを入力する必要があります。名前はスペースで区切る必要があります。

## Configure the JVM

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

## Parallelize consumption

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Parallelize consumption (詳細モード) [ 献 41]」の節を参照してください。

## 付属シナリオ - Management Portalコネクタ

このコネクタには、付属シナリオはありません。



## 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。

---

### Action Request System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)

---

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。

#### 前提条件

##### Action Request Systemコネクタ：ダイナミックライブラリ

Action Request Systemコネクタを使用するには、これらのライブラリのパスが適切な環境変数（Windowsの%PATH%、LinuxとSolarisの\$LD\_LIBRARY\_PATH、AIXの\$LIBPATH）で定義されていることを確認する必要があります。

## 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

- 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す。

- **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置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、**Load Blob from file**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。

#### サポート対象外のフィールドタイプ

- BLOBフィールド (**AR\_DATA\_TYPE\_BYTES**) を含むデータタイプ
- 添付ファイル (**AR\_DATA\_TYPE\_ATTACH**)
- **STAT\_HISTORY**、**VALUE\_SET**、**LOCAL\_VARIABLE**、**QUERY**の各フィールド

## 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。



#### 注意:

このウィザードの一部のページは、詳細モードではアクセスできません。

表 4.1. Action Request Systemコネクタの設定

|                                                                  |
|------------------------------------------------------------------|
| Name and describe the connector [ 献 16]                          |
| Define the connection parameters [ 献 150]                        |
| Advanced configuration (詳細モード) [ 献 27]                           |
| Configure the cache (詳細モード) [ 献 36]                              |
| Configure the reconnection parameters (詳細モード) [ 献 29]            |
| Determine server delay (詳細モード) [ 献 31]                           |
| Configure schedule pointers (詳細モード) [ 献 33]                      |
| List of last-modification fields of the schemas (詳細モード) [ 献 151] |

#### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Define the connection parameters

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。

### Server name

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。

### Connection port

サーバ接続ポート番号を入力します。値を指定しない場合、デフォルト接続ポートが使用されます。

### RPC number

特定サーバを使用する場合、RPC番号を入力します。

### Login

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。

### Password

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。

## Advanced configuration（詳細モード）

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration（詳細モード） [ 献 27]」の節を参照してください。

### Dynamic library to use

このフィールドに、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（arapi45.dll）の完全パスを入力します。バージョン5.0以降のライブラリを使用する必要があります。

### Format of the pointer

このフィールドには、ポインタに適用するユーザフォーマットの名前を入力します。

## Configure the reconnection parameters（詳細モード）

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters（詳細モード） [ 献 29]」の節を参照してください。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### List of last-modification fields of the schemas (詳細モード)

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す。しかし、スキーマとAction Request Systemのバージョンによっては、このフィールド名は異なることがあります。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、このスキーマに対応する、コネクタが発行する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が発行する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 **Diagram**列内のテキストゾーンをクリックし、Action Request Systemコネクタが発行するドキュメントタイプ (スキーマ) の名前を入力します。
- 3 **Field**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。

## 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） [ 献 43]」章の「生成用ルール [ 献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。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） [ 献 43]」章の「生成用ルール [ 献 44]」節、「WHERE句とORDER BY句 [ 献 45]」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

### 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)

## 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） [ 献 43]」章の「取り込み用ルール [ 献 46]」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、**Reconciliation**と**Advanced reconciliation**タブ内で照合更新用のパラメータを入力します。

これら照合更新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） [ 献 43]」章の「取り込み用ルール [ 献 46]」節内「照合更新 [ 献 47]」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コネクタが発行するドキュメントタイプ内のフィールド名

- ARSバージョン5サーバに接続する場合、添付ファイルはサーバ上に直接保存されます。
- Action Request Systemコネクタが発行するドキュメントタイプでは、文字「#」は「\_」に置き換えられます。

例：発行ドキュメントタイプ内で、*Field#1*は*Field\_1*と表示されます。

*champ#1*または*champ\_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。

- WHEREまたはORDERBY句で：
  - フィールドの表示名を使用するには、以下のシンタックスを使用します。

|             |
|-------------|
| DisplayName |
|-------------|

- フィールドのシステム名を指定するには、以下のシンタックスを使用します。

|                |
|----------------|
| [DatabaseName] |
|----------------|

## 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を以下に挙げます。

- `ars\rsm\rsm4ac36\cmpo.scn` [ 献 375]
- `ars\rsm\rsm4ac36\sharedat.scn` [ 献 375]

## TEDW統合シナリオ

TEDW統合シナリオは、Connect-Itのインストールフォルダである `tivoli\tedwExcept` にあります。これらのシナリオについては、Connect-Itの `Doc` フォルダに個別に記載されています。

## ダイナミックリンクライブラリ

Action Request Systemは特定のDLLが必要です。該当するDLLがシステムパスに存在しない場合、エラーメッセージが表示されます。この問題を回避するには、以下に一覧されている.dllファイルをConnect-Itインストールフォルダのbin32フォルダにコピーします。これらのファイルは、通常、Action Request Systemインストールフォルダのサブフォルダに配置されます。

- `arapi45.dll`
- `arrpc45.dll`
- `arutl45.dll`
- `arcatalog_eng.dll`

## 日時の変換

Action Request Systemシナリオをスケジュールモードで実行すると、変換の問題が原因でWHERE句が拒否される場合があります。通常、シナリオを実行するコンピュータと同じ日時構成をAction Request Systemサーバが使用していない場合に、この問題が発生します。この問題を回避するには、日時のユーザフォーマットを定義できます。

- 1 **ARSScheduleTimeStampFormat**というユーザフォーマットを以下の定義で作成します。

```
yyyy' 'mm' 'dd' 'hh':'nn':'ss
```

- 2 ..\config\arsフォルダにあるcondsc.dscファイルを編集します。  
**POINTER\_USER\_FMT=ARSScheduleTimeStampFormat**エントリを追加します。

```
{ CONNECTORDESC
INTERNALNAME=ARS
PARENTINTERNALNAME=Peregrine
....
DB_DESC_FILE=arsdb.cfg
POINTER_USER_FMT=ARSScheduleTimeStampFormat
}
```

- 3 このユーザフォーマットを含むファイルを、Action Request Systemコネクタを使用するすべてのシナリオに追加します。

---

## 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。

特に、WindowsのPATHにはLotus Notesフォルダのパスを含める必要があります。

## 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す。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。
- 添付ファイル








## Lotus Notesコネクタの設定

コネクタを変更する前に、Lotusフォルダにあるvim32.dllファイルへのアクセスパスを、システム環境変数内で宣言する必要があります。

ウィザードでコネクタを設定できます。「コネクタの設定 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 4.2. Lotus Notesコネクタの設定

|                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------|
|  Name and describe the connector [ 献 16]                 |
|  Define the connection parameters [ 献 155]              |
|  Configure the reconnection parameters (詳細モード) [ 献 29] |
|  Configure schedule pointers (詳細モード) [ 献 33]           |
|  Advanced options (詳細モード) [ 献 157]                     |
|  Define document types (詳細モード) [ 献 39]                 |
|  Parallelize consumption (詳細モード) [ 献 41]               |

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Define the connection parameters

**Basic configuration of the connector** 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

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、**Database**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。

### Password

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。

DLLファイルnextpwd.dllによって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。

コネクタ開始時の毎回のパスワード入力を求めないようにするには、以下の手順を実行します。

- 1 Connect-It bin32フォルダをWindowsの環境変数に追加します。
- 2 以下のパラメータをnotes.iniファイルの[Notes]セクション以降に追加します。

```
EXTMGR_ADDINS=nextpwd
```

- 3 変更を保存してConnect-Itを再起動します。



#### 注意:

他のプログラムと競合が発生するようであれば、nextpwd.dllファイルの名前を変更し、前述の方法でnotes.iniファイルを更新します。

### Server name

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、**Local**を選択します。

## Database

Connect-Itシナリオで使用するLotus Notesデータベースを、**Database**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。
- **Server name**フィールドで**Local**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。

## View all database fields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、コネクタが発行するドキュメントタイプ内に現れます。

このオプションは、添付ファイルを管理可能にする**\$FILE**コレクションを公開します。

## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「接続テストのクエリ [ 献 30]」の節を参照してください。

## Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

## Advanced options (詳細モード)

このページで、一部の生産ドキュメントタイプ要素の情報の一部を入力できます。

Lotus Notesテキストフィールドのステータスは、内容に応じて変化します。

- 文字列を含むテキストタイプフィールドは、テキストタイプ属性として解釈されます。
- セミコロンを区切り文字として含む文字列を含むテキストフィールドは、属性のコレクションとして解釈されます。

Lotus Notesコネクタの設定ウィザードのこのページで、属性のコレクションとして公開される属性を識別できます。

- **Name**列には、対象となる属性を識別する（ドキュメント名の追加先となる）Connect-Itでの完全パスが含まれます。
- **Value**列には、単一属性ではなくリストを識別する*NotesList*文字列が含まれます。

## Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Parallelize consumption (詳細モード) [ 献 41]」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

この句で、Lotus Notesデータベースのレコードにフィルタを適用できます。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に対する、以下の句を作成すれば済みます。

```
>= [2000/01/01]
```

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)   |

### Design-form name

Lotus Notesコネクタが発行する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、**Design-form name**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。

**@All**フォームを使用して、各Lotus Notesフォームに関連付けられているデータを表示します。**@All**フォームを使用すると、特定フォームにWHERE句を適用しない場合と同じ結果が得られます。

## 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。

### reconciliationタブの詳細

#### Design-form name

Lotus Notesコネクタが発行する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、**Design-form name**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。

#### Font

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。

#### Reconciliation key

| Element            | Keyset #1 |
|--------------------|-----------|
| Reconciliation key |           |
| Behavior           | Replace   |

このチェックボックスでは、現在選択されている要素が照合更新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照合更新キーとして指定するには：

- 要素を作業枠内で選択します。
- **Reconciliation key** 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。

### Case-sensitive reconciliation

このオプションにより、Connect-Itの照合更新は以下のようになります。

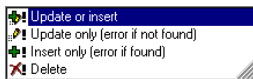
- **Case-sensitive reconciliation** オプションを選択する  
このオプションが選択されていると、照合更新キー値の大文字と小文字は区別して取り扱われます。例：従業員のEメールアドレスを含むフィールドが照合更新キーとして使用されている場合、'jmartin@company.com'、'JMARTIN@company.com' と 'jMartin@company.com' は、ターゲットのLotus Notesデータベース内で3つの別々のレコードになります。
- **Case-sensitive reconciliation** オプションを選択しない  
この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

→ 複数の照合更新キーセットの使用 [ 献 52]

→ 『ユーザガイド』、「Putting a scenario into production (シナリオの本番投入)」の章の「Reconciliation keys (照合更新キー)」の節。

### Reconciliation type



**Reconciliation type** フィールドでは、各親ノードとドキュメントタイプ間の照合更新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照合更新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 **Reconciliation type** 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。

- **Update or insert**  
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。
- **Update only**  
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。
- **Insert only**  
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。
- **Delete**  
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。

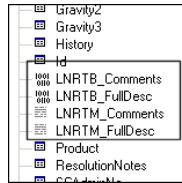
Lotus Notesコネクタが発行するドキュメントタイプでは、各バイナリフィールドは2つのフィールドに分かれます。

- 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。

## 添付ファイル

**\$FILE**コレクションを使用して添付ファイルを処理できます。このコレクションには以下の2つの要素が含まれます。

- **Content** : 添付ファイル、バイナリフォーマットでの内容
- **Name** : 添付ファイルの名前

## 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を以下に挙げます。

- `ldap\ldapnote\names.scn` [ 献 399]

---

## 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。

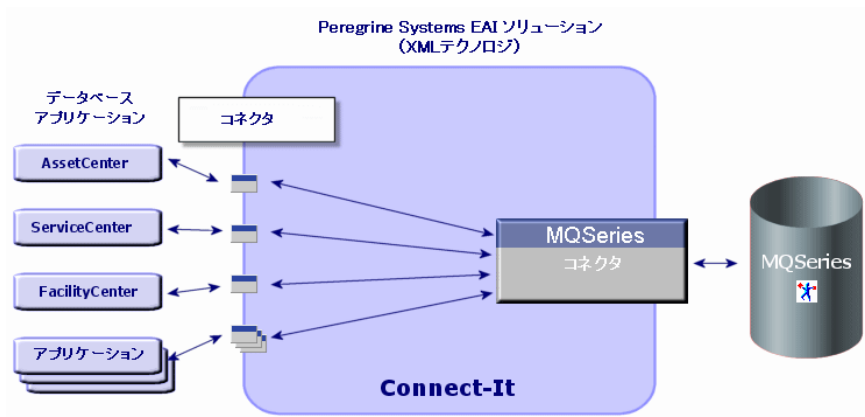
MQSeriesの新機能は、システムやネットワークインフラストラクチャに欠陥がある場合でも、データの送信を保証します。

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、*queues*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*queue manager*（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キュー内に配置できません。適切なキューに接続すると、MQSeriesコネクタはこのメッセージを読み取り、シナリオで指定されたHP Softwareアプリケーション内に送信します。デー

データ交換は、MQSeriesとHP Softwareの各アプリケーション間で双方向に行えます。



## 前提条件

Connect-Itをインストールするコンピュータには、MQSeriesアプリケーションクライアントをインストールする必要があります。

## 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 4.3. MQSeriesコネクタ（読み取り）の設定

|   |                                                       |
|---|-------------------------------------------------------|
| ■ | Name and describe the connector [ 献 16]               |
| ■ | Choosing a processing mode [ 献 165]                   |
| ■ | Configure the queue manager connection [ 献 165]       |
| ■ | Choosing a queue [ 献 166]                             |
| ■ | Define post-processing actions [ 献 167]               |
| ■ | Choose a DTD/XSD [ 献 167]                             |
| ■ | Configure the reconnection parameters (詳細モード) [ 献 29] |
| ■ | Determine server delay (詳細モード) [ 献 31]                |
| ■ | Configure schedule pointers (詳細モード) [ 献 33]           |

## Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Choosing a processing mode

**Basic configuration** 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。**Read**モードを選択します。

## Configure the queue manager connection

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。

- **Use an extended connection**

このオプションはデフォルトで選択されています。標準接続を使用するにはこのオプションをクリアします。

- **Server name**

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。

- **Connection port**

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。

- **Connection channel name**

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。

- **Queue manager**

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。

- **Alternate user ID**

必要に応じて代替ユーザIDを指定し、キューマネージャを開きます。

各キュー（メイン、成功、失敗）に対し、異なる代替ユーザIDを指定できます。



### 注意:

**Use extended connection**デフォルトオプションを選択しないと、**Queue manager name**フィールドのみが入力可能になります。

## Test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 **Test**をクリックします。

**Test the connection**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 **Close**をクリックして設定ウィザードへ戻ります。

### 注意:

コネクタを開くと、サーバのオペレーティングシステムが自動的に検出されます。

## Choosing a queue

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。

### Advanced configuration (詳細モード) オプション

このオプションで、キューの以下の項目を定義できます。

- ダイナミックキューを作成するための名前テンプレート  
使用するダイナミックキュー名の作成用フォーマットテンプレートを入力します。文字"\*"は作成されるダイナミックキュー名が一意になるように、キューマネージャにより置換されます。  
ダイナミックキューを閉じるための以下に挙げる3つのオプションから選択できます。
  - **Delete permanent dynamic queue if it is empty (MQCO\_DELETE)**  
一時ダイナミックキューも物理的に削除されます。
  - **Retain dynamic queue if it is permanent and delete it if it is temporary**
  - **Purge messages from permanent dynamic queue and delete it (MQCO\_DELETE\_PURGE)**  
一時ダイナミックキューも物理的に削除されます。
- 認証  
各キューに代替ユーザIDを指定できます。

## Define post-processing actions

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に応じて、以下の3つのオプションを選択できます。

- Leave the message in the queue.
- Delete it.
- Move it to a queue, which you will specify using the available text zone each time that this option is selected.

このおオプションを選択する際、メッセージのコンテキスト全体の保持（デフォルト動作）、識別コンテキストのみの保持、全コンテキストの破棄のいずれかを指定できます。

コンテキストは、以下のように構成されます。

- **ID context** : メッセージをキューに挿入するユーザに関する情報
- **Original context** : 挿入されたメッセージに関する情報（日付、メッセージを挿入したアプリケーション）

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。

## 処理後のアクションの使用

処理後のアクションを使用するには：


- ソースコネクタが生成したドキュメントタイプの*MessageInfo*構造体の**PutDate**および**MsgID**要素と、同じソースコネクタが取り込んだ**SuccessReport**ドキュメントタイプの*MessageInfo*構造体の同じ要素間に、マッピングを作成します。

## Choose a DTD/XSD

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、**File location**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。

### File location

以下の3つのオプションがあります。

- **Local/Network**

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、DTDまたはXSDの完全パス名を入力します。

- **FTP**

**Server**、**Login**、**Password**と**Path**フィールドにFTPパラメータを入力します。

**Path**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、**Server**フィールドに**ftp.mycompany.com**を入力し、**Path**フィールドに**/myfolder/DTV/mydtd.dtd**を入力します。

**注意**：DTDを学習モードでHTTP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**

**Server**、**Login**、**Password**と**Path**フィールドにHTTPパラメータを入力します。

**Path**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、**Server**フィールドに**http://mycompany.com**を入力し、**Path**フィールドに**/myfolder/DTV/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。

## Publish a document type for each first-level collection

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*Company*で、第1レベルに3つのコレクション*Employees*、*Suppliers*と*Locations*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*Company*になります。

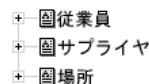


注意:

DTDの第1レベルのコレクションは、コネクタが発行する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。



**Publish a document type for each first-level collection** 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*Company*）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。



## ルート要素の決定

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*要素は複数の*Chapter*要素に含まれ、更に*Chapter*要素は1つの要素*Book*に含まれます）。他の要素に

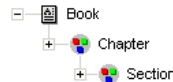
含まれることがない要素は、ルート要素と呼ばれます。DTDまたはXSDは、1つまたは複数のルート要素から構成されることもあれば、全くルート要素が存在しないこともあります。

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、Connect-Itでは以下の2つのオプションが選択できます。

## Publish a document type for each root element found in the DTD/XSD (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コネクタが発行するドキュメントタイプは1つのみです。



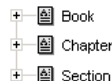
### 注意:

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。

## Publish a document type for each root element selected by the user (values are separated by commas)

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MQSeriesコネクタは、ユーザが選択したルート要素ごとに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。

## Learning mode

learning modeを使用すると、MQSeriesコネクタは、Choose a queue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:

learning mode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：

- **Learning mode** オプションを選択し、MQSeriesコネクタを設定します。
- Scenario diagram部分でMQSeriesコネクタを選択します。
- **Tools/ Produce** 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**Produce**を選択します。  
または
- **F5**キーを押します。

learning modeを使用しない場合は、MQSeriesコネクタを再度設定し、**Choose a DTD** 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。要素の少なく、限定された数のXMLを学習モードで処理した時に得られたDTDおよびXSDを使用すると、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。

- **Reject document and continue**  
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**Action after processing**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。

- Issue warning and continue

矛盾が発生しても、メッセージはMQSeriesによって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。

### Ignore the XML files whose root element does not correspond to the current document type

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。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

- **Convert dates and times to the local time zone** : このオプションは、UTCフォーマットで保存されている非日時を、ローカルタイムゾーンで 사용되는フォーマットに変換します。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Configure the JVM

このページが表示されるのは、設定ウィザードのChoose a DTD/XSD [ 献 87] ページでXSDを選択している場合のみです。

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 4.4. MQSeriesコネクタ（書き込み）の設定

|                                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------|
|  Name and describe the connector [ 献 16]               |
|  Choosing a processing mode [ 献 173]                   |
|  Configure the queue manager connection [ 献 173]       |
|  Choosing a queue [ 献 174]                             |
|  Choose a DTD/XSD [ 献 175]                             |
|  Configure the reconnection parameters (詳細モード) [ 献 29] |
|  Parallelize consumption (詳細モード) [ 献 41]               |

#### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

#### Choosing a processing mode

**Basic configuration** 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。**Write**モードを選択します。

#### Configure the queue manager connection

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。

- **Server name**  
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。
- **Connection port**  
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。
- **Channel name**  
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。
- **Name of the queue manager**  
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。
- **Alternate user ID**

必要に応じて代替ユーザIDを指定し、キューマネージャを開きます。  
各キュー（メイン、成功、失敗）に対し、異なるユーザIDを指定できます。



注意:

**Use extended connection**デフォルトオプションを選択しないと、**Queue manager name**フィールドのみが入力可能になります。

### Test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 **Test**をクリックします。

**Test the connection**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 **Close**をクリックして設定ウィザードへ戻ります。



注意:

コネクタを開くと、サーバのオペレーティングシステムが自動的に検出されます。

## Choosing a queue

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。

### Advanced configuration（詳細モード）オプション

このオプションで、キューの以下の項目を定義できます。

- ダイナミックキューを作成するための名前テンプレート  
使用するダイナミックキュー名の作成用フォーマットテンプレートを入力します。文字"\*"は作成されるダイナミックキュー名が一意になるように、キューマネージャにより置換されます。
- ダイナミックキューを閉じるための以下に挙げる3つのオプションから選択できます。
  - **Delete permanent dynamic queue if it is empty (MQCO\_DELETE)**  
一時ダイナミックキューも物理的に削除されます。
  - **Retain dynamic queue if it is permanent and delete it if it is temporary**

- **Purge messages from permanent dynamic queue and delete it (MQCO\_DELETE\_PURGE)**

一時ダイナミックキューも物理的に削除されます。

- **認証**

各キューに代替ユーザIDを指定できます。

### Specify the identity context option


このオプションを選択すると、キューに書き込むメッセージの識別コンテキストを指定できるようになります。

このオプションを選択すると、**MessageInfo**生成ドキュメントタイプの構造体を変更して、**UserID**、**AccountingToken**、**APppIdentityData**の各要素を表示できるようになります。

このオプションはデフォルトでは選択されていません。

### Choose a DTD/XSD

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。

をクリックすると、**File location**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。

### File location

以下の3つのオプションがあります。

- **Local/Network**

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**

**Server**、**Login**、**Password**と**Path**フィールドにFTPパラメータを入力します。

**Path**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、**Server**フィールドにftp.mycompany.comを入力し、**Path**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

**Server**、**Login**、**Password**と**Path**フィールドにHTTPパラメータを入力します。

**Path**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、**Server**フィールドに「`http://mycompany.com`」を入力し、**Path**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。

### Do not reference the resulting DTD in the XML file

このオプションはデフォルトで選択されています。

### Write resulting DTD inside the XML file

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。

### Write resulting DTD to an external file and reference it in the XML file

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の保存先とする場所を、フィールドに入力する必要があります。



### 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）。

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、使用されることになります。

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。

### ルート要素の決定

DTDは複数の要素から構成され、これらの要素に更に別の要素を含めることができます（例：文書発行用のDTDでは、複数のSection要素は複数のChapter要素に含まれ、更にこの要素は1つのBook要素に含まれます）。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には、複数のルート要素を含めることができます。



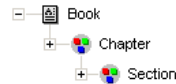
ルート要素（つまり、MQSeriesコネクタに発行される使用可能なドキュメントタイプに当たります）を指定するために、Connect-Itでは以下の2つのオプションが提示されます。

- Publish a document type for each root element found in the DTD/XSD（推奨オプション）
- Publish a document type for each root element selected by the user (values are separated by commas)

## Publish a document type for each root element found in the DTD/XSD（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。



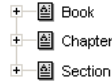
### 注意:

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。

## Publish a document type for each root element selected by the user (values are separated by commas)

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。



#### 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。

#### Configure the reconnection parameters（詳細モード）

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters（詳細モード） [ 献 29]」の節を参照してください。

#### Configure the JVM

このページが表示されるのは、設定ウィザードのChoose a DTD/XSD [ 献 87]ページでXSDを選択している場合のみです。

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

#### Parallelize consumption（詳細モード）

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Parallelize consumption（詳細モード） [ 献 41]」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**Produced document types**タブ内で、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キューの中の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われます：**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。

## メッセージの変換

すべての生成ドキュメントタイプで、**Enable conversion of messages to** オプションを利用できます。このオプションで、以下のコードページでキューメッセージの変換を利用できるようになります。

- ISO-8859-1
- ISO-8859-2
- UTF-8

このオプションはデフォルトでは有効ではありません。

## Get messages only if all messages in the group are available 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**Consumed document types**タブ内で、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ボックスを、編集または作成します。

- 3 **Message options**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
- 相関ID

IDは16進数形式で入力します。

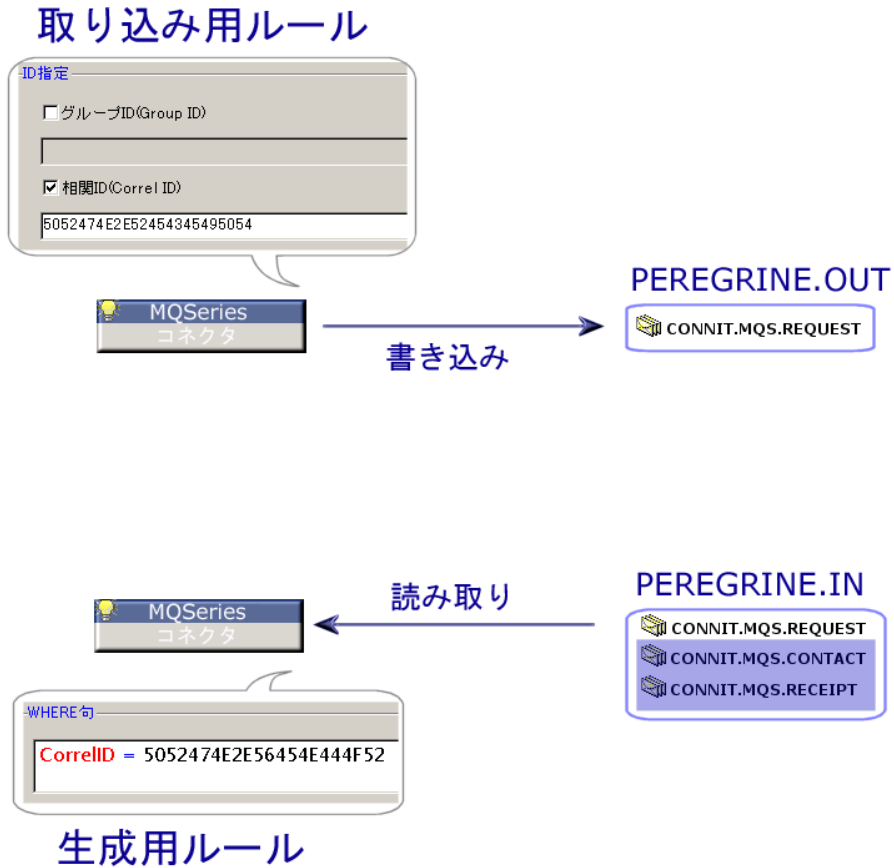
MQSeriesコネクタ付属のMQSeries - Asset Managementシナリオでは、取り込み用ドキュメントタイプ用の相関ID (CorrelID) は以下の通りです。

- **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、相関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 - 相関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

**MessageInfo**構造体を書き込みモードのコネクタが発行する取り込みドキュメントタイプにあります。この構造体を使用して、送信メッセージの一部パラメータを指定できます。

この構造体に含まれるフィールドを以下に挙げます。

- **Expiry**

このフィールドで、メッセージの遅延期限を定義します。この期限を経過するとキューマネージャがメッセージを削除します。このフィールドの値は、10分の1秒単位で入力されます。メッセージに期限がない場合、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

キューマネージャが再起動されると、メッセージは有効ではなくなります。

- 1

キューマネージャが再起動されても、メッセージは有効です。

- 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。

- **AccountingToken**

このフィールドは、メッセージの識別コンテキストの要素です。このフィールドが表示されるのは、コネクタ設定ウィザードで**Specify the identity context**オプションが選択されている場合のみです。

- **ApplIdentityData**

このフィールドは、メッセージの識別コンテキストの要素です。このフィールドが表示されるのは、コネクタ設定ウィザードで**Specify the identity context**オプションが選択されている場合のみです。

- **UserID**

このフィールドは、メッセージの識別コンテキストの要素です。このフィールドが表示されるのは、コネクタ設定ウィザードで**Specify the identity context**オプションが選択されている場合のみです。

### 読み取りモードの生成ドキュメントタイプにあるMessageInfo構造体

コネクタが発行する生成ドキュメントタイプには、**MessageInfo**構造体があります。この構造体には、読み取られたメッセージのパラメータの一部が含まれます。

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。

- **AccountingToken**
- **ApplIdentityData**
- **BackoutCount**
- **Expiry**
- **MsgType**
- **Persistence**
- **Priority**
- **PutDate**
- **ReplyToQ**
- **ReplyToQMgr**
- **Report**
- **UserID**
- **MsgID**
- **CorrelationID**
- **GroupID**



## MessageInfoドキュメントタイプの使用

書き込みモード（メッセージの送信）では、送信対象メッセージに対応するドキュメントが取り込まれるたびに、コネクタはドキュメントタイプ**MessageInfo**を公開します。このドキュメントタイプには、送信メッセージのパラメータに対応する以下のフィールドがあります。

- **AccountingToken**
- **ApplIdentityData**
- **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を以下に挙げます。

- `mqseries\ac44mq\mqtoprgn.scn` [ 献 397]
- `mqseries\ac44mq\prgntomq.scn` [ 献 397]
- `mqseries\sc51mq\mqsc.scn` [ 献 397]
- `mqseries\sc60mq\mqsc.scn` [ 献 398]
- `mqseries\sc61mq\mqsc.scn` [ 献 398]

---

## 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:

新しいの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ではサポートされていないためです。

## 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 4.5. NTセキュリティコネクタの設定

 Name and describe the connector [ 献 16]

 Enter an NT domain [ 献 186]

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Enter an NT domain

**Domain** 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:

Scenario diagram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。

### Recovering the list of computers in a domain

**Use the domain controllers to recover a domain's list of computers**を使用すると、特定のNTドメインにあるコンピュータの完全なリストを取得できます。

## NTセキュリティコネクタが発行するドキュメントタイプ

NTセキュリティコネクタが発行する3つのドキュメントタイプを以下に挙げます。

- Machine
- NtDomain
- User



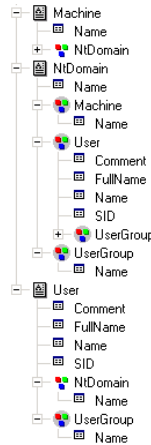
#### 注意:

「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す。

NTセキュリティコネクタが発行する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 - 発行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（ディレクトティブ） [ 献 43]」章の「生成用ルール [ 献 44]」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、名前でフィルタをかける

**Filter items by name** 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 4.6. 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 | Platform 1 | Platform 1 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 |

| コレクション | フィルタ | 取得する値                                     |
|--------|------|-------------------------------------------|
| User   | Doe? | ドメインのユーザの内、名前の始めがDoeで、名前の終わりの文字が不明である全ユーザ |

## NTセキュリティコネクタのスケジュール

AssetCenter Serverでは、NTドメインで宣言されているユーザの取得をスケジュールできます。

シナリオのスケジューリングについては、AssetCenter『管理』の「AssetCenter Server」の章、「AssetCenter Serverでモニタするモジュールを設定する」を参照してください。

## 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を以下に挙げます。

- NTセキュリティシナリオ [ 献 403]

## Mercuryコネクタ

これらのコネクタにより、Mercury MAMやBACの各スイートに関連するデータが読み込み可能になります。

これらのコネクタは、Connect-It付属の『ServiceCenter-MAM/BAC Integration Solution』に記載されています。

## 付属シナリオ - Mercuryコネクタ

これらのコネクタを使用する付属シナリオを以下に挙げます。

- Mercuryシナリオ [ 献 395]

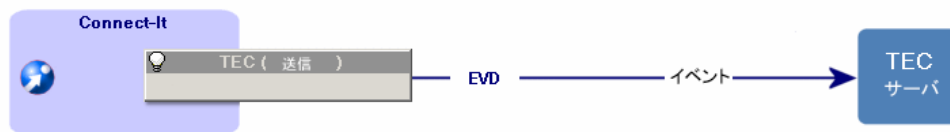
## Tivoli Enterprise Consoleコネクタ（送信）

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) ヘイブントを送信できます。

このコネクタは、TEC EVD APIを使用してイベントを送信するTECアダプタです。



## 前提条件

Tivoli Enterprise Consoleコネクタでは、Tivoli Enterprise Consoleのインストールに含まれるEFI (Event Integration Facility) APIにあるJavaライブラリを使用します。

該当するライブラリを以下に挙げます。

- `evd.jar`
- `log.jar`

これらは、[TECのインストールフォルダ]/bin/generic\_unix/TME/TECにあります。

コネクタを使用するには：

- ライブラリをConnect-Itのインストールフォルダにある'lib'サブフォルダにコピーします。
- Connect-Itのグローバルな'classpath'を変更し、これらのライブラリを追加します (**Java/Configure JVM**メニュー)。
- これらのライブラリを個別に'classpath'に追加します (コネクタ設定ウィザードの**Configure JVM**ページ)。

## Tivoli Enterprise Consoleアダプタ (送信) の設定

ウィザードでコネクタを設定できます。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 4.7. Tivoli Enterprise Consoleアダプタ (送信) の設定

[Name and describe the connector \[ 献 16\]](#)

[Configure TEC server configuration \[ 献 191\]](#)

📖 イベントクラスの指定 [ 献 191]

📖 Advanced configuration (詳細モード) [ 献 193]

📖 Configure the JVM [ 献 193]

---

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Configure TEC server configuration

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。

#### Server

TECサーバの名前またはアドレスを入力します。

これは必須フィールドです。

#### Port

サーバポートを指定します。デフォルトのポートは5529です。

これは必須フィールドです。

#### The server does not support UTF-8.

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をサポートしない場合は、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、**Server code page**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。

#### Server code page

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。

## イベントクラスの指定

### BAROCファイル

TECイベントクラスを含むBAROCファイルを指定する必要があります。これら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する必要があります。

Tivoli Enterprise Console BAROCファイルには、イベントクラスの階層が含まれます。このため、BAROCファイルの宣言の順番は重要であり、この階層構造を反映します。two.barocファイルが、one.barocファイルで定義されているイベントBを継承するイベントAを定義する場合、one.barocファイルはtwo.barocファイルより前にリストされる必要があります。



#### 注意:

バージョン3.4より前のTECコネクタをベースにしているシナリオは、再設定する必要があります。

これは必須フィールドです。


イベントクラスに含まれるフィールドは、拡張ファイルで拡張できます。

### 拡張ファイル



イベントクラスを定義するBAROCファイルは、INTEGER、INT32、STRING、およびREALのデータタイプを処理します。

**Extension file**フィールドを使用して、BAROC拡張ファイルを指定できます。これは、BAROCファイルに含まれるイベントクラスの定義を拡張するXMLファイルです。このファイルで行う作業を以下に挙げます。

- 各イベントクラスの説明、および各イベントクラスフィールドの説明を指定します。この説明は、Connect-Itドキュメントタイプエディタに表示されません。
- イベントクラスフィールドタイプを拡張します。これにより、フィールドを'Date and time'タイプとして宣言することで、Connect-Itでの処理がより容易になります。

ウィザードで、この拡張ファイルの作成と変更を行います。拡張ファイルにアクセスするには、拡大鏡をクリックします。

拡張ファイルの作成/変更を行うには：


- 1 拡張ファイルのパスを指定します
- 2 をクリックします。
- 3 イベントクラスフィールド、特に「日時」タイプフィールドを変換するのに使用する地域設定を定義します (**Locale**)。デフォルトでは、Connect-Itが使用する地域設定です。
- 4 拡張対象とするイベントクラスのリストに入力します ()
- 5 **Next**をクリックして、イベントクラスフィールドの定義を拡張します。



#### 注意:

イベントクラスとそのフィールドはBAROCフィールドで定義済みである必要があります。定義されていない場合、コネクタを開いたときにエラーがドキュメントログに記録されます。

しかし、すべてのイベントクラスや、イベントクラスのすべてのフィールドを拡張する必要はありません。拡張しないイベントクラスとフィールドは、BAROCファイルの定義通りになります。

- 6 をクリックして、フィールドの拡張とそのタイプへの入力を行います。「日付」、「時刻」「日時」タイプのフィールドでは、使用するフォーマットを



指定できます。このフィールドタイプで使用されるデフォルトのフォーマットは、1970年1月1日からの経過秒数です。

- 7 すべてのフィールドの定義が終わったら、**Finish**をクリックします。

## Advanced configuration (詳細モード)

### Activate the event buffer

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。

### Buffer file

バッファファイルの完全パスを指定します。

**Activate event buffer**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。

### Maximum size (KB) of the buffer file

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。

## Configure the JVM

このページが表示されるのは、設定ウィザードのChoose a DTD/XSD [ 献 87] ページでXSDを選択している場合のみです。

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

## 付属シナリオ

付属シナリオは提供されていません。

## Tivoli Enterprise Consoleコネクタ（受信）

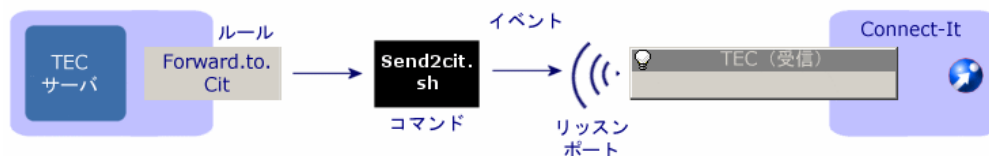
オプションコネクタ

■ 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ッスンポートにより、TECが送信するイベントをリッ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「TECでのイベント送信ルールの作成 [ 献 196]」を参照してください。



### 前提条件

Tivoli Enterprise Consoleコネクタでは、Tivoli Enterprise Consoleのインストールに含まれるEFI（Event Integration Facility）APIにあるJavaライブラリを使用します。

該当するライブラリを以下に挙げます。

- evd.jar
- log.jar

これらは、[TECのインストールフォルダ]/bin/generic\_unix/TME/TECにあります。

コネクタを使用するには：

- ライブラリをConnect-Itのインストールフォルダにある'lib'サブフォルダにコピーします。
- Connect-Itのグローバルな'classpath'を変更し、これらのライブラリを追加します（**Java/Configure JVM**メニュー）。
- これらのライブラリを個別に'classpath'に追加します（コネクタ設定ウィザードの**Configure JVM**ページ）。

## 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 4.8. TECコネクタの設定

 [Name and describe the connector \[ 献 16\]](#)

 [Configure events listening \[ 献 195\]](#)

 [イベントクラスの指定 \[ 献 195\]](#)

 [Configure the JVM \[ 献 196\]](#)

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Configure events listening

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に対応します。詳細については、本節の「[send2cit.shファイルの設定 \[ 献 198\]](#)」を参照してください。

これは必須フィールドです。

### イベントクラスの指定

#### BAROCファイル

TECイベントクラスを含むBAROCファイルを指定する必要があります。これら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する必要があります。

これは必須フィールドです。

Tivoli Enterprise Console BAROCファイルには、イベントクラスの階層が含まれます。このため、BAROCファイルの宣言の順番は重要であり、この階層構造を反映します。two.barocファイルが、one.barocファイルで定義されているイベントBを継承するイベントAを定義する場合、one.barocファイルはtwo.barocファイルより前にリストされる必要があります。

#### 注意:

バージョン3.4より前のTECコネクタをベースにしているシナリオは、再設定する必要があります。

イベントクラスに含まれるフィールドは、拡張ファイルで拡張できます。イベントクラスに含まれるフィールドのカスタマイズについては、本章の節「拡張ファイル [ 献 192]」を参照してください。

### 拡張ファイル

「拡張ファイル [ 献 192]」を参照してください。

### Configure the JVM

このページが表示されるのは、設定ウィザードのChoose a DTD/XSD [ 献 87]ページでXSDを選択している場合のみです。

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

## 追加情報

### TECでのイベント送信ルールの作成

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。例えば、TEC管理者がエラー警告を受信し、個別の問題の解決手順を実装するようにできます。

#### 注意:

send2citコマンドは、UNIXのシェルスクリプト (.sh) ですが、このスクリプトはWindowsバージョンのTECでも使用できます。UNIXシェルがTECに用意されています。

イベントをコネクタに送信するには、send2citを設定してコネクタの位置、Connect-Itをホストするサーバの名前やアドレス、およびコネクタのリスニング

ポートを与える必要があります。節「[send2cit.shファイルの設定 \[ 献 198\]](#)」を参照してください。

### send2citファイルの場所

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。

- getvalue.exe
- tecevent.jar
- tecevent\_lang.jar
- send2cit.sh
- tecevent.config
- 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 - TEST\_CLASSイベントクラスからConnect-Itへのイベントの送信

以下の例は、TECが受信した際、TECコネクタ（受信）にイベントを送信するルールをTECで作成する簡単な例です。

イベントクラスTEST\_CLASSのイベントを、送信するためのルールを作成するには：

- 1 Tivoli Desktopを起動します。
- 2 新規ルールベースを作成します。
- 3 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。
- 4 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。
- 5 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。
- 6 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 **Launch a command**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。

- -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。

7 ルールベース (**Forward**) を保存します。

8 ルールベースをコンパイルします。

9 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**に受信されます。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**のマニュアルを参照してください。

## send2cit.shファイルの設定

**send2cit.sh**の設定方法を以下に挙げます。

- コマンドラインから設定
- 設定ファイルを使用して設定

設定ファイルを**send2cit.sh**ファイルの設定に使用した場合、コマンドラインオプションは無視されます。

## コマンドラインでの設定

**Connect-It**バージョン3.4.0の前では、**send2cit**コマンドはコマンドラインから設定できました。この機能は今でもサポートされていますが、設定ファイルを使用することを推奨します。

コマンドパラメータには大文字小文字の区別があります。

### 注意:

**-h**コマンドラインパラメータを使用すると、**send2cit**コマンドにヘルプが表示されます。

## コマンドラインで使用する**TEC**コネクタ (受信) の場所

**TEC**コネクタ (受信) の場所は、2つのパラメータによって与えられます。

- **Connect-It**をホストするサーバの名前またはアドレス。これは、**-s**パラメータを使用して指定します。デフォルト値は'**localhost**'です。

- コネクタのリスニングポートです。これは、`-p`パラメータを使用して指定します。

例：

```
send2cit -s connectitsvr -p 12345
```

## コマンドラインからのログファイルの使用

`send2cit`コマンドで、ログファイルに書き込みできます。

`-l`パラメータを使用して、ログファイルを指定できます。`-m`パラメータを使用してログファイルの最大サイズをキロバイトで指定することもできます。

例：

```
send2cit -s connectitsvr -p 12345 -l /tec/connectit/send2cit.log -m 2048
```

## 設定ファイルを使用した設定

`send2cit`コマンドは、設定ファイルから設定を読み込むことができます。設定ファイルにより、`send2cit`の詳細な機能を使用することができます。

`send2cit`設定ファイルは、プロパティファイルと同じシンタックスを使用します。

```
Comment
Property = Value
OtherProperty = Other value
```

サンプルの設定ファイルが、Connect-Itに用意されています。

### 注意:

`-listCfgProps`コマンドラインパラメータを使用すると、設定ファイルのプロパティとその説明のリストが表示されます。

## 使用する設定ファイルの定義

デフォルトで`send2cit`は、`send2cit.sh`の次に見つければ、`tecevent.config`ファイルを使用します。このファイルが存在しない場合や、別のファイルを使用する場合、`-cfg`コマンドラインパラメータを使用できます。

例：

```
send2cit -cfg /tec/connectit/myConfig.conf
```

## TECコネクタ（受信）の場所

TECコネクタ（受信）の場所は、設定ファイル中にある以下の2つのパラメータによって与えられます。

- **Server** : Connect-Itをホストするサーバの名前またはアドレス。デフォルト値は'localhost'です。
- **Port** : TECコネクタ (受信) のリスニングポートです。

 **注意:**

`-s`および`-p`コマンドラインパラメータを使用すると、設定ファイル中の値は無視されます。

## ログファイルの使用

設定ファイルにある2つのプロパティで、ログファイルを指定できます。

- **LogFile** : 使用するログファイルの完全パスです。
- **MaxLogSize** : ログファイルの最大サイズ (キロバイト) です。デフォルトのサイズは'2000'です。

 **注意:**

`-l`および`-m`コマンドラインパラメータを使用すると、設定ファイル中の値は無視されます。

## 一般イベントクラスの定義

Tivoli Enterprise Console設定では、イベントクラスから成る集合の親クラスとしてイベントクラスが多用されます。

イベントクラスであるNetworkProblemは、イベントクラスNP1、NP2、NP3、NP4...の親クラスとして使用されます。NPxイベントクラスの名前はネットワーク問題を識別しますが、NPxイベントクラスがNetworkProblem親クラスのフィールドを満たしてしまふことはありません。

これらのイベントをConnect-Itで処理するのに、一例として、NetworkProblemから導出されるすべてのイベントクラスに対して同一マッピングを使用し、ServiceCenterでチケットを開くという方法が考えられます。

これを行うには、NPxクラスをグループ化するsend2cit設定ファイルで、一般イベントクラスであるNetworkProblemを定義します。send2citがNPxクラスからConnect-Itにイベントを送信するよう呼び出されると、send2citはNetworkProblemイベントを送信します。

一般イベントクラスを定義すると、実際のイベントクラスは、Connect-It内のTECコネクタ (受信) が生成するドキュメントタイプの'EVENT\_CLASS'要素内に存在します。

一般イベントクラスは、プレフィックス'G\_'を使用して設定ファイル内で定義されます。

```
G_MyGenericClass = SpecificClass1 SpecificClass2 SpecificClass3
```



この例では、一般イベントクラスであるNetworkProblemを定義するため、設定ファイルには以下の行が含まれている必要があります。

```
G_NetworkProblem = NP1 NP2 NP3 NP4
```

## 接続問題の処理

TECコネクタ（受信）の接続タイムアウトを指定することができます。この時間を超過すると接続は失敗したとみなされます。この期間を指定するには、*TimeOut* プロパティを使用します。値はミリ秒で表します。

デフォルトでは、TECコネクタ（受信）への接続が失敗した場合にイベントログがアクティブになっていないと、エラーが発生し、エラーコードがsend2citによって返されます。

接続が失敗した場合にイベントをTECに送信するよう、send2citを設定することもできます。このイベントにより、TEC管理者は問題を修正できます。送信されるイベントのクラスは、*CIT\_TecEvent\_ConnectionFailure*です。この機能を有効にするには、以下のプロパティを使用します。

- **SendEventOnCnxFailure**：イベントを送信するには、このプロパティの値を'true'に設定する必要があります。
- **CnxEventFailureSeverity**：送信されるイベントの重大度です。デフォルト値は'WARNING'です。

TECにイベントを送信するには、send2citとTECの追加設定が必要になります。詳細については、節「TECへのイベントの送信 [ 献 202 ]」を参照してください。

## イベントキャッシュの使用

TECサーバとTECコネクタ（受信）間で接続エラーが発生した場合、送信できなかったイベントがキャッシュフォルダに格納されます。接続が回復すると、キャッシュの内容がサーバに送信されます。この方法でイベントが保たれます。

以下の2つのプロパティを使用して、キャッシュの使用を指定します。

- **UseCache**：このプロパティの値が'true'に設定されていると、キャッシュが有効になります。
- **CacheDir**：このプロパティは、イベントキャッシュの格納に使用されるフォルダを指定します。

キャッシュを有効にする場合、使用するフォルダを指定する必要があります。

### キャッシュの概要

send2citがTECコネクタ（受信）へのイベントの送信に失敗すると、このイベントはキャッシュにキャッシュフォルダ内のファイルとして格納されます。フォルダの名前は、<サーバ>\_<ポート>\_XXXX.cacheという形式です。ここで、XXXXはファイル名の一意性を保証するシーケンス番号です。

send2citの次の呼び出しが行われると、接続が再確立され、send2citは新規イベントの送信前にキャッシュの削除、つまりキャッシュに格納されているイベントの送り返しを開始します。

### キャッシュのサイズの制限

キャッシュのサイズを制限すると、キャッシュフォルダに保存されるイベント数が制限されます。

デフォルトでは、キャッシュのサイズは100イベントまでに制限されています。この値を変更するには、**MaxCacheSize**プロパティを使用します。

**MaxCacheSizeAction**プロパティにより、イベントが送信不可能である場合、およびキャッシュが最大サイズに達している場合のsend2citの動作を決定します。取り得る値を以下に挙げます。

- **Drop** : イベントは却下され、send2citによってエラーメッセージが生成され、エラーコードが返されます。
- **Error** : イベントはキャッシュに配置されますが、send2citによってエラーメッセージが生成され、エラーコードが返されます。
- **Warning** : イベントはキャッシュに配置され、警告メッセージが生成されません。
- **None** : イベントがキャッシュに配置され、その他のアクションは発生しません。この値は、キャッシュの無効化と同じ意味を持ちます。
- **SendEvent** : イベントがキャッシュに配置されますが、イベントはTECサーバに送信されます。

デフォルトの動作は'None'、つまり、デフォルトではキャッシュのサイズは制限されていません。**MaxCacheSizeAction**プロパティが**SendEvent**に設定されていると、**CIT\_TecEvent\_CacheFull**クラスのイベントがConnect-Itに送信されます。

**CacheFullEventSeverity**プロパティで、送信されるイベントの重大度レベルを指定できます。デフォルトの重大度は'MINOR'です。

TECにイベントを送信するには、send2citとTECの両方に追加設定が必要になります。詳細については、「[TECへのイベントの送信 \[ 献 202\]](#)」の節を参照してください。

## TECへのイベントの送信

接続の問題が発生した場合やキャッシュが最大サイズに達した場合に、send2citがイベントをTECに送信できるようにするには、send2citとTECを設定する必要があります。

### TECへのイベント送信用のsend2citの設定

イベントをTECに送信する際、以下のプロパティが使用されます。

- **TECServer** : イベントの送信先となるTECサーバの名前またはアドレスです。デフォルト値は'localhost'です。
- **TECPort** : イベントの送信先となるTECサーバの接続ポートです。デフォルト値は'5529'です。
- **SendEventCommand** : イベントの送信に使用するコマンドです。デフォルト値は'postemsg'です。この場合、postemsgコマンドはPATHに存在する必要があります。コマンドの完全パスを指定することができます。
- **EventSource** : 送信イベントのソース

## TECからのイベント受信用のsend2citの設定

send2citが送信するイベントクラスは、send2citに用意されているBAROCファイルcit\_event.barocで定義します。send2citが送信したイベントをTECが受信するには、このファイルをTECにインポートする必要があります。

send2citイベントクラスは、同一親クラスであり、以下の4つのフィールドを定義するCIT\_TecEvent\_Errorから導出されます。

- tecevent\_error\_code
- src\_event\_handle
- src\_date\_reception
- src\_server\_handle

**tecevent\_error\_code**フィールドには、イベント送信を引き起こすsend2citエラーコードを入力します。

その他の3つのフィールドで、イベントとsend2citエラーの原因となったイベントとを関連付けることができます。

send2citが送信するイベントのソースとsend2cit設定ファイルにある**EventSource**が定義するイベントのソースも、TECで宣言する必要があります。

## send2citエラーコード

- 1 送信するイベントデータを含む環境変数を取得する際のエラー。
- 2 コマンドラインパラメータが正しくない場合。
- 3 イベントをTECコネクタ（受信）に送信する際のエラー。
- 4 設定ファイル読み込み中のエラー。
- 5 コマンドラインパラメータ処理中のエラー。
- 6 環境変数処理中のエラー。変数がない場合。
- 7 エラーが不明の場合。

## 付属シナリオ

付属シナリオは提供されていません。



## 5 プロトコルコネクタ

プロトコルコネクタは、特定の通信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、DB2や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 Administratorが前もってインストールされている必要があります。

### データベースコネクタの互換性

Database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                       | Sybase      | DB2     |
|------------------------------------------------|------------------------------|-------------|---------|
| Accessバージョン<br>4.00.4403.02                    | Oracle 7バージョン<br>2.05.0301   | Sybase 11.9 | DB2 7.x |
| Excel 4.00.4403.02                             | Oracle 8.0バージョン<br>8.00.0500 | Sybase 12.x | DB2 8.x |
| Sybase 11.9バージョン<br>3.11.0001                  | Oracle 8iバージョン<br>8.01.0500  |             |         |
| Sybase SQL<br>Anywhereバージョン<br>5.05.052739     |                              |             |         |
| Textバージョン<br>4.00.4403.02                      |                              |             |         |
| Microsoft SQL Server<br>バージョン3.70.08.20        |                              |             |         |
| IBM DB2バージョン<br>7.x (バージョン6以降<br>のDB2)         |                              |             |         |
| INFORMIX 3.33 32<br>BIT バージョン<br>3.33.00.10688 |                              |             |         |

Databaseコネクタは、全てのODBCドライバのバージョン3.xxと互換性はありません。



**注意:**

DatabaseコネクタはAssetCenterデータベースには接続できません。

## 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、同一バージョンレベルのDLLファイルを使用するODBCドライバとODBC Administratorと併せて使用することを強く推奨します。

例:

ODBCAdministratorバージョン3.0と、Access4.00ODBCドライバを使用していると、Database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

Databaseコネクタが発行する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、Database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。

## DB2 MVS - Microsoft ODBC driver for DB2

以下の設定が使用されています。

- DB2 Connect DB2 SDK 8.1.10
- ODBC IBM DB2 driver 8.01.00.36
- Z/OS V1.2 DB2 V7.2 server

| フィールドタイプ     | 読み込みのみのサポート | 書き込みのサポート |
|--------------|-------------|-----------|
| Blob         | 不可          | 不可        |
| character(n) | 可           | 可         |
| date         | 可           | 可         |
| decimal(p,s) | 可           | 可         |
| float(p)     | 可           | 可         |
| integer      | 可           | 可         |
| long varchar | 可           | 可         |
| timestamp    | 可           | 可         |
| varchar      | 可           | 可         |

## DB2 - ネイティブ接続

| フィールドタイプ         | 読み込みのみのサポート   | 書き込みのサポート     |
|------------------|---------------|---------------|
| bigint (64ビット整数) | 可 (32ビット整数のみ) | 可 (32ビット整数のみ) |
| blob             | 不可            | 不可            |
| character        | 不可            | 不可            |
| clob             | 可             | 可             |
| date             | 可             | 可             |
| datalink         | 不可            | 不可            |
| decimal          | 可             | 可             |
| dbclob           | 不可            | 不可            |
| double           | 可             | 可             |
| char             | 可             | 可             |
| graphic          | 不可            | 不可            |
| integer          | 可             | 可             |
| real             | 可             | 可             |
| time             | 可             | 可             |
| smallint         | 可             | 可             |
| timestamp        | 可             | 可             |
| varchar          | 可             | 可             |
| varchargraphic   | 不可            | 不可            |

## Oracle 8.0 - Microsoft ODBC driver for Oracle

| フィールドタイプ | サポート可／不可 |
|----------|----------|
| bfile    | 不可       |
| Blob     | 可        |

| フィールドタイプ | サポート可／不可 |
|----------|----------|
| char     | 可        |
| clob     | 不可       |
| date     | 可        |
| float    | 可        |
| integer  | 可        |
| long     | 可        |
| long raw | 可        |
| nchar    | 可        |
| nclob    | 不可       |
| nvarchar | 不可       |
| number   | 可        |
| ms_label | 可        |

### Oracle 8.0 - ネイティブ接続

| フィールドタイプ | 書き込みのサポート | 読み込みのみのサポート |
|----------|-----------|-------------|
| bfile    | 不可        | 不可          |
| Blob     | 不可        | 不可          |
| char     | 可         | 可           |
| clob     | 不可        | 不可          |
| date     | 可         | 可           |
| float    | 可         | 可           |
| integer  | 可         | 可           |
| long     | 可         | 可           |
| long raw | 可         | 可           |
| nchar    | 不可        | 不可          |
| nclob    | 不可        | 不可          |
| nvarchar | 不可        | 不可          |
| number   | 可         | 可           |
| ms_label | 不可        | 不可          |

### 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    | 可        |
| image    | 可        |
| int      | 可        |
| money    | 可        |
| nchar    | 不可       |
| ntext    | 可        |



| フィールドタイプ         | サポート可／不可 |
|------------------|----------|
| numeric          | 可        |
| nvarchar         | 不可       |
| real             | 可        |
| smalldatetime    | 可        |
| smallint         | 可        |
| smallmoney       | 可        |
| text             | 可        |
| timestamp        | 可        |
| tinyint          | 可        |
| uniqueidentifier | 不可       |
| varbinary        | 可        |
| varchar          | 可        |

### MySQL - ネイティブ接続

| フィールドタイプ         | 読み込みのみのサポート | 書き込みのサポート |
|------------------|-------------|-----------|
| bigint           | 可           | 可         |
| bit              | 可           | 可         |
| blob             | 可           | 可         |
| bool             | 可           | 可         |
| char             | 可           | 可         |
| date             | 可           | 可         |
| decimal          | 可           | 可         |
| double           | 可           | 可         |
| double precision | 可           | 可         |
| enum             | 不可          | 不可        |
| float            | 可           | 可         |
| int              | 可           | 可         |
| longblob         | 可           | 可         |
| longtext         | 可           | 可         |
| mediumblob       | 可           | 可         |
| mediumtext       | 可           | 可         |
| numeric          | 可           | 可         |
| set              | 不可          | 不可        |
| smallint         | 可           | 可         |
| text             | 可           | 可         |
| time             | 可           | 可         |
| timestamp        | 可           | 可         |
| tinyblob         | 可           | 可         |
| tinyint          | 可           | 可         |
| tinytext         | 可           | 可         |

| フィールドタイプ | 読み込みのみのサポート | 書き込みのサポート |
|----------|-------------|-----------|
| varchar  | 可           | 可         |

## Sybase12 - Sybase System 11ドライババージョン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         | 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        |

## Sybase12 - ネイティブ接続

| フィールドタイプ      | 読み込みのみのサポート | 書き込みのサポート       |
|---------------|-------------|-----------------|
| binary        | 不可          | 不可              |
| bit           | 不可          | 不可              |
| char          | 可           | 可               |
| datetime      | 可           | 可               |
| decimal       | 可           | 可               |
| float         | 可           | 可               |
| image         | 可           | 可 (バージョン2.70以上) |
| int           | 可           | 可               |
| money         | 可           | 可               |
| nchar         | 不可          | 不可              |
| ntext         | 不可          | 不可              |
| numeric       | 可           | 可               |
| nvarchar      | 不可          | 不可              |
| real          | 可           | 可               |
| smalldatetime | 可           | 可               |
| smallint      | 可           | 可               |

| フィールドタイプ   | 読み込みのみのサポート | 書き込みのサポート |
|------------|-------------|-----------|
| smallmoney | 可           | 可         |
| text       | 可           | 可         |
| tinyint    | 可           | 可         |
| varbinary  | 不可          | 不可        |
| varchar    | 可           | 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

このコネクタに関する一部のオプションを有効にするには、**Edit/ Options/ Connector**メニューを使用します。詳細については、『ユーザガイド』、「シナリオビルダ」、「コネクタ」を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 5.1. データベースコネクタの設定

|                                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------|
|  Name and describe the connector [ 献 16]               |
|  Select a connection type [ 献 18]                      |
|  Configuring the connection [ 献 19]                    |
|  Advanced configuration (詳細モード) [ 献 27]                |
|  Configure the reconnection parameters (詳細モード) [ 献 29] |
|  Determine server delay (詳細モード) [ 献 31]                |
|  Configure schedule pointers (詳細モード) [ 献 33]           |
|  Manage transactions (詳細モード) [ 献 35]                  |
|  Configure the cache (詳細モード) [ 献 36]                 |
|  Advanced options (詳細モード) [ 献 37]                    |
|  Define document types (詳細モード) [ 献 39]               |
|  Parallelize consumption (詳細モード) [ 献 41]             |

### 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Select a connection type \[ 献 18\]](#)」の節を参照してください。

## Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Configuring the connection \[ 献 19\]](#)」の節を参照してください。

## Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Advanced configuration \(詳細モード\) \[ 献 27\]](#)」の節を参照してください。

- **SQL92 supported** : ANSISQL92標準のMySQL拡張を使用する場合、このオプションを選択してください。
- **Execute an initial import** : このオプションを選択すると、データベースにレコードがない場合や挿入する要素がまだ存在しない場合といった初期インポートの場合に、照合更新スクリプトでSELECTクエリを使用しないで済ませることができます。これによりレコードの処理時間を短縮します。

## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Configure the reconnection parameters \(詳細モード\) \[ 献 29\]](#)」の節を参照してください。

## Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Determine server delay \(詳細モード\) \[ 献 31\]](#)」の節を参照してください。

## Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Configure schedule pointers \(詳細モード\) \[ 献 33\]](#)」の節を参照してください。

## Manage transactions (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Manage transactions \(詳細モード\) \[ 献 35\]](#)」の節を参照してください。

## Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Configure the cache \(詳細モード\) \[ 献 36\]](#)」の節を参照してください。

## Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Advanced options \(詳細モード\) \[ 献 37\]](#)」の節を参照してください。

- ODBC 3のOracleとの互換性：**IsODBC3Compliant**詳細オプションを追加して、互換性を確保してください。

ODBCドライバがODBC 3互換ではない場合、**IsODBC3Compliant=0**のように値を定義します。

#### Define document types（詳細モード）

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types（詳細モード） [ 献 39]」の節を参照してください。

#### Parallelize consumption（詳細モード）

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Parallelize consumption（詳細モード） [ 献 41]」の節を参照してください。

## Database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） [ 献 43]」章の「生成用ルール [ 献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。これらの句の詳細については「コネクタのルール（ディレクトティブ） [ 献 43]」章の「生成用ルール [ 献 44]」節、「WHERE句とORDER BY句 [ 献 45]」を参照してください。

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。

#### 注意:

WHERE句とORDERBY句は、生成されたドキュメントタイプのルートレベルとコレクションのレベルに適用されます。

AQLシンタックスの詳細については、「AQLクエリ [ 献 435]」の章、「生成用ルール [ 献 44]」の「WHERE句とORDER BY句 [ 献 45]」を参照してください。

#### sysdateフィールド

AQLで記述するWHERE句では、**sysdate**フィールドを**getdate()**関数に置換する必要があります。

#### NULL

「数値」型のフィールドが未入力（値がNULL）の場合は、値は"0"に設定されます。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。"Location=0"または"LocaId=0"などのようにします。

## Database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） [ 献 43]」章の「取り込み用ルール [ 献 46]」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、**Reconciliation**と**Advanced reconciliation**タブ内で照合更新用のパラメータを入力します。

これら照合更新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） [ 献 43]」章の「取り込み用ルール [ 献 46]」節内「照合更新 [ 献 47]」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データソースの宣言に関する詳細

### ODBCデータソースの宣言に関する詳細

Databaseコネクタを含むシナリオを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

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 Administrator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、**System DNS**タブ内にあるかどうか確認します。

System DNSタブ内に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 - Databaseコネクタ

このコネクタ用のシナリオはありません。

---

## XMLリスニングコネクタ

オプションコネクタ

- 生成（ソース）

受信するクエリを使用することでXMLリスニングコネクタが行える操作を以下に挙げます。

- これらのクエリからドキュメントを生成します。
  - クエリを処理する別のコネクタとインタラクションします。
- XMLデータは、サーブレットへ送信された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。



## インタラクションモードのXMLリスニングコネクタ

インタラクションモードにあるコネクタは、クライアントが送信するクエリをリスンします。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

コネクタの制約点は、使用するオプションWebリソースにより変化します。

## サーブレットについて

Webサーバ内に統合されているサーブレットは、以下のタスクを行います。

- 受信HTTP要求の処理
- 受信要求からの適切なXMLデータの展開
- 必要な場合のクライアントソケットの作成
- 作成したソケットを経由した、XMLドキュメントのXMLリスニングコネクタへの送信

サーブレットはコネクタと通信する際、ソケットパラメータ（ポート、ホスト）を必ず把握しているため、XMLリスニングコネクタを使用すると、以下に挙げる書式でサーブレットのHTTP POST要求が作成されます。

```
POST servletURL?socketHost=string?socketPort=string HTTP/1.1
Host: webServerHost
```

実際にはサーブレットはHTTP POSTクエリをサポートしている必要があります。

URL `http://www.peregrine.com/conit/connector` がクライアントコンピュータからデータを受信したときに、'localhost'ホストのポート'1700'のソケットを作成するように指定するには、HTTP POSTは以下のようになります。

```
POST conit/connector?socketHost=localhost?socketPort=1700 HTTP/1.1
Host: www.peregrine.com
```

## 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 5.2. XMLリスニングコネクタの設定

|                                                                 |
|-----------------------------------------------------------------|
| <a href="#">Name and describe the connector [ 献 16]</a>         |
| <a href="#">Configure connection [ 献 217]</a>                   |
| <a href="#">Configure the processing of XML events [ 献 217]</a> |
| <a href="#">Configure the JVM [ 献 217]</a>                      |

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。



## Configure connection

### 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。

#### ■ Port

XMLリスニングコネクタが使用するリスンポートを入力します。

このポートを使用することでWebサーバは、WebサーバとXMLコネクタ間の通信チャンネル（ソケット）を作成できます。

オプション**Link the connection to a Web resource**はデフォルトで選択されています。このオプションを無効にすることで、Webサーバにリンクしないでコネクタを開くことができるようになります。

#### ■ URL

XMLイベントを生成するHTTP要求（POSTまたはGET）を処理するサーブレットのURLを指定します。

#### ■ Maximum number of connections

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。

このフィールドは、詳細モードで設定を行う場合のみ利用できます。

**Refuse connection after maximum limit reached**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。

## Configure the processing of XML events

この枠では要求の処理用のパラメータを入力できます。

#### ■ クエリ処理のポリシー

コネクタがHTTP要求を処理する際のモードを指定します。**Publication**と**Interaction**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。

#### ■ インタラクション用コネクタ

インタラクション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。

## Configure the JVM

このページが表示されるのは、設定ウィザードのChoose a DTD/XSD [ 献 87] ページでXSDを選択している場合のみです。

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

## XMLリスニングコネクタの使用

### 発行

XMLリスニングコネクタが発行モードである場合、XMLリスニングコネクタが受信するXMLドキュメントは、コネクタが生成したものとみなされ、他のドキュメントと同様に処理されます。

処理が正常終了すると、生成ドキュメントが返されます。処理が失敗すると、エラーメッセージが返されます。

発行ドキュメントは以下のように作成されます。

```
<DocToPublish>
<STRUCTURE name='Employee'>
<ATTRIBUTE name='FirstName'>John</ATTRIBUTE>
<ATTRIBUTE name='LastName'>Smith</ATTRIBUTE>
</STRUCTURE>
</DocToPublish>
```

The configuration file for publications or the 'SchemaFile' property is supplied to Connect-It in order to validate the publication. For example:

```
<PublishingSchema>
<STRUCTURE name='Employee'>
<ATTRIBUTE name='FirstName' type='String'/>
<ATTRIBUTE name='LastName' type='String'/>
</STRUCTURE>
<STRUCTURE name='amAsset'>
<ATTRIBUTE name='AssetTag' type='String'/>
<ATTRIBUTE name='dtLastModif' type='Date'/>
</STRUCTURE>
<STRUCTURE name='amProduct'>
<ATTRIBUTE name='BarCode' type='String'/>
</STRUCTURE>
</PublishingSchema>
```

#### 注意:

送信されるドキュメントが正しく処理されるためには、設定ファイルと完全に互換である必要があります。

### 対話

XMLリスニングコネクタがインタラクティブモードにあるとき、XMLドキュメントはターゲットリソースクエリをバインド（つまり、結果セットタイプ操作を実行）する必要があります。

## 例

デフォルトターゲットデータベースからの'AmAsset'タイプレコードの取得

```
<operation>
<STRUCTURE name='AmPortfolio'>
<ATTRIBUTE name='AssetTag' type='String'/>
<ATTRIBUTE name='FullName' type='String'/>
</STRUCTURE>
</operation>
```

'count'タイプ属性を指定した場合、結果として生成されるドキュメントの最大サイズが設定されます。

## 例

デフォルトターゲットデータベースからの最初の10レコードの取得

```
<operation count='10'>
<STRUCTURE name='AmPortfolio'>
<ATTRIBUTE name='AssetTag' type='String'/>
<ATTRIBUTE name='FullName' type='String'/>
</STRUCTURE>
</operation>
```

'target'属性を指定した場合、ウィザードで指定されているコネクタの名前ではなく、指定したコネクタの名前が使用されます。

## 例

コレクション要素を取得します。

XMLリスニングコネクタが受信した.xmlファイルにコレクションが存在する場合、.xml発行の設定ファイルは、以下の形式を持つ必要があります。

```
<PublishingSchema>
....
<COLLECTION name='<COLLECTION_NAME>'>
<STRUCTURE name='<COLLECTION_NAME>'>
...
</STRUCTURE>
</COLLECTION>
...
</PublishingSchema>
```

データを正しく処理するには、コレクションに含まれる構造体の名前がコレクション自体の名前と同一である必要があります。

結果として、処理される.xmlファイルは設定ファイル中で定義されている形式に従う必要があります。

```
<Document>
...
<COLLECTION name='<COLLECTION_NAME>'>
<STRUCTURE name='<COLLECTION_NAME>'>
...
</STRUCTURE>
<STRUCTURE name='<COLLECTION_NAME>'>
...
</STRUCTURE>
<STRUCTURE name='<COLLECTION_NAME>'>
...
</STRUCTURE>
...
<STRUCTURE name='<COLLECTION_NAME>'>
...
</STRUCTURE>
</COLLECTION>
...
</Document>
```

生成されたXMLドキュメントの例：

```
<Document>
<STRUCTURE name='Employee'>
<ATTRIBUTE name='Name'>Bailly</ATTRIBUTE>
<ATTRIBUTE name='BarCode'>DEMO-U061</ATTRIBUTE>
<ATTRIBUTE name='Field1'>test</ATTRIBUTE>
<COLLECTION name='Group'>
<STRUCTURE name="Group">
<ATTRIBUTE name='BarCode'>EG000002</ATTRIBUTE>
</STRUCTURE>
<STRUCTURE name="Group">
<ATTRIBUTE name='BarCode'>DEMO-EG03</ATTRIBUTE>
</STRUCTURE>
</COLLECTION>
</STRUCTURE>
</Document>
```

```
<Document>
<STRUCTURE name='computers'>
<ATTRIBUTE name='test'>tt</ATTRIBUTE>
<COLLECTION name='computer'>
<STRUCTURE name='computer'>
<ATTRIBUTE name='AssetTag'>XX1</ATTRIBUTE>
<ATTRIBUTE name='PhysicalAddress'>00:00:00:00:00</ATTRIBUTE>
```

```
</STRUCTURE>
<STRUCTURE name='computer'>
<ATTRIBUTE name='AssetTag'>XX2</ATTRIBUTE>
<ATTRIBUTE name='PhysicalAddress'>00:00:00:00:01</ATTRIBUTE>
</STRUCTURE>
</COLLECTION>
</STRUCTURE>
</Document>
```

記述ファイルは以下のようになります。

```
<PublishingSchema>
<STRUCTURE name="computers">
<ATTRIBUTE name="test" type="String" />
<COLLECTION name="computer">
<STRUCTURE name="computer" >
<ATTRIBUTE name="AssetTag" type="String" />
<ATTRIBUTE name="PhysicalAddress" type="String" />
</STRUCTURE>
</COLLECTION>
</STRUCTURE>
<STRUCTURE name='Employee'>
<ATTRIBUTE name='Name' type='String'/>
<ATTRIBUTE name='BarCode' type='String'/>
<ATTRIBUTE name='Field1' type='String'/>
<COLLECTION name='Group'>
<STRUCTURE name="Group">
<ATTRIBUTE name='BarCode' type='String'/>
<ATTRIBUTE name='Name' type='String'/>
</STRUCTURE>
</COLLECTION>
</STRUCTURE>
</PublishingSchema>
```

## 例

```
<operation target='ODBC'>
<STRUCTURE name='AmPortfolio'>
<ATTRIBUTE name='AssetTag' type='String'/>
<ATTRIBUTE name='FullName' type='String'/>
</STRUCTURE>
</operation>
```

取得したドキュメントの体裁の制御用に、ルールを指定することができます。

## 例

WHERE句を使用したレコードの取得

```
<operation>
<STRUCTURE name='AmPortfolio'>
<ATTRIBUTE name='AssetTag' type='String'/>
<ATTRIBUTE name='FullName' type='String'/>
</STRUCTURE>
<layer>
<STRUCTURE name='AmPortfolio'>
<where>AssetTag LIKE 'UTL%'/</where>
</STRUCTURE>
</layer>
</operation>
```

この例では、クライアントソケットに対し、XMLドキュメントが要素のコレクションとして返されます。

```
<COLLECTION>
<STRUCTURE name='AmPortfolio'>
<ATTRIBUTE name='AssetTag'>UTL000338</ATTRIBUTE>
<ATTRIBUTE name='FullName'>/GENASSET-338/UTL000338</ATTRIBUTE>
</STRUCTURE>
<STRUCTURE name='AmPortfolio'>
<ATTRIBUTE name='AssetTag'>UTL000442</ATTRIBUTE>
<ATTRIBUTE name='FullName'>/GENASSET-442/UTL000442</ATTRIBUTE>
</STRUCTURE>
...
</COLLECTION>
```

### 注意:

これらすべての例では、'operation'クエリのルートタグは、クエリを実行するためにターゲットコネクタに対して使用する操作に対応します。ターゲットコネクタがJavaコネクタであり、コネクタ設定ウィザードの**Connector for interactions**フィールドに対応する場合、これに該当します。非Javaのコネクタの場合、このタグは'NULL'値を持つ必要があります。

```
<null>
...
</null>
```

## 例

ターゲットコネクタがJavaコネクタではない場合のレコードの取得

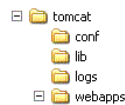
```
<null>
<STRUCTURE name='AmPortfolio'>
<ATTRIBUTE name='AssetTag' type='String'/>
<ATTRIBUTE name='FullName' type='String'/>
</STRUCTURE>
<layer>
<STRUCTURE name='AmPortfolio'>
<where>AssetTag LIKE 'UTL%'</where>
</STRUCTURE>
</layer>
</null>
```

## 例 - Tomcat Webサーバ

この節には、展開テストとサブレットテストが記載されています。クライアントクエリがHTTP GETタイプ要求であること、および、適切なXMLドキュメントがクエリの'document'パラメータに含まれていることを前提条件とします。

### Tomcatサーバ上のサブレットの展開

以下の図に、この例で使用するTomcatインストールフォルダの構造を図示します。



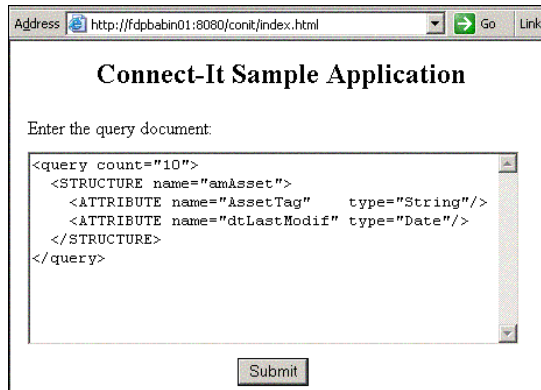
datakitに含まれるWebアプリケーション（.war）を**webapps**フォルダにコピーする必要があります。このフォルダは、Tomcatサーバがサブレットの展開に使用します。

### XMLリスニングコネクタのテスト

- 1 Tomcat Webサーバを開始します
- 2 Connect-ItでXMLリスニングコネクタを使用してシナリオを作成します。以下のパラメータを使用できます。

```
Port=1024
URL=http://localhost:8080/conit/queryDocument
Policy=Interaction
Resource=AssetManagement
```

- 3 シナリオにスケジュールを適用します
- 4 インターネットブラウザで、URL `http://localhost:8080/conit`を入力します。  
以下のHTMLページが表示されます。



- 5 ドキュメントを入力して送信します
- 6 シナリオを閉じてTomcatサーバを停止します

## 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ドキュメントの形式で処理するクエリ内にあります。

## XMLリスニングコネクタが発行する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

このコネクタ用の付属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-mail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（*finrequc.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。

- Eメールコネクタの既知の制約点
- Eメールコネクタの設定手順

### 警告:

ConnectIt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-mail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

Hewlett-Packard Development Company, L.P.サポートにある互換対応表を参照してください。

### 警告:

英語のUNICODEバージョンは、Hewlett-Packard Development Company, L.P.サポートから入手可能です。UNICODEバージョンが要件を満たすかどうかサポートが判断できるよう、処理が必要なコネクタとコードページを指定してください。

## その他の制約点

e-mailコネクタの様々な制約点は以下の通りです。

- Eメールの 'Priority'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-mailコネクタが適切に機能するためには、ConnectIt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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 5.3. Eメールコネクタ（受信）の設定

Name and describe the connector [ 献 16]
Define the connection parameters [ 献 226]
メッセージ内容の処理 [ 献 230]
Define post-processing actions [ 献 230]

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Define the connection parameters

このページでは、Eメールコネクタ（受信）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、**Engine type**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）コネクタで使用できるプロトコルは以下の通りです。

- Lotus VIM (Vendor Independent Messaging)  
コネクタを変更する前に、Lotusフォルダにあるvim32.dllファイルへのアクセスパスを、システム環境変数内で宣言する必要があります。

- POP3 (Post Office Protocol)
- Microsoft MAPI (Messaging Application Program Interface)
- Microsoft Extended MAPI (Extended Messaging Application Program Interface)
- IMAP

### 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。

- Login  
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。
- Password  
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

VIM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。

- Retrieve unread e-mail messages only
- Use DOS/Windows text formatting: Carriage return (CR) and line return (LF)  
外部アプリケーションが、DOSまたはWindows 32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。
- Activate the conversion of the e-mail body from the Lotus Notes internal code page  
メッセージテキスト中の文字が正しく変換されていない場合にこのオプションを選択します。

### 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 server  
POP3サーバの名前を指定します。
- Connection port  
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。
- Login  
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。
- Password

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。

- Use DOS/Windows text formatting: Carriage return (CR) and line return (LF)

外部アプリケーションが、DOSまたはWindows 32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。

- Profile  
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。
- Password  
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。

## Eメール処理のオプション

MAPI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。

- Retrieve unread e-mail messages only
- Use DOS/Windows text formatting: Carriage return (CR) and line return (LF)

外部アプリケーションが、DOSまたはWindows 32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。

## 拡張MAPI

拡張MAPIプロトコルでは、以下のフィールドが入力されている必要があります。

- Profile  
コンピュータにインストールされている拡張MAPIメッセージングシステムにユーザがアクセスできるプロファイルを示します。
- Password

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。

- **Directory**

ディレクトリのパスを入力します。

ローカルフォルダのパスまたは、Eメールサーバ上のフォルダへのパスを入力できます。

サーバにあるフォルダの場合の一例を以下に挙げます。

ローカルフォルダの例：

## Eメール処理のオプション

拡張MAPI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。

- **Retrieve unread e-mail messages only**
- **Use DOS/Windows text formatting: Carriage return (CR) and line return (LF)**

外部アプリケーションが、DOSまたはWindows 32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。

## IMAP

IMAPプロトコルを選択する場合は、以下のフィールドに入力します。

- **IMAP Server**  
FTPサーバ名を指定します。
- **Login**  
IMAPサーバへのアクセスを許可するログインを入力します。
- **Password**  
ログインに対応するパスワードを入力します。このパスワードは入力時には表示されず、暗号化された形式で格納されます。

## Test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 **Test**をクリックします。

**Test the connection** 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 **Close** をクリックして設定ウィザードへ戻ります。

## メッセージ内容の処理

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。



### 注意:

Eメールの内容が解析され、最初のXMLタグ（<?xml version="1.0">）より前に含まれるテキストと最後のXMLタグが自動的に削除されます。この解析により、メールサーバがXML要素の自動的に追加した内容の隔離などができます。

### DTDファイルを使ってメッセージの内容を処理する

- 1 **Process contents of e-mail body** オプションを選択します。
- 2 **DTD file** 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 **Process contents of e-mail attachments** オプションを選択します。
- 2 **DTD file** 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。

## Define post-processing actions

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。

### Apply actions just after fetching messages

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。

### Available post-processing actions

**In case of successful processing of message** と **On failure in processing a file** 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。

- **Do not modify e-mail messages from server**
- **Flag e-mail messages as read**

## ■ Delete e-mail messages from server

### 処理後のアクションの使用

処理後のアクションを使用するには：

- ソースコネクタが生成したドキュメントタイプの*MailInfo.UniqueID*構造体と*SuccessReport*ドキュメントタイプの*MailInfo.UniqueID*構造体との間にマッピングを作成します。





Eメールコネクタが発行するドキュメントタイプについては、本章の「Eメールコネクタ（受信）が生成するドキュメントタイプ [ 献 234]」の節を参照してください。

## Eメールコネクタ（送信）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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 5.4. Eメールコネクタ（送信）の設定

 Name and describe the connector [ 献 16]
 Configuring the connection [ 献 231]
 Define sending frequency [ 献 233]
 Parallelize consumption（詳細モード） [ 献 41]

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Configuring the connection

**Configure the connector**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、**Engine type**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)  
コネクタを変更する前に、Lotusフォルダにあるvim32.dllファイルへのアクセスパスを、システム環境変数内で宣言する必要があります。
- Microsoft SMTP (Simple Mail Transfer Protocol)
- Microsoft MAPI (Messaging Application Program Interface)

- Microsoft Extended MAPI (Extended 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。

- Login  
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。
- Password  
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

必要に応じて、以下のオプションのいずれか1つを選択します。

- Request a receipt
- Detect HTML messages

### 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 server  
SMTPサーバの名前を指定します。
- Connection port  
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。
- Display name  
メッセージの受信時に、メッセージの'**From**'フィールドに表示される名前を指定します。
- Reply address  
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（Reply@hp.comなど）を入力します。

### MAPI / Extended MAPI

MAPIおよび拡張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。

- Login  
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。
- Password  
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。



## Test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 **Test**をクリックします。

**Test the connection**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 **Close**をクリックして設定ウィザードへ戻ります。

## Detect HTML messagesオプション

このオプションで、HTMLタグを含むメッセージを送信できます。HTMLコードは解釈され、Eメールの本文のソースフォーマットでは表示されません。

## Define sending frequency

**Configure the connector**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。

- **Number of messages**

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、30 s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:**

**Pause length**フィールドと**Number of messages**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。

- **Pause duration**

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。

## Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Parallelize consumption (詳細モード) [ 献 41]」の節を参照してください。

## Eメールコネクタ（受信）が生成するドキュメントタイプ

e-mail（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）コネクタが生成する **InMailMessage** ドキュメントタイプ

Eメールコネクタ（受信）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 5.5. Eメールコネクタ（受信）が生成する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
<b>InMailMessage</b> ルートノード	メッセージ本文 ( <b>Body</b> フィールド) メッセージの日付 ( <b>Date</b> フィールド) メッセージの優先順位 ( <b>Priority</b> フィールド) メッセージの件名 ( <b>Subject</b> フィールド)
<b>From</b> 構造体	メッセージ送信者のEメールアドレス ( <b>Address</b> フィールド) メッセージ送信者の名前 ( <b>Name</b> フィールド) メッセージ送信者のタイプ ( <b>Type</b> フィールド)
<b>MailInfo</b> 構造体	以下のフィールドが含まれます。 <ul style="list-style-type: none"><li>■ <b>UniqueID</b> 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。このフィールドは、メールメッセージの一意の識別子です</li><li>■ <b>SignedBy</b> (<i>VIM</i> プロトコル) : メールに署名したユーザ</li><li>■ <b>CertifiedBy</b> (<i>VIM</i> プロトコル) : サーバが発行した証明書</li><li>■ <b>Trusted</b> (<i>VIM</i> プロトコル) : メッセージは署名され証明されているが、署名を検証できなかった状態。</li></ul> これにより、コネクタが生成したメッセージを一意な方法で識別できるようになります。

## 生成用ドキュメントタイプの構成要素

### Attachmentコレクション

## 要素

添付ファイルの内容は3つのフィールドに分かれます。

- 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

### 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)

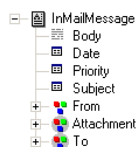
## Eメールコネクタ（受信）に生成されるメッセージ本文の処理

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メールコネクタ（受信）」の設定 [ 献 226 ] 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。

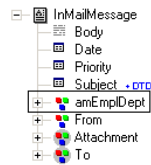
### 例 5.1. DTDにより拡張されたドキュメントタイプの例

E-mail（受信）- Asset Management用例シナリオ（newemplo.scnファイル）では、DTDnewemplo.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



### MAPIプロトコルを使用したEメールコネクタ（受信）のDTDの解析

**From**構造体および**To**と**Cc**の各コレクションの**Address**要素には、メッセージの送信者や受信者のEメールアドレスが含まれます。これらのアドレスはSMTPアドレスですが、送信者と受信者が同一サーバに属す場合、Eメールサーバ固有の内部アドレスとすることもできます。

たとえばMicrosoft Exchangeサーバの場合、以下のようにアドレスはSMTPアドレスにもMicrosoft Exchangeアドレスにもなります。

- SMTP:john.doe@company.com（外部ユーザ）
- EX:/O=HP/CN=RECIPIENTS/CN=JDOE（内部ユーザ）

**From**構造体および**To**と**Cc**の各コレクションには、**SmtAddress**要素が追加されています。この要素には、メッセージが内部で送信されたかどうかに関わらず、SMTPアドレスが含まれます。

このため、先の例での**SmtAddress**要素の値を以下に挙げます。

- john.doe@company.com（外部ユーザ）
- john.doe@company.com（内部ユーザ）

### Eメールコネクタ（送信）が取り込むドキュメントタイプ

e-mail（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-mail 付属シナリオ（finreq.scn）の、Asset Managementコネクタ）

### Eメールコネクタ（送信）が取り込むドキュメントタイプ

Eメールコネクタ（送信）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 5.6. Eメールコネクタ（送信）が取り込む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
<b>OutMailMessage</b> ルートノード	メッセージ本文 ( <b>Body</b> フィールド) メッセージの優先順位 ( <b>Priority</b> フィールド) メッセージの件名 ( <b>Subject</b> フィールド)
<b>From</b> 構造体 (SMTPプロトコルのみ)	メッセージ送信者のEメールアドレス ( <b>Address</b> フィールド) メッセージ送信者の名前 ( <b>Name</b> フィールド) メッセージ送信者のタイプ ( <b>Type</b> フィールド)
<b>Attachment</b> 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"> <li>■ 添付ファイルのデータを含む<b>BlobToSend</b>フィールド</li> <li>■ ファイル名を含む<b>FileToSend</b>フィールド</li> </ul>
<b>Bcc</b> コレクション	メッセージの <b>Bcc</b> 受信者のEメールアドレス ( <b>Address</b> フィールド) メッセージの <b>Bcc</b> 受信者の名前 ( <b>Name</b> フィールド) <b>Bcc</b> のタイプ ( <b>Type</b> フィールド)
<b>Cc</b> コレクション	メッセージの <b>Cc</b> 受信者のEメールアドレス ( <b>Address</b> フィールド) メッセージの <b>Cc</b> 受信者の名前 ( <b>Name</b> フィールド) <b>Cc</b> のタイプ ( <b>Type</b> フィールド)
<b>To</b> コレクション	メッセージ受信者のEメールアドレス ( <b>Address</b> フィールド) メッセージ受信者の名前 ( <b>Name</b> フィールド) メッセージ受信者への送信のタイプ ( <b>Type</b> 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メールコネクタが発行する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。

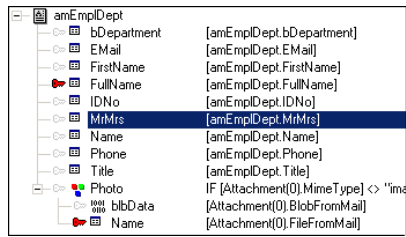
## 受信される添付ファイルのマッピング

添付ファイルは、e-mail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-mail (fetching) to 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 **amEmplDept**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。



amEmplDept	
bDepartment	[amEmplDept.bDepartment]
EMail	[amEmplDept.EMail]
FirstName	[amEmplDept.FirstName]
FullName	[amEmplDept.FullName]
IDNo	[amEmplDept.IDNo]
MlMrs	[amEmplDept.MlMrs]
Name	[amEmplDept.Name]
Phone	[amEmplDept.Phone]
Title	[amEmplDept.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を以下に挙げます。

- mail\mailac36\finconfi.scn [ 献 377]
- mail\mailac36\finreque.scn [ 献 377]
- mail\mailac36\newemplo.scn [ 献 378]
- mail\mailac43\finconfi.scn [ 献 378]
- mail\mailac43\finreque.scn [ 献 378]
- mail\mailac43\newemplo.scn [ 献 379]

---

## 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コネクタについて

LDAPコネクタは、LDAPv3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、名前付けコンテキストなど) を提供します。

### 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。

### 運用属性のカスタマイズ

Red Hat Directory LDAPサーバの場合、セキュリティや個別ユーザに関連する情報の保存に、多数の運用属性が使用されています。

LDAPコネクタが使用する属性は、config/ldap/ldapball.cfgファイルに保存されます。

例：

```

{ STRUCT VIRTUAL

{ STRUCT AllTables
TABLE = self
TIMESTAMP modifyTimestamp
TIMESTAMP createTimestamp
{ STRING distinguishedName
INSERTONLY = 1
KEY = 4
}
{ ATTRIBUTE AllFields
FIELD = self
Exception = modifyTimestamp, createTimestamp, distinguishedName
CIRCULAR = AllTables
}
}

```

このファイルを編集して設定し、属性のリストを追加できます。  
Novell Directoryサーバ用の*nsaccountlock*属性を追加するには：

```

{ STRUCT VIRTUAL

{ STRUCT My_Dummy_Object
ObjectClass = *
STRING nsaccountlock
}

{ STRUCT AllTables
TABLE = self
TIMESTAMP modifyTimestamp
TIMESTAMP createTimestamp
{ STRING distinguishedName
INSERTONLY = 1
KEY = 4
}
{ ATTRIBUTE AllFields
FIELD = self
Exception = modifyTimestamp, createTimestamp, distinguishedName
CIRCULAR = AllTables
}
}

```

これらの属性にはアスタリスク(\*)をプレフィックスし、LDAP検索で考慮されないようにする必要があります。



## 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

このコネクタに関する一部のオプションを有効にするには、**Edit/ Options/ Connector**メニューを使用します。詳細については、『ユーザガイド』、「シナリオビルダ」、「コネクタ」を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 5.7. LDAPコネクタの設定

 Name and describe the connector [ 献 16]
 LDAPコネクタの設定 [ 献 241]
 Advanced configuration (詳細モード) [ 献 243]
 Configure the reconnection parameters (詳細モード) [ 献 29]
 Determine server delay (詳細モード) [ 献 31]
 Configure schedule pointers (詳細モード) [ 献 33]
 Configure the cache (詳細モード) [ 献 36]
 Define document types (詳細モード) [ 献 39]
 Parallelize consumption (詳細モード) [ 献 41]

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## LDAPコネクタの設定

**Basic configuration of the connector**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

### LDAP server name

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。

### Server connection port

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。）

### Login

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。

**Windows Active Directory** オプションを選択する場合、このログインは `john.doe@hp.com` のように [氏名] @domain.name というシンタックスを使用する必要があります。



#### 注意:

iPlanet LDAPサーバの場合、ログインはユーザの識別名です。

### Password

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。

### Indicate the server type.

利用可能なサーバタイプを以下に挙げます。

- Novell Directory Service、Netscape Directory Serverなど
- Microsoft Exchange
- IBM z/OS Security Server
- Microsoft Active Directory
- OpenLDAP

### SSL connection

セキュアな接続プロトコルを使用する場合、このオプションを選択します。

使用オペレーティングシステムに応じて、以下の手順を実行します。

- **Solaris** : Solaris 9より前の全バージョンで、Solaris 8用のパッチ112438を適用する必要があります。このパッチは、[www.sunsolve.com](http://www.sunsolve.com)から入手可能です。Solarisバージョン 2.6および2.7の修正が入手可能です。
- **AIX** : AIX 5.2より前の全バージョンで、デフォルトで周辺ファイル/dev/random および/dev/urandomがインストールされていない場合、prngdパッケージをダウンロードする必要があります。このファイルはIBM Webサイトから入手可能です  
(<http://www.ibm.com/servers/aix/products/aixos/linux/download.html>) 。

### Specify the codepage of the server

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。

## Test

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。
- **Test**をクリックします。

**Test the connection**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。

- **Close**をクリックして設定ウィザードへ戻ります。

## Advanced configuration (詳細モード)

このページでは以下の内容を指定します。

- **The size of the page**

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 (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。この場合は、このフィールドで指定します。

識別名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。

識別名を取得するには、ルートDSE (Directory Service Entries) の **subschemaSubentry** エントリを参照してください。

- **The format of the dates**

日付の形式については、本節内の「追加情報 [ 献 249]」の節の「LDAP日付形式 [ 献 252]」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GMT 0形式であることを示す必須記号

`%[number][symbol]`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。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

#### サーバの現在日時

- **Attribute Name** フィールドを使用して、サーバの現在日時を持つ属性を入力します。
- **DN** フィールドは、属性の識別子（識別名）の入力に使用されます。デフォルトのエントリは *currentTime* です。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

#### Last modification field

一部のドキュメントタイプには、データベーススキーマのレコードの最終更新日を示すフィールドがあります。通常の場合、**Modified-date** フィールドが使用されます。しかし、データベーススキーマによってはこのフィールド名が異なることがあります。

このフィールドの編集可能部分で、LDAPコネクタがスケジュールのポイントとして使用する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。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

## Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Parallelize consumption (詳細モード) [ 献 41]」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) [ 献 43]」章の「取り込み用ルール [ 献 46]」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。

- **Reconciliation**と**Advanced reconciliation**タブ内で照合更新用のパラメータを入力します。  
これら照合更新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) [ 献 43]」章の「取り込み用ルール [ 献 46]」節内「照合更新 [ 献 47]」を参照してください。
- SCOPE句とSEARCH DN句を作成します。

### SCOPE句

この句は、DN (識別名) で識別されるLDAPディレクトリツリー内のエン트리ポイントに応じて、要求の適用範囲を決定します。

- **Base**  
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。
- **First level**  
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。
- **Recursive**  
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

この句を使用すると、接続先LDAPサーバのディレクトリのツリー構造内にある、エン트리ポイントのDN (識別名) を指定できます。コネクタを起動すると、ディレクトリによりいくつかのエン트리ポイントが示されます。これらは、LDAPコネクタが公開する命名コンテキストです。

名前付けコンテキストは、**SEARCH DN**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:

**SEARCH DN**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、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

この句は、DN（識別名）で識別されるLDAPディレクトリのツリー構造内のエントリポイントに応じて、要求の適用範囲を決定します。

- Base  
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。
- First level  
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。
- Recursive

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

この句を使用すると、接続先LDAPサーバのディレクトリのツリー構造内にある、エン트리ポイントのDN（識別名）を指定できます。コネクタを起動すると、ディレクトリによりいくつかのエン트리ポイントが示されます。これらは、LDAPコネクタが公開する命名コンテキストです。

名前付けコンテキストは、**SEARCH DN**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:

**SEARCH DN**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트리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 5.8. 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 filertype value
filertype = 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)
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 5.9. マップテーブルファイル (ASCII値の文字)

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：RDN *cn*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、次のようには書きません。	( & ( 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

### コレクションの単独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리는、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*Mapping source elements to destination elements*の章の「ドキュメントタイプのマッピング」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
- 2 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：

- 1 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。

- 2 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**Mapping script**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：

- 1 マッピング編集ウィンドウ内で、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。
- 2 **Mapping script**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 *OrganizationnalPerson* クラスのうち、どれを使用するか。

これらの情報を取得するには：

- 1 作成または編集したシナリオで、LDAPコネクタを選択するかまたは作成します。
- 2 **Document types** 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**timestamp**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**で表現されます。

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、**timestamps**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を以下に挙げます。

- `ldap\ldapac36\complete.scn` [ 献 398]
- `ldap\ldapac36\simple.scn` [ 献 399]
- `ldap\ldapnote\names.scn` [ 献 399]

---

## 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**Command-line**コネクタの内容を説明します。

- 取り込みモードで、**Command-line**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、**Command-line**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 エラー出力

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、**Command-line**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

**Command line**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でコマンドラインを起動できます。

UNIXでは、コネクタはコマンドの戻り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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 5.10. コマンドラインコネクタの設定

☒ Name and describe the connector [ 献 16]
☒ Execution frequency [ 献 253]
☒ Parallellize consumption (詳細モード) [ 献 41]

**Connect-It**のコネクタの設定は、ウィザードで実行できます。

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「**Name and describe the connector** [ 献 16]」の節を参照してください。

### Execution frequency

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。

- **Number of documents**

2回の一時的停止時間の間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**の一時的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**Pause duration**のフィールド値と**Number of documents**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。

- **Pause duration**

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。

- **Timeout**

**Kill the process if it has not finished before timeout** 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、**Kill the process if it has not finished before timeout** オプションは使用不可能です。

### Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15 ]」章の「Parallelize consumption (詳細モード) [ 献 41 ]」の節を参照してください。

## 発行ドキュメントタイプ

Command-line コネクタには 2 つの発行ドキュメントタイプがあります。

- **Command** ドキュメントタイプ

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。この発行ドキュメントタイプにより、Command-line 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。この発行ドキュメントタイプにより、Command-line 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\HP OpenView\AssetCenter 4.40 en\bin  
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：explorer.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 *Start in* 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。

## BIOSコマンド

一部のコマンドは実行形式であり、その他のコマンドはBIOSコマンドです。

BIOSコマンドをコマンドラインコネクタで使用するには、**cmd.exe**実行形式と併せ以下のシンタックスを使用する必要があります。

```
cmd /C move d:\HR\Employee.xml d:\HR\done\Employee.xml
```

Connect-Itでは、**Name**要素は"cmd /C move"でマップされ、**arg**要素は"d:\HR\Employee.xml d:\HR\done\Employee.xml"でマップされます。

## 取り込み用ルール

Command-line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編集ウィンドウにある**Synchronous execution**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 - Command-lineコネクタ

このコネクタ用の付属シナリオはありません。

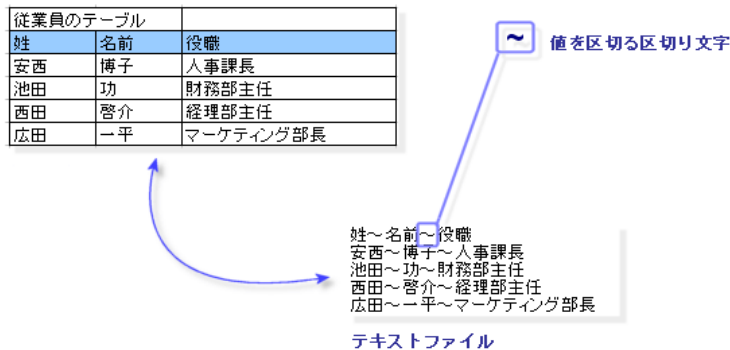
## 区切りテキストコネクタ

ベースコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

Delimited-text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. 区切りテキストコネクタ - テキストファイル内のデータの記録法



Delimited-text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（Database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 (Database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。

text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)。

## 区切り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、**Date型**と**Date and Time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。

*string*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 (読み取り) 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

このコネクタに関する一部のオプションを有効にするには、**Edit/ Options/ Connector**メニューを使用します。詳細については、『ユーザガイド』、「シナリオビルダ」、「コネクタ」を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 5.11. 区切りテキストコネクタ (読み取り) の設定

 Name and describe the connector [ 献 16]
 Choosing a processing mode [ 献 258]
 Selecting a connection protocol [ 献 258]

☰ Choose a file or folder  
[ 献 258]

☰ HTTP Webサイトへの接続  
[ 献 259]

☰ Connect to the FTP server  
[ 献 262]

☰ 処理後のアクション [ 献 259]

☰ Define post-processing  
actions [ 献 263]

☰ Choose a description file [ 献 264]



### 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、設定ウィザードでファイルやフォルダのパスを指定しなくてはいけない場合があります。この場合、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。

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Choosing a processing mode

このページでは、コネクタを読み取りモードまたは書き込みモードのどちらで使用するかを指定できます。

### Selecting a connection protocol

以下の3つのオプションがあります。

- HTTP Web site
- FTP server
- Local or network file(s)

### Choose a file or folder

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

### Choose a file or folder

このページでは、コネクタが読み取るtext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。

- Read a file

## ■ Read files in a folder

### Read a file

このオプションを選択すると、ウィザードで、コンピュータやネットワーク上に存在するテキストファイルに対するパスの入力が求められます。

### Read files in a folder

このオプションを選択すると、ウィザードで、以下のフィールドへの入力が求められます。

#### ■ Folder name

ファイルを含むフォルダのパスを入力します。

#### ■ Extension

コネクタが読み取るファイルの拡張子を入力します。'txt'などです。

### Read the sub-folders

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、Connect-Itは次の3つのオプションを提示します。

#### ■ Leave it in the folder

#### ■ Delete it from the folder

#### ■ Move it to another folder

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。

#### ■ Address

[プロトコル]://[アドレス]:[ポート]/[パス]のように、タイプの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。

- Login  
サイトへのアクセスを許可するログインを入力します。
- Password  
ログイン用のパスワードを入力します。

### Secured connection (HTTPS protocol)

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 [protocol] の部分がHTTPSプロトコルでない場合でも、**Secured connection (HTTPS protocol)** 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

Address	Secured connection (HTTPS protocol)	Connection mode
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

### Client certificate

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、Scenario Builder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

configuration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Root console/ Certificates - current user/ Personal/ Certificatesフォルダにあるリストと一致します。

Windows XPでこのフォルダに証明書を追加するには：

- 1 Windowsの**Start**メニューで**Run**を選択します。
- 2 **Open** フィールドに**MMX**と入力します。
- 3 **File/ Add/Remove Snap-in**を選択します。
- 4 表示されるウィンドウで**Add**をクリックします。
- 5 表示されるウィンドウで**Certificates**を選択します。
- 6 **Add**をクリックします。
- 7 表示されるダイアログボックスで**My user account**オプションを選択します。
- 8 **Finish**をクリックします。
- 9 **OK**をクリックします。
- 10 **Certificates - Current user/ Personal/ Certificates folder**フォルダ内でファイルを追加または削除します。

#### Do not verify the server identity

セキュアな接続でのプロキシサーバをサポートするには、このオプションを使用します。

プロキシサーバがインストールされている場合、サーバ証明書に示されているアドレスは、プロキシサーバのアドレスとは異なります。このオプションを無効にして、サーバ識別の自動検証を無効にしてください。

#### Connect via a proxy

このオプションは、**HTTP**および**FTP**接続の場合に利用できます。このオプションを使用して、特定の接続に対するプロキシサーバにアクセスします。

このオプションを選択する場合、プロキシサーバの名前（またはIPアドレス）、ポートを入力し、必要であれば、**Use authentication**オプションを選択します。



#### 注意:

すべてのプロキシに同一の認証情報を使用する場合、多段プロキシがサポートされません。

#### Do not use a proxy server

HTTPタイプ接続の場合、プロキシサーバをバイパスする状況を指定できます

(**Do not use a proxy server for** フィールド)。使用する接続に応じて、HTTPタイプのアドレスを入力する必要があります。名前はスペースで区切る必要があります。

## FTP server

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

### Connect to the FTP server

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。

- **Server**  
FTPサーバ名を指定します。
- **Login**  
サイトへのアクセスを許可するログインを入力します。
- **Port**  
FTPサーバへのアクセスに使用するポートを指定します。
- **Password**  
ログイン用のパスワードを入力します。
- **Connect in passive mode**  
このオプションを選択して、パッシブモードで動作させます。
- **Connect via a proxy**  
このオプションは、HTTPおよびFTP接続の場合に利用できます。このオプションを使用して、特定の接続に対するプロキシサーバにアクセスします。このオプションを選択する場合、プロキシサーバの名前（またはIPアドレス）を入力し、必要であれば、**Use authentication**オプションを選択します。
- **Use authentication**  
このオプションが有効である場合、ユーザ名とパスワードを入力します。
- **Do not use a proxy server**  
HTTPおよびFTPタイプ接続の場合、プロキシサーバをバイパスする状況を指定できます（**List**フィールド）。使用する接続に応じて、HTTPまたはFTPタイプのアドレスを入力する必要があります。名前はスペースで区切る必要があります。

接続の有効性をテストするには、**Test**ボタンを使用します。

ウィザードの次のページでは、必要に応じて**Read files**または**Read folders**のいずれかを選択します。**Read files**オプションを指定する場合、以下の項目を指定する必要があります。

- FTPサイト上にあるフォルダのパス
- ファイル拡張子名  
デフォルトでは、このフィールドの値は`txt`です。

## Read files

このオプションを選択すると、ウィザードで、コンピュータやネットワーク上に存在する1つまたは複数のテキストファイルに対するパスの入力が求められます。

## Read folders

このオプションを選択すると、ウィザードで、以下のフィールドへの入力が求められます。

- **Path**

ファイルを含むフォルダのパスを入力します。

- **Extension**

コネクタが読み取るファイルの拡張子を入力します。

## Read the sub-folders

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、

## Define post-processing actions

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、Connect-Itは3つのオプションを提示します。

- **Leave it in the folder**

- **Delete it from the folder**

- **Move it to another folder**

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。

処理後のアクションとアーカイバルの詳細については、「コネクタの設定 [ 献 15]」の章の「[Advanced configuration \(詳細モード\)](#) [ 献 27]」の節を参照してください。

## 処理後のアクションの使用



処理後のアクションを使用するには：

- ソースコネクタが生成したドキュメントタイプの *UrlFileInfo.Path* 構造体と *SuccessReport* ドキュメントタイプの *UrlFileInfo.Path* 構造体との間にマッピングを作成します。

この構造体の詳細については、本章の「発行ドキュメントタイプ [ 献 278]」の節を参照してください。

### Choose a description file

テキストファイル进行处理する際、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 file** 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 file** 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 [ 献 272]」を参照してください。

#### 注意:

2つの区切りテキストコネクタを同一シナリオで使用する場合、もう一方のコネクタが上書きしないよう、最初のDSCファイルのバックアップコピーを作成してください。

UNICODEバージョンを使用している場合、追加オプションで使用するエンコードタイプを選択できます。

### 区切りテキストコネクタに関連するオプション

Textコネクタのオプションにアクセスするには、**Edit/ Options/ Connector/ Delimited text and XML**メニューを使用します。

FTPプロトコルでのテキストコネクタに関するオプションを以下に挙げます。

- **Copy locally the files to read from FTP server.**

このオプションを選択して、FTPサーバにあるファイルのローカルコピーを作成し、ローカルファイルからデータを読み込みます。

ネットワーク設定では、ファイルの処理に十分な長さの接続をFTPサーバで維持できない場合、このオプションを選択する必要があります。

## テキストコネクタ (書き込み) 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。



text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 5.12. 区切りテキストコネクタ（書き込み）の設定

Name and describe the connector [ 献 16]		
Processing mode [ 献 265]		
Selecting a connection protocol [ 献 265]		
Choosing a file or folder [ 献 268]	HTTP Web site [ 献 266]	FTP connection parameters [ 献 268]
Behavior between two sessions [ 献 269]	Behavior between two sessions [ 献 269]	
Choose a description file [ 献 270]		

 **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、設定ウィザードでファイルやフォルダのパスを指定しなくてはいけない場合があります。この場合、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。

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Processing mode

このページでは、コネクタを読み取りモードまたは書き込みモードのどちらで使用するのかを指定できます。

### Selecting a connection protocol

以下のオプションがあります。

- HTTP
- FTP
- Local or network file(s)

## HTTP Web site

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 connection parameters

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。

- Address  
HTTPサーバへの接続用アドレスを指定します。
- Login  
サイトへのアクセスを許可するログインを入力します。
- Port  
HTTPサーバ用の通常のポート番号は80です。
- Password  
ログイン用のパスワードを入力します。

## 書き込みコマンド

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。
- 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。

## HTTPS protocol

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。

## Client certificate

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、**Scenario Builder**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

configuration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Root console/ Certificates - current user/ Personal/ Certificatesフォルダにある一覧と一致します。

Windows XPでこのフォルダに証明書を追加するには：

- 1 Windowsの**Start**メニューで**Run**を選択します。
- 2 **Open**フィールドに**MMC**と入力します。
- 3 **File/ Add/Remove Snap-in**を選択します。
- 4 表示されるウィンドウで**Add**をクリックします。
- 5 表示されるウィンドウで**Certificates**を選択します。
- 6 **Add**をクリックします。
- 7 表示されるダイアログボックスで**My user account**オプションを選択します。
- 8 **Finish**をクリックします。
- 9 **OK**をクリックします。
- 10 **Certificates - Current user/ Personal/ Certificates folder**フォルダ内でファイルを追加または削除します。

### Do not verify the server identity

セキュアな接続でのプロキシサーバをサポートするには、このオプションを使用します。

プロキシサーバがインストールされている場合、サーバ証明書に示されているアドレスは、プロキシサーバのアドレスとは異なります。このオプションを無効にして、サーバ識別の自動検証を無効にしてください。

### Connect via a proxy

このオプションは、HTTPおよびFTP接続の場合に利用できます。このオプションを使用して、特定の接続に対するプロキシサーバにアクセスします。

このオプションを選択する場合、プロキシサーバの名前（またはIPアドレス）、ポートを入力し、必要であれば、**Use authentication**オプションを選択します。

#### 注意:

すべてのプロキシに同一の認証情報を使用する場合、多段プロキシがサポートされません。

### Do not use a proxy server

HTTPタイプ接続の場合、プロキシサーバをバイパスする状況を指定できます（**Do not use a proxy server for** フィールド）。使用する接続に応じて、

HTTPタイプのアドレスを入力する必要があります。名前はスペースで区切る必要があります。

### Choosing a file or folder

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

### Choosing a file or folder

このページでは、コネクタが書き込むtext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。

### Write documents to one single folder

ローカルコンピュータまたはネットワーク上のtext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。

### Write to a different file for each document

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、text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「区切りテキストコネクタ（書き込み）の取り込み用ルール [ 献 279]」を参照してください）。

## FTP server

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 connection parameters

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。

- **Server**  
FTPサーバ名を指定します。
- **Login**  
サイトへのアクセスを許可するログインを入力します。
- **Port**  
FTPサーバへのアクセスに使用するポートを指定します。
- **Password**  
ログイン用のパスワードを入力します。
- **Connect in passive mode**  
このオプションを選択して、パッシブモードで動作させます。

- **Connect via a proxy**  
このオプションは、HTTPおよびFTP接続の場合に利用できます。このオプションを使用して、特定の接続に対するプロキシサーバにアクセスします。このオプションを選択する場合、プロキシサーバの名前（またはIPアドレス）を入力し、必要であれば、**Use authentication**オプションを選択します。
- **Use authentication**  
このオプションが有効である場合、ユーザ名とパスワードを入力します。
- **Do not use a proxy server**  
HTTPおよびFTPタイプ接続の場合、プロキシサーバをバイパスする状況を指定できます（**List**フィールド）。使用する接続に応じて、HTTPまたはFTPタイプのアドレスを入力する必要があります。名前はスペースで区切る必要があります。

接続の有効性をテストするには、**Test**ボタンを使用します。

ウィザードの次のページでは、必要に応じて**Write the documents to a single file**または**Write to a different file for each document**のいずれかを選択します。

### Write a file

text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。

### Write to a folder

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、text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「区切りテキストコネクタ（書き込み）の取り込み用ルール [ 献 279]」を参照してください）。

## Behavior between two sessions

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。

### Write the documents to a single file

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。

- **Append to the same file**  
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。
- **Overwrite the last file**  
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。
- **Number the different files**

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。

### Write to a different file for each document

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。

- **Delete all previous files and start numbering them from the beginning**  
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。
- **Continue numbering files**  
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。

### Choose a description file

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 file** 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 file** 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 [献 272]」を参照してください。

#### 注意:

2つの区切りテキストコネクタを同一シナリオで使用する場合、もう一方のコネクタが上書きしないよう、最初のDSCファイルのバックアップコピーを作成してください。

### ブラインドモード

**Activate blind mode** 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。

- **Tab**
- **Comma**
- **Semi-colon**
- **Space**

- **Other**

**Other** 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**Delimited-text**コネクタの機能

ブラインドモードで**Delimited-text**コネクタを使用するには：

- **Activate blind mode** オプションを選択し、区切りテキストコネクタを設定します。
- データベースタイプのコネクタ（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。

- Scenario diagram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）。
- データベースタイプのコネクタを生成モードにします（**Tools**の**Produce now**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

**Write documents to one single folder**オプションが**Basic configuration**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。

**Write to a different file for each document**オプションが**Basic configuration**ウィザードで選択されている場合、

- text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）の書き込み

**Write DSC file** 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。**Enable blind mode**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。

### 使用エンコード

UNICODEバージョンを使用している場合、追加オプションで使用するエンコードタイプを選択できます。利用可能なエンコードタイプについては、互換対応表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。

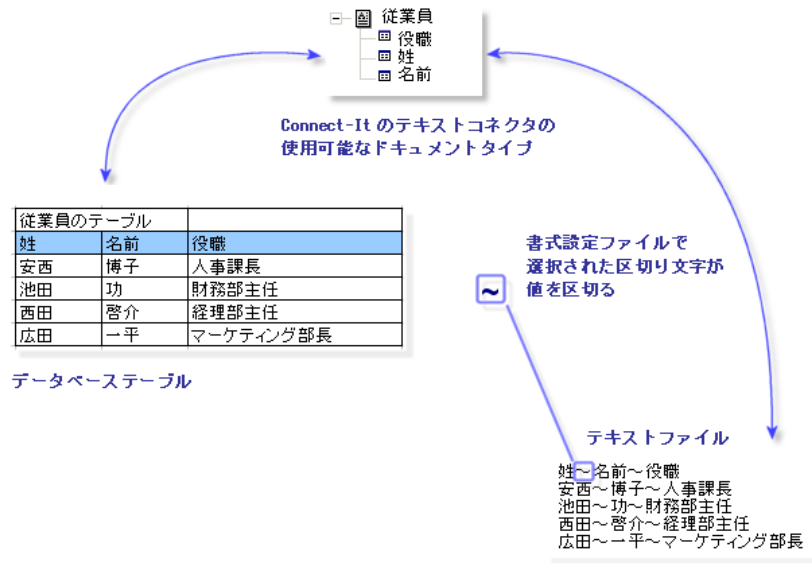
Textコネクタが発行するドキュメントタイプは、書式設定ファイル内で作成されたドキュメントタイプです。

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. 区切りテキストコネクタ - 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、**Configure the connector**ウィザードの**Choose a description file**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。

-  Select a document type [ 献 273]


---

-  Select a file for the preview [ 献 274]


---

-  Specify the column delimiters [ 献 274]

---

-  Specify the data-processing options [ 献 275]

---


-  Specify the column header and type [ 献 277]

---

### Select a document type

このページでは、テキストコネクタが発行する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。
- **Document type**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。
- **Next**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。
- **x** をクリックします。

### Select a file for the preview

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。

### File to preview

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）

### Number of lines to preview

ウィザードの次のページの **Data previewing** で表示する行の数を指定します。

### Specify the column delimiters

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。

#### ■ Fixed width

このオプションを選択すると、ウィザードの **Columns** ページの **Width** 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。

#### ■ Delimiter

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。

- **Tab**
- **Comma**
- **Semi-colon**
- **Space**
- **List of characters used as delimiters**

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。

### Specify the data-processing options

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
```

### Enter the column names

このオプションを選択して、生成ドキュメントの列ヘッダを入力します。

### Import column titles from the first line

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。

#### 注意:

列ヘッダとして使用される最初の行は、生成ドキュメントタイプでも処理されません。

	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\	116

#### 警告:

一度このオプションを有効にすると、変更できなくなります。

## Do not generate errors if a line contains a number of columns different than what is indicated in the description

このオプションを選択すると、区切りテキストの説明と一致しない行が無視されます。

行に含まれている列数が説明に示されている列数と異なる場合、その行は無視されエラーが生成されます。

## Quotation character

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\

**Keep quotes around values** 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。

## Commentary starting-line

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	

### Escape character

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

### Specify the column header and type

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。
  - **Name**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、**Type**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、**Text**、**Number**、**Date**、**Date and time**と**Currency**です。

- **Delimiters or fixed width**ページで**Fixed width**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、**Width**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。
- **X** 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。

## 発行ドキュメントタイプ

区切りテキストコネクタは、全ドキュメントタイプの以下の構造体を発行します。

- **UrlFileInfo**

この構造体の主な使用目的は処理後のアクションです。

**UrlFileInfo**構造体には以下のフィールドが含まれます。

- **creationdate**

このフィールドは、ドキュメントの作成日に対応します

- **lastmodificationdate**

このフィールドは、ドキュメントの最終更新日に対応します

- **line**

このフィールドは、ドキュメントの行数に対応します

- **path**

このフィールドは、ドキュメントのパスに対応します

## 区切りテキストコネクタ（書き込み）の取り込み用ルール

text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（ディレクティブ）を指定できます。

- A generic name

このフィールドのデフォルト値は、Textコネクタの取り込み用ドキュメントタイプの名前です。

- A file name extension

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。

Textコネクタの設定時に、**Write to a different file for each document**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、「区切りテキストコネクタ（書き込み）の設定 [ 献 265]」の節の「**Choosing a file or folder [ 献 268]**」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:

Text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. 区切りテキストコネクタ - 取り込み用ルール

Generic name
anAssetDst
Extension file name
txt

## 付属シナリオ - 区切り文字コネクタ

このコネクタ用の付属シナリオはありません。

---

## JMSコネクタ

オプションコネクタ

- 1 読み取り
  - 2 書き込み
- 

**Java Message Service**は、**Java**クライアントが非同期メッセージバスを利用できるようにするための**API**と必須セマンティクスを記述します。

以下の通信モードがこのコネクタでサポートされています。











- ポイントツーポイント（キュー）  
メッセージは単一のクライアントが取り込みます。
- 発行/加入（配布リスト）  
配布リストに加入しているすべてのクライアントがメッセージを取り込みます。

**JMS**コネクタにより、**XML**ドキュメントを**JMS**経由で交換できるようになります。

以下の動作の定義にコネクタを使用します。

- **JNDI**プロバイダとのインタラクション：接続アドレス、**ConnectionFactory**オブジェクト、キュー、トピックへのアクセスを可能にするルートクラス
- **JMS**サーバとの接続の定義
- プロキシサーバとのインタラクション

## JMSコネクタの設定

-  [Name and describe the connector \[ 献 16\]](#)
-  [Define a processing mode \[ 献 281\]](#)
-  [Configure the connection to the JNDI provider \[ 献 281\]](#)
-  [Configure the connection to the JMS server \[ 献 281\]](#)
-  [Define post-processing actions \[ 献 282\]](#)
-  [Specify the XML schema of the messages \[ 献 282\]](#)
-  [Configure the reconnection parameters \[ 献 282\]](#)
-  [Configuring the proxy server \[ 献 282\]](#)
-  [Configure the JVM \[ 献 282\]](#)
-  [Parallelize consumption \[ 献 283\]](#)

Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。



## Define a processing mode

このページで、コネクタの処理モードを指定できます（読み取りまたは書き込み）。

## Configure the connection to the JNDI provider

JNDI（Java Naming and Directory Interface）拡張は、クライアントがディレクトリにアクセスできるようにするJava標準です。

JMSサーバへの接続に関する情報はJNDIプロバイダに保存されます。このディレクトリにアクセスするには、JNDIプロバイダの情報が必要です。

- **Class** : JNDIプロバイダの初期コンテキストファクトリ実装クラス（`javax.naming.spi.InitialContextFactory`）の完全名を入力してください。JBoss JNDI実装の例を以下に挙げます。

```
org.jnp.interfaces.NamingContextFactory
```

このクラスがコネクタのクラスパスに存在する必要があります。JNDIプロバイダに対応するJNDIクライアントのJavaライブラリ（jar）をコネクタのクラスパスに追加する必要があります（▶ **Configure the JVM**ウィザードページ）。

- **URL** : JNDIディレクトリのURLを入力します。URLフォーマットはJNDIプロバイダによって異なります。たとえば、JBoss JNDIプロバイダの場合、URLは以下のフォーマットを持ちます。

```
jnp://myserver:1099
```

- 初期JNDIコンテキストを作成するための補足プロパティを**Name**および**Value**の各フィールドに入力できます。  
これらの補足プロパティの名前および解釈の方法は、JNDIプロバイダによって異なります。

## Configure the connection to the JMS server

JMSサーバへの接続は、JMS接続ファクトリ（`javax.jms.ConnectionFactory`）を使用して行ないます。接続ファクトリはJMS管理者が用意し、JNDIディレクトリで利用可能である必要があります。

JMSクライアントの実装クラスがコネクタのクラスパスに存在する必要があります。JMSプロバイダに対応するJMSクライアントのJavaライブラリ（jar）をコネクタのクラスパスに追加する必要があります（▶ **Configure the JVM**ウィザードページ）。

このページで、JMSサーバへの接続を設定します。

以下のフィールドを入力する必要があります。

- **ConnectionFactory** : JMS接続ファクトリを含むJNDIコンテキスト名です。
- **Destination** : XMLドキュメントの読み取りや書き込みを行うJMSディステーション（キューや配布リスト）を入力します。

## Define post-processing actions

JMSコネクタがファイルを読み取った後、処理が成功したか失敗したかに応じて、以下のオプションがあります。

- なにもしない
- メッセージを別のターゲットにコピー

JMSメッセージを取り込むとメッセージは廃棄されます。一度取り込むと、メッセージは配信されなくなります（また、ターゲットがキューである場合、メッセージは廃棄されます）。**Copy the message to another destination**オプションを選択していると、メッセージのコピーが選択したターゲットに送信されます。このターゲットの相関識別子（`JMSCorrelationId`）の値は、元のメッセージの識別子の値（`JMSMessageID`）に設定されます。

この機能は、処理レポートの使用に関係しています。

- ▶ 『Connect-Itユーザガイド』、「統合シナリオをプロダクションモードにする」、「統合シナリオのテストとデバッグ」

## Specify the XML schema of the messages

メッセージは.xmlフォーマットで転送されます。このページで、メッセージの処理に使用するxmlスキーマのURLを定義できます。

## Configure the reconnection parameters

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「**Configure the reconnection parameters**（詳細モード） [ 献 29]」の節を参照してください。

## Configuring the proxy server

このページで、使用する接続タイプを定義できます。

- **Direct connection**
- **HTTP**  
サーバの名前またはIPアドレス、および必要であればユーザ名とパスワードを入力します。
- **SOCKS**  
サーバの名前またはIPアドレス、および必要であればユーザ名とパスワードを入力します。

HTTPおよびSOCKSタイプ接続の場合、プロキシサーバをバイパスする状況を指定できます（**Do not use a proxy server for** フィールド）。使用する接続に応じて、HTTPまたはSOCKSタイプのアドレスを入力する必要があります。

## Configure the JVM

このページで、JVMが必要とするすべてのオプションとクラスパスを定義できます。

このページにクライアントで使用するすべてのオプションとクラスパスを入力します。

JNDIおよびJMSの各クライアントの実装クラスを**Classpath**枠で指定する必要があります。

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

### Parallelize consumption

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Parallelize consumption (詳細モード) [ 献 41]」の節を参照してください。

## 発行ドキュメントのタイプ

- 読み取りモードの場合：  
XMLスキーマと、JMSメッセージに関する情報を含むJMSMessageInfor構造体に対応するドキュメントです。  
ディレクティブ：
  - メッセージフィルタ：JMSメッセージのプロパティに応じて受信メッセージをフィルタするのに使用する、JMSフィルタ。
- 書き込みモードの場合：  
JMSMessageInfo取り込み時に生成されるドキュメントタイプには、送信されたメッセージに関する情報が含まれます。

## 取り込みドキュメントのタイプ

- 読み取りモードの場合：  
なし
- 書き込みモードの場合：  
XMLスキーマと、JMSメッセージに関する情報を指定するJMSMessageInfor構造体に対応するドキュメントです。
  - ディレクティブ：
    - Codepage：デフォルトでUTF-8
    - Priority：送信されるJMSメッセージの優先順位
    - Life cycle：送信されるJMSメッセージのライフサイクル (0=無制限)
    - Persistence：持続JMSメッセージを送信するかどうか

---

## Webサービスコネクタ

オプションコネクタ

- 自発生成（ソース）
  - 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サービスとインタラクションできます。Webサービスは、複数の操作からなる1つのサービスまたは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/POST、MIMEです。

SOAP（Simple Object Access Protocol）メッセージのプロトコルとは、HTTP、SMTPやその他のインターネットプロトコルなどの通信プロトコル（トランスポート層）の使用に基づく分散環境向けのデータ交換規格であり、XMLメッセージに基づいたデータ交換メカニズムを提供します。

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、サービスから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、操作が処理できるフォーマットのリストが含まれています。

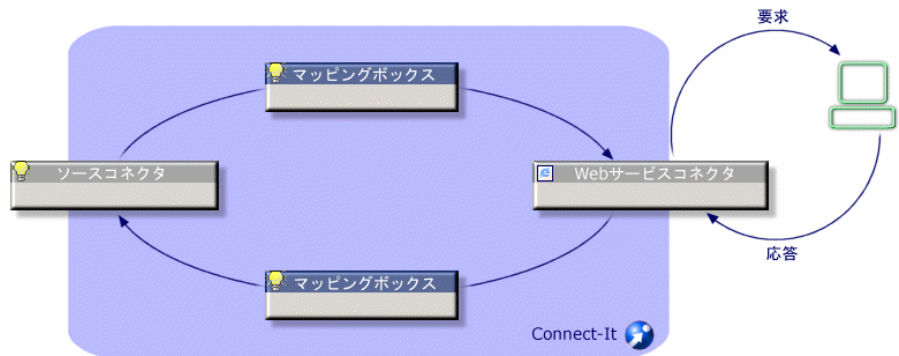
使用可能なWebサービスのリストは、<http://www.xmethods.com>などの他多数のサイトに記載されています。

統合シナリオでは、Webサービスコネクタはドキュメントを取り込み、要求としてWebサービスに送信します。その後、生成されたドキュメントタイプに自動的に変換された応答を受信します。ドキュメントを受信するにはこのコネクタはクエリをWebサービスに送信し、Webサービスから応答を受信するにはドキュメントを生成します。

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 5.6. Webサービスコネクタ - 一般的な原理



 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。

### 制約点 - Webサービスコネクタ

Webサービスコネクタのサポート対象外の操作を以下に挙げます。

- 同名の複数の操作  
Webサービスコネクタは、1つのWebサービス内で固有名の操作のみをサポートします。Webサービスが同一名の2つの操作を参照すると、第1の操作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。
- 複数の応答があるクエリ  
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ている場合、最初の操作のみが発行されます。
- 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、最初の操作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。

表 5.13. 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
HTTPプロトコルによるSOAP	可
HTTP GET	可
HTTP POST	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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 5.14. Webサービスコネクタの設定

 Name and describe the connector [ 献 16]
 Specify the WSDL [ 献 286]
 SSL configuration (詳細モード) [ 献 287]
 Parallelize consumption (詳細モード) [ 献 41]
 Configure the JVM [ 献 288]

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Specify the WSDL

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 address** フィールド：WSDLアドレスで、各Webサービスに関する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
  - サポートされる操作
  - これらの操作のフォーマット
  - 取り込まれたデータと生成されたデータ
- **Login** フィールド：ユーザ識別子
- **Password** フィールド

例：http://www.xmethods.net/sd/2001/TemperatureService.wsdl

## Specify the protocol

表示されているプロトコルのリストから、使用する転送プロトコルを選択します。

デフォルトで、すべてのプロトコルが使用され、操作はプロトコルに応じてプレフィックスが添付されます。

## SSL configuration (詳細モード)

このページでは以下の項目を定義できます。

- **セキュア接続用Java証明書ファイル (CA certificates file (truststore))**  
デフォルトでConnect-Itは、JREのJava証明書ファイルに関する情報を、`/lib/security/jsse.cacerts`ファイルおよび`/lib/security/cacerts`ファイルから検索します。
- **Truststore password**
- **Truststore type**
- **秘密キーファイル (Certificates and client keys file (keystore))** です。  
秘密キーがデータベースに格納されている場合、このファイルが必要になります。
- **秘密キーファイルのパスワード (Keystore password)** です。  
デフォルトでConnect-Itは、JREのJava証明書ファイルに関する情報を、`/lib/security/jsse.cacerts`ファイルおよび`/lib/security/cacerts`ファイルから検索します。

## Configuring the proxy server

このページで、使用する接続タイプを定義できます。

- **Direct connection**
- **HTTP**  
サーバの名前またはIPアドレス、および必要であればユーザ名とパスワードを入力します。
- **SOCKS**  
サーバの名前またはIPアドレス、および必要であればユーザ名とパスワードを入力します。

HTTPおよびSOCKSタイプ接続の場合、プロキシサーバをバイパスする状況を指定できます (**Do not use a proxy server for** フィールド)。使用する接続に応じて、HTTPまたはSOCKSタイプのアドレスを入力する必要があります。

## Parallelize consumption (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Parallelize consumption (詳細モード) [ 献 41]」の節を参照してください。

## Configure the JVM

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

## 取り込み用ルール - Webサービスコネクタ

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） [ 献 43]」章の「取り込み用ルール [ 献 46]」の節を参照してください。

### Operations to apply

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、操作`query`を選択します。

**Operation to apply** フィールドでなしを選択しても、`query`操作が使用されます。この場合、この操作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。

## 発行ドキュメントタイプ

Webサービスコネクタは、WSDLの定義内容に対応する生成ドキュメントタイプを発行します。

取り込み用に使用可能なドキュメントタイプは、Webサービスの操作に当たりません。

各操作ごとに、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。

### 発行ドキュメントタイプの名前



注意:

Webサービスコネクタは、サポート対象外プロトコルを使用する操作を発行しません（例：MIMEプロトコル）。

Webサービスコネクタが発行するドキュメントタイプに割り当てる名前には、通信プロトコルに対応するプレフィックスと操作の名前を含める必要があります。



## 取り込み用ドキュメントタイプ

表 5.15. Webサービスコネクタ - 取り込みドキュメントタイプ

プロトコル	取り込み用ドキュメントタイプ
SOAP Messaging	Soap-[操作名]
HTTP GET	HttpGet-[操作名]
HTTP POST	HttpPost-[操作名]

## 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を以下に挙げます。

- ws/wsac43/currency.scn [ 献 433]

---

## XML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)

---

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ファイル进行处理する。
- FTPサイトやWebサイトにあるXMLファイル进行处理する。

 注意:

処理されるXMLファイルは圧縮 (gzip) できません。

## 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

このコネクタに関する一部のオプションを有効にするには、**Edit/ Options/ Connector**メニューを使用します。詳細については、『ユーザガイド』、「シナリオビルダ」、「コネクタ」を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 5.16. XMLコネクタ（読み取り）の設定

Name and describe the connector [ 献 16]		
Choosing a processing mode [ 献 291]		
Selecting a connection protocol [ 献 291]		
HTTP Webサイトへの接続 [ 献 291]		
Define the connection parameters of the HTTP server [ 献 302]	FTP server connection [ 献 293]	Local or network file(s) [ 献 295]
Choose a DTD/XSD [ 献 295]	Select files or folders [ 献 25]	Select files or folders [ 献 295]
	Define post-processing actions [ 献 295]	Define post-processing actions [ 献 295]
	Choose a DTD/XSD [ 献 295]	Choose a DTD/XSD [ 献 295]
	Configure the JVM [ 献 300]	Configure the JVM [ 献 300]

### 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、設定ウィザードでファイルやフォルダのパスを指定しなくてはいけない場合があります。この場合、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」の節の「ドキュメント要素のパス」を参照してください。

#### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Choosing a processing mode

このページでは、コネクタを読み取りモードまたは書き込みモードのどちらで使用するのかを指定できます。

## Selecting a connection protocol

以下の3つのオプションがあります。

- HTTP Web site
- FTP server
- Local or network file(s)

## 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。

- Address  
[プロトコル]:/[アドレス]:[ポート]/[パス]のように、タイプの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。
- Login  
サイトへのアクセスを許可するログインを入力します。
- Password  
ログイン用のパスワードを入力します。

## Secured connection (HTTPS protocol)

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 [protocol] の部分がHTTPSプロトコルでない場合でも、**Secured connection (HTTPS protocol)** 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

Address	Secured connection (HTTPS protocol)	Connection mode
https://a.b.c.d/	選択	https port 443
https://a.b.c.d:9000/	選択	https port 9000

Address	Secured connection (HTTPS protocol)	Connection mode
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

### Client certificate

このフィールドで、コンピュータに既存のHTTPS証明書の中から1つを選択できます。

**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、**Scenario Builder**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

configuration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Root console/ Certificates - current user/ Personal/ Certificatesフォルダにあるリストと一致します。

Windows XPでこのフォルダに証明書を追加するには：

- 1 Windowsの**Start**メニューで**Run**を選択します。
- 2 **Open**フィールドに**MMC**と入力します。
- 3 **File/ Add/Remove Snap-in**を選択します。
- 4 表示されるウィンドウで**Add**をクリックします。
- 5 表示されるウィンドウで**Certificates**を選択します。
- 6 **Add**をクリックします。
- 7 表示されるダイアログボックスで**My user account**オプションを選択します。
- 8 **Finish**をクリックします。
- 9 **OK**をクリックします。
- 10 **Certificates - Current user/ Personal/ Certificates folder**フォルダ内でファイルを追加または削除します。

### Do not verify the server identity

セキュアな接続でのプロキシサーバをサポートするには、このオプションを使用します。

プロキシサーバがインストールされている場合、サーバ証明書に示されているアドレスは、プロキシサーバのアドレスとは異なります。このオプションを無効にして、サーバ識別の自動検証を無効にしてください。

### Connect via a proxy

このオプションは、HTTPおよびFTP接続の場合に利用できます。このオプションを使用して、特定の接続に対するプロキシサーバにアクセスします。

このオプションを選択する場合、プロキシサーバの名前（またはIPアドレス）を入力し、必要であれば、**Use authentication** オプションを選択します。

#### 注意:

すべてのプロキシに同一の認証情報を使用する場合、多段プロキシがサポートされません。

### Do not use a proxy server

HTTPタイプ接続の場合、プロキシサーバをバイパスする状況を指定できます（**Do not use a proxy server for** フィールド）。使用する接続に応じて、HTTPタイプのアドレスを入力する必要があります。名前はスペースで区切る必要があります。

## 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

#### 注意:

パッシブFTPモードでの接続がサポートされます。

### FTP server connection

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。

- **Server**  
FTPサーバ名を指定します。
- **Login**  
サイトへのアクセスを許可するログインを入力します。
- **Port**  
FTPサーバへのアクセスに使用するポートを指定します。
- **Password**  
ログイン用のパスワードを入力します。
- 接続がファイアウォールによってブロックされるために追加ポートを開く場合、**Connect in passive mode** オプションを選択します。
- **Connect via a proxy**

このオプションは、HTTPおよびFTP接続の場合に利用できます。このオプションを使用して、特定の接続に対するプロキシサーバにアクセスします。このオプションを選択する場合、プロキシサーバの名前（またはIPアドレス）を入力し、必要であれば、**Use authentication** オプションを選択します。プロキシサーバのアドレス（**Server address** フィールド）と、除外のリスト（**List** フィールド）を指定できます。



#### 注意:

すべてのプロキシに同一の認証情報を使用する場合、多段プロキシがサポートされます。

- **Use authentication**  
このオプションが有効である場合、ユーザ名とパスワードを入力します。
- **Do not use a proxy server**  
HTTPおよびFTPタイプ接続の場合、プロキシサーバをバイパスする状況を指定できます（**List** フィールド）。使用する接続に応じて、HTTPまたはFTPタイプのアドレスを入力する必要があります。名前はスペースで区切る必要があります。

接続の有効性をテストするには、**Test** ボタンを使用します。

#### Select files or folders

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select files or folders [ 献 25]」の節を参照してください。

#### Define post-processing actions

XMLコネクタがファイルを読み取ったあとの操作として、Connect-It には以下の3つのオプションがあります。

- **Leave it in the folder**
- **Delete it from the folder**
- **Move it to another folder**  
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。

処理後のアクションを考慮に入れるマッピングについては、本章の「処理後のアクションの使用 [ 献 295]」の節を参照してください。

## Local or network file(s)

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

## Select files or folders

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Select files or folders \[ 献 25\]](#)」の節を参照してください。

## Define post-processing actions

XMLコネクタがファイルを読み取ったあとの操作として、**Connect-It** には以下の3つのオプションがあります。

- Leave it in the folder
  - Delete it from the folder
  - Move it to the folder
- 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。

## 処理後のアクションの使用

処理後のアクションを使用するには：

- ソースコネクタが生成したドキュメントタイプの *URLFileInfo.Path* 構造体と *SuccessReport* ドキュメントタイプの *URLFileInfo.Path* 構造体との間にマッピングを作成します。


この構造体の詳細については、本章の「発行ドキュメントタイプ [ 献 314]」の節を参照してください。

## Choose a DTD/XSD

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、**File location**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。

 **警告:**

ドキュメントの部分的な検証の実行には、指定されたDTDを使用します。検証される要素には、DTDで宣言されていない要素、階層（親子）関係の整合性などの要素が検証されます。

## File location

以下の3つのオプションがあります。

### ■ Local/Network

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

**Server**、**Login**、**Password**と**Path**フィールドにFTPパラメータを入力します。

**Path**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、**Server**フィールドにftp.mycompany.comを入力し、**Path**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

**Server**、**Login**、**Password**と**Path**フィールドにHTTPパラメータを入力します。



**Path**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、**Server**フィールドに`http://mycompany.com`を入力し、**Path**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。

### ルート要素の決定

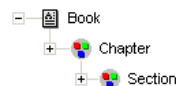
DTDまたはXSDは複数の要素から構成されており、これらの要素は更に別の要素を含んでいます（例：文書編集用のDTDでは、複数の**Section**要素は複数の**Chapter**要素に含まれ、更に**Chapter**要素は1つの要素**Book**に含まれます）。他の要素に含まれることがない要素は、ルート要素と呼ばれます。DTDまたはXSDは、1つまたは複数のルート要素から構成されることもあれば、全くルート要素が存在しないこともあります。

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、Connect-Itは以下の2つのオプションを提示します。

## Publish a document type for each root element found in the DTD/XSD（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。





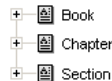
注意:

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。

## Publish a document type for each root element selected by the user (values are separated by commas)

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、ユーザが選択したルート要素ごとに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。

## Publish a document type for each first-level collection

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:

あるDTDのルート要素が*Company*で、第1レベルに3つのコレクション*Employees*、*Suppliers*と*Locations*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*Company*になります。

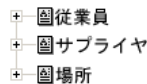
 **注意:**

第1レベルのコレクションとは、コネクタが発行するドキュメントタイプのルート要素の直下に現れるコレクションのことです。



たとえば、データベースにあるすべての従業員、サプライヤ、場所を含む単一の「会社」XML生成ドキュメントがある場合です。

**Publish a document type for each first-level collection** 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（*Company* 要素）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。



たとえば、データベースにあるすべての従業員、サプライヤ、場所を含むXML生成ドキュメントが複数ある場合です。

### Learning mode (no documents produced)

学習モードを使用すると、XMLコネクタは**Select a file or folder** 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：

- **Learning mode (no documents produced)** オプションを選択し、XMLコネクタを設定します。
- Scenario diagram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**Produce**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、**Choose a DTD** 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を使用すると、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。

- **Reject file and continue**

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**Action after processing** 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。

- **Issue warning and continue**

不整合が発生しても、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。

アプリケーションが処理する競合のタイプを以下に挙げます。

- 要素名
- 属性名
- 文字の出現を表す?および\*ワイルドカード文字

### Ignore the XML files whose root element does not correspond to the current document type (do not apply post-processing actions)

このオプションで、ドキュメントタイプ別、つまりリンク先DTD別にXMLドキュメントを処理できます。

このオプションを選択すると、現在のDTDのみを考慮し、その結果として他のドキュメントタイプにリンクされているXMLドキュメントの処理エラーを回避できます。

例えば、複数のXMLドキュメントがあるフォルダを指定し、このオプションを選択すると、**DTD/XSD** フィールドで定義したDTDのみを考慮できます。

### Configure the JVM

このページが表示されるのは、設定ウィザードの**Choose a DTD/XSD** [ 献 87 ] ページでXSDを選択している場合のみです。

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

## XMLコネクタに関連するオプション

XMLコネクタのオプションにアクセスするには、**Edit/ Options/ Connector/ Delimited text and XML**メニューを使用します。

以下のオプションがFTPプロトコルでのXMLコネクタに関連します。

- Display the URL being processed in the Connect-It log
- Copy locally the files to read from FTP server.

このオプションを選択して、FTPサーバにあるファイルのローカルコピーを作成し、ローカルファイルからデータを読み込みます。

ネットワーク設定では、ファイルの処理に十分な長さの接続をFTPサーバで維持できない場合、このオプションを選択する必要があります。

## 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 5.17. XMLコネクタ（書き込み）の設定

☒ Name and describe the connector [ 献 16]		
☒ Select a processing mode [ 献 302]		
☒ Selecting a connection protocol [ 献 302]		
☒ Define the connection parameters of the HTTP server [ 献 302]	☒ FTP接続 [ 献 304]	☒ Select files or folders [ 献 303]
☒ Choose a DTD/XSD [ 献 307]	☒ Select files or folders [ 献 303]	☒ Behavior between two sessions [ 献 306]
	☒ Behavior between two sessions [ 献 306]	☒ Choose a DTD/XSD [ 献 307]
	☒ Configure the JVM [ 献 310]	☒ Configure the JVM [ 献 310]



### 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、設定ウィザードでファイルやフォルダのパスを指定しなくてはいけない場合があります。この場合、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」の節の「ドキュメント要素のパス」を参照してください。

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Select a processing mode

このページでは、コネクタを読み取りモードまたは書き込みモードのどちらで使用するのかを指定できます。

### Selecting a connection protocol

以下のオプションがあります。

- FTP
- Local or network file(s)
- HTTP

### Define the connection parameters of the HTTP server

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。

- Address  
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、タイプアドレスを入力します。
- Login  
サイトへのアクセスを許可するログインを入力します。
- Password  
ログイン用のパスワードを入力します。

### Secured connection (HTTPS protocol)

このオプションの使用方法については、「Secured connection (HTTPS protocol) [ 献 291]」の節を参照してください。

## 書き込みコマンド

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へデータを送信します。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。

## Local or network file(s)

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。

## Select files or folders

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。

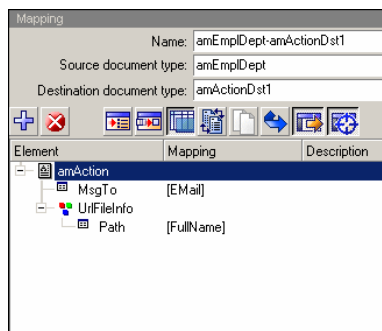
コネクタの動作を定義する前に、ファイルの書き込み先となるフォルダ (**Folder name**フィールド) を指定します。このフォルダのパスは、*UrlFileInfo.Path*要素と組み合わせてマッピングで使用されます。

以下の動作から選択できます。

- **Write documents to one single file**  
ローカルコンピュータまたはネットワーク上のフォルダのパスを指定します (**Folder name**フィールド)。ドキュメントが上書きされるかどうかは、ウィザードの次のページで定義するパラメータによります。
- **Write to a different file for each document**  
ファイルの書き込み先フォルダ名を指定します (**Folder name**フィールド)。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モード [ 献 310]」を参照してください。

*UrlFileInfo.Path*要素をマッピングすることで、これらのファイルの名前を定義することもできます。例えば、AssetCenterの各従業員の名前を持つファイ

ル名を作成するなどです。この場合、ウィザードページで定義するパスは、マッピングで定義するパスと連結します。



マッピングで定義したパスをデフォルトパスとして使用し、ウィザードで定義したパスを使用しないためには、'/'文字をマッピングで使用します。

#### 注意:

この動作により、コネクタ設定ウィザードで有効なパスを指定できなくなります。

*UrlFileInfo.Path*要素の指定されたパスが存在しない場合、以下の条件の下でパスが自動的に作成されます。

- パスは、設定ウィザードで指定されているパスの一部を使用します。
- パスは、指定されたツリーに2レベル以上は追加しません。設定ウィザードでa/b/cというパスが指定されている場合、a/b/c/d/filenameというパスのみ、マッピングで指定できます。

## 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接続

このページでは、FTPサーバに接続するために以下のフィールドに入力します。

- **Server**  
FTPサーバ名を指定します。
- **Login**  
サイトへのアクセスを許可するログインを入力します。  
ログインには大文字小文字の区別があります。



- **Port**  
FTPサーバへのアクセスに使用するポートを指定します。
- **Password**  
ログイン用のパスワードを入力します。
- **Connect in passive mode**  
このオプションを選択して、パッシブモードで動作させます。
- **Connect via a proxy**  
このオプションは、HTTPおよびFTP接続の場合に利用できます。このオプションを使用して、特定の接続に対するプロキシサーバにアクセスします。このオプションを選択する場合、プロキシサーバの名前（またはIPアドレス）を入力し、必要であれば、**Use authentication**オプションを選択します。
- **Use authentication**  
このオプションが有効である場合、ユーザ名とパスワードを入力します。
- **Do not use a proxy server**  
HTTPおよびFTPタイプ接続の場合、プロキシサーバをバイパスする状況を指定できます（**List**フィールド）。使用する接続に応じて、HTTPまたはFTPタイプのアドレスを入力する必要があります。名前はスペースで区切る必要があります。

接続の有効性をテストするには、**Test**ボタンを使用します。

### Select files or folders

コネクタの動作を定義する前に、以下の項目を指定します。

- 書き込みファイルフォルダ（**Path**フィールド）。指定したパスは、*UrlFileInfo.Path*要素と組み合わせられ、マッピングで使用されます。

必要に応じて**Write the documents to a single folder**または**Write to a different file for each document**オプションを選択します。

### Write documents to one single folder

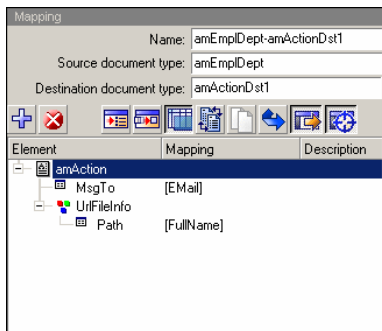
XMLコネクタで作成するファイルのパスと名前を入力します（**Path**フィールド）。このファイルが既に存在する場合、その内容は上書きされます。

### Write to a different file for each document

ファイルの書き込み先となるフォルダを指定します（**Path**フィールド）。これらのファイルの名前は、XMLコネクタの取り込みルールで入力した一般名に対応します。本節の「XMLコネクタ（書き込み）の取り込み用ルール - FTPモード [ 献 310]」を参照してください。

*UrlFileInfo.Path*要素をマッピングすることで、これらのファイルの名前を定義することもできます。例えば、AssetCenterの各従業員の名前を持つファイル名

を作成するなどです。この場合、ウィザードページで定義するパスは、マッピングで定義するパスと連結します。



マッピングで定義したパスをデフォルトパスとして使用し、ウィザードで定義したパスを使用しないためには、'/'文字をマッピングで使用します。

#### 注意:

この動作により、コネクタ設定ウィザードで有効なパスを指定できなくなります。

*UrlFileInfo.Path*要素の指定されたパスが存在しない場合、以下の条件の下でパスが自動的に作成されます。

- パスは、設定ウィザードで指定されているパスの一部を使用します。
- パスは、指定されたツリーに2レベル以上は追加しません。設定ウィザードで *a/b/c* というパスが指定されている場合、*a/b/c/d/filename* というパスのみ、マッピングで指定できます。

### Behavior between two sessions

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。

#### Write the documents to a single file

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。

- **Append to the same file**  
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。
- **Overwrite the last file**  
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。

- **Number the different files**

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。


### Write to a different file for each document

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。

- **Delete all previous files and start numbering them from the beginning**  
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。
- **Continue numbering files**  
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。

### Choose a DTD/XSD

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コネクタが発行する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、**File location**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。

### File location

以下の3つのオプションがあります。

- **Local/Network**

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**

**Server**、**Login**、**Password**と**Path**フィールドにFTPパラメータを入力します。

**Path**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、**Server**フィールドに**ftp.mycompany.com**を入力し、**Path**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

**Server**、**Login**、**Password**と**Path**フィールドにHTTPパラメータを入力します。

**Path**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、**Server**フィールドに「ftp.mycompany.com」を入力し、**Path**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。

## ルート要素の決定

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要素は複数のChapter要素に含まれ、更にChapter要素は1つの要素Bookに含まれます）。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。

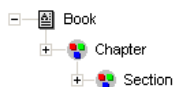
ルート要素（つまり、XMLコネクタに発行される使用可能なドキュメントタイプに当たります）を指定するために、Connect-Itni\は以下の2つのオプションがあります。

- Publish a document type for each root element found in the DTD/XSD（推奨オプション）
- Publish a document type for each root element selected by the user (values are separated by commas)

## Publish a document type for each root element found in the DTD/XSD（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。



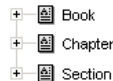
 **注意:**

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。

## Publish a document type for each root element selected by the user (values are separated by commas)

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、ユーザが選択したルート要素ごとに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。

### Do not reference the resulting DTD in the XML file

このオプションはデフォルトで選択されています。

### Write resulting DTD/XSD inside the XML file

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。

### Write resulting DTD to an external file and reference it in the XML file

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開きます。各ソースコネクタに対応する1つのセッションは、スケジューラがシナリオで開始します（例：毎週ドキュメントを生成するソースコネクタ）。

DTDまたはXSDは、シナリオの第1回目のセッション時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。

## Configure the JVM

このページが表示されるのは、設定ウィザードのChoose a DTD/XSD [ 献 87] ページでXSDを選択している場合のみです。

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」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モード

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（ディレクティブ）を指定できます。

- A generic name

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。

- A file name extension

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、**Write to a different file for each document**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） [ 献 301]」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：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 5.7. XMLコネクタ - の取り込み用ルール

Generic name
amAssetDst
Extension file name
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。

#### Write the documents to a single file

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、単一XMLファイルがAssetCenterに記録されている資産のインベントリ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 **Publish a document type for each first-level collection** オプションを選択してはなりません。

### Write to a different file for each document

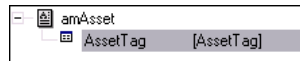
このモードでは、XMLコネクタが取り込む各ドキュメントを別々のファイル内に保存できます。

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になります。

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 **Publish a document type for each first-level collection** オプションを選択してはなりません。

### end-of-lineマーカーの変換ルール

XMLの使用モードが読み取りモードか書き込みモードであるかに応じて、end-of-lineマーカー（CRLFまたはLF）のルールは異なります。

### 読み取りモードでの使用

デフォルト

- Windowsでは、すべてのend-of-line文字がCRLF（復帰と改行）に変換されます。
- UNIXでは、すべてのend-of-line文字がLF（改行）に変換されます。

この動作は以下の例にあるように、スクリプトを使用して変更できます。

```
RetVal = Replace([Record.cc.dd.ee], Chr(13)&Chr(10),Chr(10),0)
```

## 書き込みモードでの使用

デフォルト

- Windowsでは、すべてのend-of-line文字がCRLF（復帰と改行）に変換されます。
- UNIXでは、すべてのend-of-line文字がLF（改行）に変換されます。  
この動作は、スクリプトの使用で変更することはできません。

## 発行ドキュメントタイプ

XMLコネクタは、全ドキュメントタイプの以下の構造体を発行します。

- **UrlFileInfo**

この構造体の主な使用目的は処理後のアクションです。

**UrlFileInfo**構造体には以下のフィールドが含まれます。

- **creationdate**

このフィールドは、ドキュメントの作成日に対応します

- **lastmodificationdate**

このフィールドは、ドキュメントの最終更新日に対応します

- **path**

このフィールドは、ファイルのパスに対応します



**注意:**

複数のXMLファイルタイプを使用する場合、XMLファイルと同数のコネクタを展開する必要があります。各コネクタには一意の動作（作成または取り込み）、動作と同数のコネクタを作成する必要があります。

## 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。

---

### Altirisコネクタ

オプションコネクタ

■ 生成 (ソース)

本節ではAltirisコネクタについて説明します。

Altirisコネクタで、Altiris Inventory Solutionデータベースを処理できます。

Altirisデータベースには、インベントリされた資産（デスクトップおよびポータブルコンピュータ、サーバなど）およびその従属資産（プリンタ）に関連するデータが含まれます。

#### 前提条件

- **LastUse**タイプデータをConnect-Itで処理できるようにするため、Altirisデータベースにビューを作成します。このビューはAltirisコネクタで必要になります。  
以下のスクリプトを使用してビューを作成します。

```
CREATE VIEW [vCIT_LastUseInformation]
AS
SELECT softinv_id, MAX(softstats.[Last Start]) AS [Last Use]
FROM vResource r INNER JOIN
Inv_AeX_AM_Monthly_Summary softstats ON r.Guid = softstats._ResourceGuid INNER JOIN
AeXInv_AeX_SW_Audit_Software softinv ON r.ResourceId = softinv.Wrks
taId AND softstats.[Internal Name] = softinv.InternalName AND
softstats.[Product Version] = softinv.ProductVersion AND softstats.[File Name] = softinv.[File Name]
GROUP BY softinv_id
```

- 処理パフォーマンスを改善するため、以下のインデックスを作成することを推奨します。

- **Inv\_AeX\_SW\_Audit\_Software**テーブル（テーブルが存在する場合）：

```
CREATE UNIQUE
INDEX [idxCIT_Inv_AeX_SW_Audit_Software__id] ON [Inv_AeX_SW_Audit_Software] ([_id])
ON [PRIMARY]
```

- **Inv\_AeX\_SW\_Audit\_Software\_spt**テーブル（テーブルが存在する場合）：

```
CREATE UNIQUE
INDEX [idxCIT_Inv_AeX_SW_Audit_Software_spt__id] ON [Inv_AeX_SW_Audit_Software_spt] ([_id])
ON [PRIMARY]
```

- **ItemResource**テーブル：

```
CREATE UNIQUE
INDEX [idxCIT_ItemResource_ResourceId] ON [ItemResource] ([ResourceId])
ON [PRIMARY]
```

## Altirisコネクタの設定

ウィザードを使用してコネクタを設定できます。「コネクタの設定 [ 献 15]」の章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 6.1. Altirisコネクタの設定

 Name and describe the connector [ 献 16]

Select a connection type [ 献 18]
Configuring the connection [ 献 19]
Advanced configuration (詳細モード) [ 献 27]
Configure the reconnection parameters (詳細モード) [ 献 29]
Determine server delay (詳細モード) [ 献 31]
Configure schedule pointers (詳細モード) [ 献 33]
Configure the cache (詳細モード) [ 献 36]
Advanced options (詳細モード) [ 献 37]
Define document types (詳細モード) [ 献 39]

## Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

## Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

## Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

- **Execute an initial import** : このオプションで、データベースが空である場合や、挿入対象の項目がまだ存在しない場合などに、初期インポートの照合更新スクリプトでSELECTクエリを使用しないで済ませることができます。これにより処理時間が短縮されます。

## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

## Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

## Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## 追加情報

### 一意な照合更新キー

- *vComputer*ビューの*Guid* フィールドが、*AssetCenter*テーブルへのマッピングの一意な識別子として使用されます。

### スケジュールポイント

*AeXInv\_AeX\_AC\_Inventory\_Results*ビューの*Collection Time*フィールドは、スケジュールポイントとして使用できます。

インベントリテーブルに入力されているコンピュータのみが考慮されます。インベントリテーブルに入力されていないテーブルは、個別にスケジュールする必要があります。

### 付属シナリオ - Altirisコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- *Altiris*シナリオ [ 献 374]

---

## Asset Insightコネクタ

### オプションコネクタ

- 生成 (ソース)

*Asset Insight 4.0*ゲートウェイコネクタで、*Asset Insight*データベースを処理できます。

*Asset Insight*データベースには、会社のIT備品に関する情報が含まれます。シナリオを使用することで、*Asset Insight*データベースに登録されている各コンピュータに関するレコードを*AssetCenter*に作成できます。

## Asset Insightの既知の制約点

Asset Insightコネクタを、同一バージョンレベルのDLLファイルを使用するODBCドライバとODBC Administratorと併せて使用することを強く推奨します。

例：

Asset Insightコネクタは、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場合、動作しません。

### ポインタのステータスの必須の初期化

Asset Insightコネクタに関連するシナリオでは、ポインタがテキストフィールドであれば、スケジューラの編集ウィンドウでポインタに'0'の値を割り当てる必要があります。日付フィールドであれば、機能するのに十分に古い日付を割り当てる必要があります。デフォルトでは、シナリオを最初にスケジュールモードで起動するより前に、Asset Insightコネクタは値をポインタに割り当てることができません。



**警告:**

この手順を終えないと、Asset Insightコネクタはドキュメントを生成しません。

この値を割り当てるには：

- 1 **Edit/Scheduling**を選択します。
- 2 Asset Insightコネクタが生成したドキュメントを、以前に作成したスケジューラに関連付けます。
- 3 リストに現れる各生成ドキュメントについて、Pointerステータス列の対応するセルをダブルクリックして、テキストボックスを表示します。
- 4 表示されるテキスト編集用枠に'0'を入力します。

## Asset Insight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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 6.2. Asset Insightコネクタの設定

 Name and describe the connector [ 献 16]

 Select a connection type [ 献 18]

 Configuring the connection [ 献 19]

 Advanced configuration (詳細モード) [ 献 27]

 Configure the reconnection parameters (詳細モード) [ 献 29]

 [Determine server delay \(詳細モード\) \[ 献 31\]](#)

 [Configure schedule pointers \(詳細モード\) \[ 献 33\]](#)

 [Configure the cache \(詳細モード\) \[ 献 36\]](#)

 [Advanced options \(詳細モード\) \[ 献 37\]](#)

 [Define document types \(詳細モード\) \[ 献 39\]](#)

---

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

### Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

### Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。



## Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## Asset Insight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) [ 献 43]」章の「生成用ルール [ 献 44]」の節を参照してください。

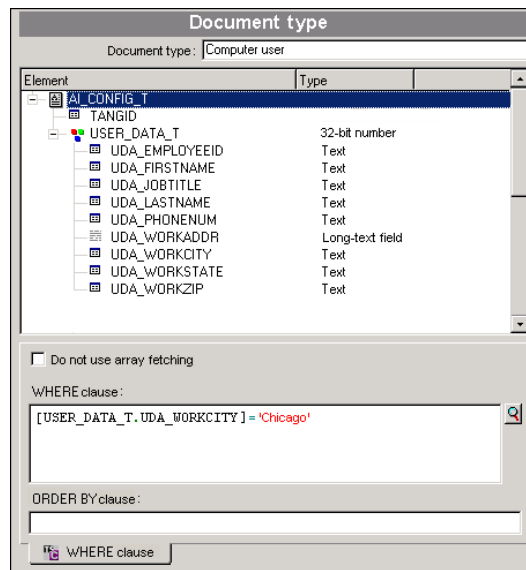
このコネクタに対し、これらのルールは**WHERE**句や**ORDERBY**句で作成されており、ソースデータベースのレコードをフィルタを適用できます。これらの句の詳細については「コネクタのルール (ディレクティブ) [ 献 43]」章の「生成用ルール [ 献 44]」節、「**WHERE**句と**ORDER BY**句 [ 献 45]」を参照してください。

## Asset InsightコネクタのWHERE句の例

シカゴで働く全従業員を取得するには、**WHERE**句フィールドに以下の関数を入力する必要があります。

```
[USER_DATA_T.UDA_WORKCITY]='Chicago'
```

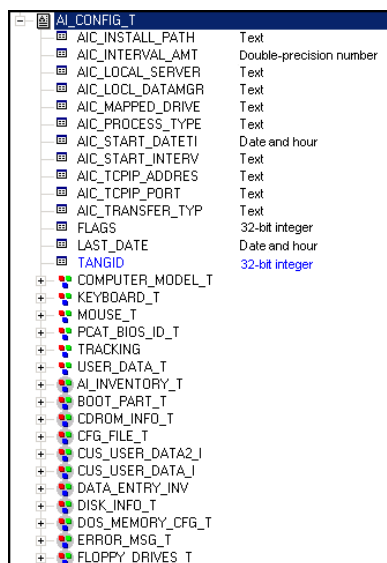
図 6.1. Asset Insightコネクタ - 生成ルールの例



## 発行ドキュメントタイプ

Asset Insightコネクタは、**AI\_CONFIG\_T**という1ドキュメントタイプのみを発行します。このドキュメントタイプは、コンピュータに対して実行可能なAsset Insightスキャンに対応します。このドキュメントの要素はスキャン設定によって異なります。一例を挙げると、データベースのテーブルを表すコレクションが現れるスキャンと現れないスキャンがあります。

図 6.2. Asset Insightコネクタ - AI\_CONFIG\_Tドキュメントタイプ



Field Name	Data Type
AIC_INSTALL_PATH	Text
AIC_INTERVAL_AMT	Double-precision number
AIC_LOCAL_SERVER	Text
AIC_LOCL_DATAMGR	Text
AIC_MAPPED_DRIVE	Text
AIC_PROCESS_TYPE	Text
AIC_START_DATETI	Date and hour
AIC_START_INTERV	Text
AIC_TCPIP_ADDRES	Text
AIC_TCPIP_PORT	Text
AIC_TRANSFER_TYP	Text
FLAGS	32-bit integer
LAST_DATE	Date and hour
TANGID	32-bit integer
COMPUTER_MODEL_T	
KEYBOARD_T	
MOUSE_T	
PCAT_BIOS_ID_T	
TRACKING	
USER_DATA_T	
AI_INVENTORY_T	
BOOT_PART_T	
CDROM_INFO_T	
CFG_FILE_T	
CUS_USER_DATA2_I	
CUS_USER_DATA_I	
DATA_ENTRY_INV	
DISK_INFO_T	
DOS_MEMORY_CFG_T	
ERROR_MSG_T	
FLOPPY_DRIVES_T	

### 警告:

このドキュメントの要素はAsset Insightデータベース構造によって異なります。AI\_CONFIG\_Tドキュメントタイプ内のコレクションによって表される一部のテーブルはデータベースに存在しないことさえがあります。シナリオマッピングがこのような存在しないテーブルにあるフィールドを参照しようとすると、このシナリオは正しく動作しなくなります。この問題を解決するには、以下の場所にある参照フィールドを削除する必要があります。

- 1 シナリオのマッピング内
- 2 Asset Insightコネクタの生成用ドキュメントタイプ内

## 付属シナリオ - Asset Insightコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- Asset Insightシナリオ [ 献 375]

---

## LANDesk for Inventory 8コネクタ

オプションコネクタ

- 生成 (ソース)

LANDeskインベントリコネクタでは、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。

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。

### 既知の制約点

LANDeskコネクタを、同一バージョンレベルのDLLファイルを使用するODBCドライバとODBC Administratorと併せて使用することを強く推奨します。

例：

LANDeskコネクタは、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場合、動作しません。

### 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

**表 6.3. LANDesk for Inventory 8コネクタの設定**

 Name and describe the connector [ 献 16]
 Select a connection type [ 献 18]
 Configuring the connection [ 献 19]
 Advanced configuration (詳細モード) [ 献 27]
 Configure the reconnection parameters (詳細モード) [ 献 29]
 Determine server delay (詳細モード) [ 献 31]
 Configure schedule pointers (詳細モード) [ 献 33]
 Configure the cache (詳細モード) [ 献 36]
 Advanced options (詳細モード) [ 献 37]
 Define document types (詳細モード) [ 献 39]

## 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

## Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

## Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

## Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

## Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

## Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

## Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

## Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

## Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## LANDeskコネクタの生成用ルール

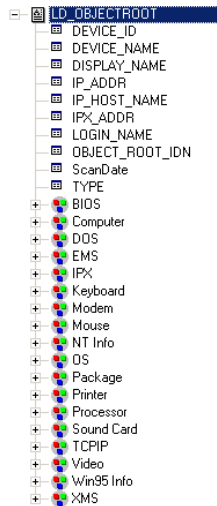
コネクタの生成用ルールの入力方法については、「コネクタのルール (ディレクティブ) [ 献 43]」章の「生成用ルール [ 献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。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) [ 献 43]」章の「生成用ルール [ 献 44]」節、「**WHERE**句と**ORDER BY**句 [ 献 45]」を参照してください。

## 発行ドキュメントタイプ

LANDeskコネクタが発行する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、LANDeskがコンピュータに実行するスキャンに相当します。

### 図 6.3. LANDesk for Inventory 8コネクタ - COMPUTER\_SYSTEMドキュメントタイプ



## 付属シナリオ - LANDesk for Inventory 8コネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- LANDeskシナリオ [ 献 386]

---

## LANDesk Software Distributionコネクタ

オプションコネクタ

- 1 取り込み (ターゲット)
- 2 取り込み - 自発生成 (要求 - 応答)

---

LANDesk Software Distributionコネクタで、LANDesk Webサービスとインタラクションできます。

統合シナリオでは、LANDesk Software Distributionコネクタはドキュメントを取り込み、クエリとしてWebサービスに送信します。その後、生成されたドキュメントに自動的に変換された応答を受信します。

- ▶ 『AssetCenter Software Distribution Guide』

### LANDesk Software Distributionの制約点

NTLM認証プロトコルのみが使用されます。

LANDesk Management Suiteバージョン8.6以降を使用し、MBSDKがLANDeskサーバにインストールされている必要があります。

### 前提条件

- 管理権限を持つLANDeskアカウントが存在する必要があります。LANDesk Software Distributionコネクタを設定し、AssetCenterシナリオに保存する際、このアカウントが入力されている必要があります。  
アカウントがLANDesk管理者アカウントとして使用され、タスクはLANDeskに含まれる情報を更新するため、AssetCenterでスケジュールされたタスクを作成する権限が与えられた人員に対し、これらのタスクへのアクセスを制限する必要があります。
- LANDesk Software Distributionシナリオは、Enterprise DiscoveryからAssetCenterシナリオで取得されたインベントリデータを使用します。
- Enterprise DiscoveryおよびLANDeskの各エージェントがインストールされている必要があります。

## LANDesk Software Distributionコネクタの設定

表 6.4. LANDesk Software Distributionコネクタの設定

☒ Name and describe the connector [ 献 16]

☒ Specify the WSDL [ 献 327]

☒ SSL configuration (詳細モード) [ 献 327]

☒ Configuring the proxy server [ 献 328]

☒ Configure the JVM [ 献 328]

☒ Parallelize consumption [ 献 328]

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Specify the WSDL

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 address** フィールド：WSDLアドレスで、各Webサービスに関する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
  - サポートされる操作
  - これらの操作のフォーマット
  - 取り込まれたデータと生成されたデータ
- **Domain\login** フィールド：ユーザID
- **Password** フィールド

### SSL configuration (詳細モード)

このページでは以下の項目を定義できます。

- セキュア接続用Java証明書ファイル (**CA certificates file (truststore)**)  
デフォルトでConnect-Itは、JREのJava証明書ファイルに関する情報を、/lib/security/jsse.cacertsファイルおよび/lib/security/cacertsファイルから検索します。
- **Truststore password**
- **Truststore type**
- 秘密キーファイル (**Certificates and client keys file (keystore)**)。秘密キーがデータベースに格納されている場合、このファイルが必要になります。
- 秘密キーファイルのパスワード (**Keystore password**)

## Configuring the proxy server

このページで、使用する接続タイプを定義できます。

- Direct connection
- HTTP

サーバの名前またはIPアドレス、および必要であればユーザ名とパスワードを入力します。

- SOCKS

サーバの名前またはIPアドレス、および必要であればユーザ名とパスワードを入力します。

HTTPおよびSOCKSタイプ接続の場合、プロキシサーバをバイパスする状況を指定できます (**Do not use a proxy server for** フィールド)。使用する接続に応じて、HTTPまたはSOCKSタイプのアドレスを入力する必要があります。

## Configure the JVM

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

## Parallelize consumption

このページの使用方法については、「コネクタの設定 [ 献 15]」章を参照してください。

## 付属シナリオ

このコネクタを使用する付属シナリオを以下に挙げます。

- LANDeskシナリオ [ 献 386]

---

## System Management System (SMS) コネクタ

オプションコネクタ

- 生成 (ソース)

SMS 2.xとSMS 2003 (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。



## コネクタ使用上の重要な情報

### gateway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を本番に移行する前、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前もって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。

- マッピングの照合更新キーの有効性を確認する  
無効な照合更新キーを使用すると、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。

## 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 6.5. SMS 2.xコネクタとSMS 2003コネクタの設定

 Name and describe the connector [ 献 16]

 Select a connection type [ 献 18]

 Configuring the connection [ 献 19]

<a href="#">Advanced configuration (詳細モード) [ 献 27]</a>
<a href="#">Configure the reconnection parameters (詳細モード) [ 献 29]</a>
<a href="#">Determine server delay (詳細モード) [ 献 31]</a>
<a href="#">Configure schedule pointers (詳細モード) [ 献 33]</a>
<a href="#">Configure the cache (詳細モード) [ 献 36]</a>
<a href="#">Advanced options (詳細モード) [ 献 37]</a>
<a href="#">Define document types (詳細モード) [ 献 39]</a>

## 前提条件

コネクタの設定前に、SMSデータベースへの有効な接続が、ODBC Administrator 内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

## Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

## Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

## Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

## Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

## Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Configure the cache \(詳細モード\) \[ 献 36\]](#)」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Advanced options \(詳細モード\) \[ 献 37\]](#)」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「[Define document types \(詳細モード\) \[ 献 39\]](#)」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) [ 献 43]」章の「[生成用ルール \[ 献 44\]](#)」の節を参照してください。

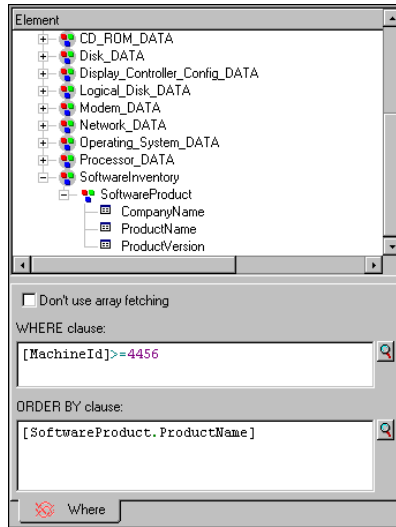
このコネクタに対し、これらのルールは**WHERE**句や**ORDERBY**句で作成されており、ソースデータベースのレコードをフィルタを適用できます。これらの句の詳細については「コネクタのルール (ディレクティブ) [ 献 43]」章の「[生成用ルール \[ 献 44\]](#)」節、「[WHERE句とORDER BY句 \[ 献 45\]](#)」を参照してください。

### Asset Insightコネクタの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 2.xとSMS 2003コネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- SMSシナリオ [ 献 414]

---

## iInventoryコネクタ

オプションコネクタ

- 生成 (ソース)

iInventoryコネクタで、ODBCデータソースからのデータを処理できます。



**警告:**

ODBC接続を使用するには、ODBC Administratorが前もってインストールされている必要があります。

## iInventoryの既知の制約点

iInventoryコネクタを、同一バージョンレベルのDLLファイルを使用するODBCドライバとODBC Administratorと併せて使用することを強く推奨します。

---

例：

iInventoryコネクタは、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場合、動作しません。

---

### メッセージの種類

iInventoryコネクタが発行する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、logフィールド、Blobフィールドなど）があります。

iInventoryコネクタがサポートするフィールドタイプは、Microsoft Accessフィールドタイプです。

## iInventory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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 6.6. iInventoryコネクタの設定

<a href="#">Name and describe the connector [ 献 16]</a>
<a href="#">Select a connection type [ 献 18]</a>
<a href="#">Configuring the connection [ 献 19]</a>
<a href="#">Advanced configuration (詳細モード) [ 献 27]</a>
<a href="#">Configure the reconnection parameters (詳細モード) [ 献 29]</a>
<a href="#">Determine server delay (詳細モード) [ 献 31]</a>
<a href="#">Configure schedule pointers (詳細モード) [ 献 33]</a>
<a href="#">Manage transactions (詳細モード) [ 献 35]</a>
<a href="#">Configure the cache (詳細モード) [ 献 36]</a>
<a href="#">Advanced options (詳細モード) [ 献 37]</a>
<a href="#">Define document types (詳細モード) [ 献 39]</a>

### 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

### Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

### Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Manage transactions (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Manage transactions (詳細モード) [ 献 35]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## iInventory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) [ 献 43]」章の「生成用ルール [ 献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。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) [ 献 43]」章の「生成用ルール [ 献 44]」節、「WHERE句とORDER BY句 [ 献 45]」を参照してください。

### 注意:

WHERE句とORDERBY句は、生成されたドキュメントタイプのルートレベルとコレクションのレベルに適用されます。

AQLシンタックスの詳細については、付録「AQLクエリ [ 献 435]」を参照してください。

## sysdateフィールド

AQLで記述するWHERE句では、**sysdate**フィールドを**getdate()**関数に置換する必要があります。

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。

## 追加情報

本節では以下の内容について説明します。

- ODBCデータソースの宣言に関する詳細

### ODBCデータソースの宣言に関する詳細

iInventoryコネクタを含むシナリオを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

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 Administratorを起動します。
- 2 コネクタが使用するODBCデータソースが**System DNS**タブに表示されていることを確認します。

System 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 - iInventoryコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- iInventoryシナリオ [ 献 385]

---

## 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はコンピュータスキャンで情報を収集します。この情報を元に、専用の付属シナリオを使用してターゲットデータベースにレコードを作成できません。

### 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:

コネクタは、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場合、動作しません。

---

### 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。


以下の表に、コネクタの設定ウィザードの全ページを一覧します。

**表 6.7. Tivoli Inventory Management 4.0コネクタの設定**

---

 Name and describe the connector [ 献 16]

 Select a connection type [ 献 18]

 Configuring the connection [ 献 19]

---



 [Advanced configuration \(詳細モード\) \[ 献 27\]](#)

 [Configure the reconnection parameters \(詳細モード\) \[ 献 29\]](#)

 [Determine server delay \(詳細モード\) \[ 献 31\]](#)

 [Configure schedule pointers \(詳細モード\) \[ 献 33\]](#)

 [Configure the cache \(詳細モード\) \[ 献 36\]](#)

 [Advanced options \(詳細モード\) \[ 献 37\]](#)

 [Define document types \(詳細モード\) \[ 献 39\]](#)

## 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

## Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

## Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

## Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

## Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

## Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」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) [ 献 43]」章の「生成用ルール [ 献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。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) [ 献 43]」章の「生成用ルール [ 献 44]」節、「WHERE句とORDER BY句 [ 献 45]」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) [ 献 43]」章の「取り込み用ルール [ 献 46]」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、**Reconciliation**と**Advanced reconciliation**タブ内で照合更新用のパラメータを入力します。

これら照合更新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) [ 献 43]」章の「取り込み用ルール [ 献 46]」節内「照合更新 [ 献 47]」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を以下に挙げます。

- Tivoliシナリオ [ 献 429]

---

## Tivoli CM for Software Distribution 4.2コネクタ

このコネクタは、Tivoli Region Managerデータ、特にSD\_PACKAGEテーブルに含まれるデータを読み込みます。

このコネクタは、Connect-Itと併せて提供されている『TCM統合』マニュアルに独立して記載されています。

---

## 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アプリケーションで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。この情報に基づき、付属シナリオがAssetCenterやServiceCenterデータベースにデータをマイグレートします。

### 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：

Tivoli CM - Inventory 4.2コネクタは、ODBC Administratorバージョン3.0とMicrosoft Access ODBCドライバ4.00を使用している場合、動作しません。

---

### 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。

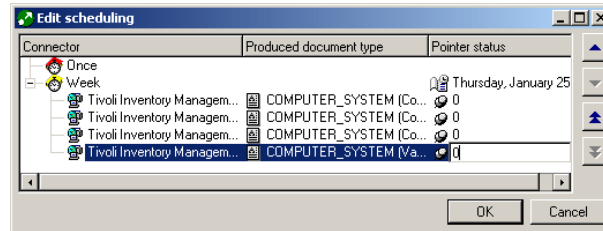


**警告:**

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 **Scenario/ Scheduling** 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、**Pointer status**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 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 6.8. Tivoli Inventory Management 4.2コネクタの設定

☐ Name and describe the connector [ 献 16]
☐ Select a connection type [ 献 18]
☐ Configuring the connection [ 献 19]
☐ Advanced configuration (詳細モード) [ 献 27]
☐ Configure the reconnection parameters (詳細モード) [ 献 29]
☐ Determine server delay (詳細モード) [ 献 31]
☐ Configure schedule pointers (詳細モード) [ 献 33]
☐ Configure the cache (詳細モード) [ 献 36]
☐ Advanced options (詳細モード) [ 献 37]
☐ Define document types (詳細モード) [ 献 39]

### 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

Connect-Itのコネクタの設定は、ウィザードで実行できます。

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

### Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

### Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

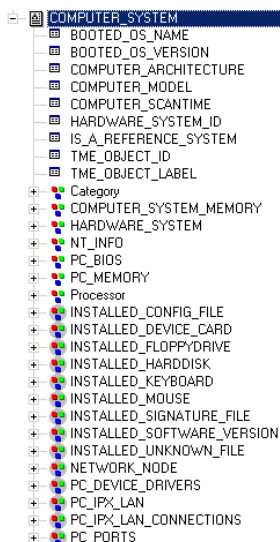
### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## 発行ドキュメントタイプ

Tivoli CM Inventory 4.2コネクタが発行する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

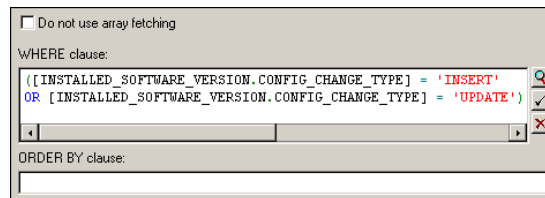
## Tivoli CM Inventory 4.2コネクタの生成用ルール

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 the Tivoli CM Inventory 4.2 connector. At the top, there is a checkbox labeled "Do not use array fetching" which is currently unchecked. Below this, there is a section for the "WHERE clause" containing a text area with the following SQL query: `([INSTALLED_SOFTWARE_VERSION.CONFIG_CHANGE_TYPE] = 'INSERT' OR [INSTALLED_SOFTWARE_VERSION.CONFIG_CHANGE_TYPE] = 'UPDATE')`. To the right of the text area are several icons, including a search icon and a close icon. Below the WHERE clause section is an empty text area for the "ORDER BY clause".

#### 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。

#### Do not use array fetching

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を以下に挙げます。

- Tivoliシナリオ [ 献 429]

---

## Tivoli CM for Software Distribution Status 4.2コネクタ

このコネクタにより、DIST\_STATEテーブルからデータを読み込むことができます。

---

### 注意:

このコネクタにはスケジュールポイントはサポートされていません。

このコネクタは、Connect-Itと併せて提供されている『TCM統合』マニュアルに独立して記載されています。

---

## TS.Census 2コネクタ

オプションコネクタ

■ 生成 (ソース)

---

TS.Census 2ゲートウェイコネクタを使用することで、Tally System TS.Census 2から取得したデータの処理が可能になります。このアプリケーションは、Tally System TS.Census 2ソフトウェアスイートにあります。

Tally System TS.Census 2はITポートフォリオに関する情報を収集します。この情報を使用することで付属シナリオで、ターゲットデータベースにレコードを作成できます。

### TS.Census 2の既知の制約点

TS.Census 2コネクタを、同一バージョンレベルのDLLファイルを使用するODBCドライバとODBC Administratorと併せて使用することを強く推奨します。

---

例:

TS.Census 2コネクタは、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場合、動作しません。

---

### TS.Census 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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。



表 6.9. TS.Census 2コネクタの設定

<a href="#">Name and describe the connector [ 献 16]</a>
<a href="#">Select a connection type [ 献 18]</a>
<a href="#">Configuring the connection [ 献 19]</a>
<a href="#">Advanced configuration (詳細モード) [ 献 27]</a>
<a href="#">Configure the reconnection parameters (詳細モード) [ 献 29]</a>
<a href="#">Determine server delay (詳細モード) [ 献 31]</a>
<a href="#">Configure schedule pointers (詳細モード) [ 献 33]</a>
<a href="#">Configure the cache (詳細モード) [ 献 36]</a>
<a href="#">Advanced options (詳細モード) [ 献 37]</a>
<a href="#">Define document types (詳細モード) [ 献 39]</a>

### 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

### Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

### Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

### Define document types (詳細モード)


このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## 発行ドキュメントタイプ

TS.Census 2コネクタは、**NC\_Component**と**NC\_Workstation**の2タイプのドキュメントを発行します。

**NC\_Component**ドキュメントタイプは、TS.Census 2が実行するコンピュータ-コンポーネントスキャン (ハードウェア、ソフトウェアの両方) に対応します。

**NC\_Workstation**ドキュメントタイプは、スキャンが収集したコンピュータ関連情報に対応します。

Element	Type
+  NC_Component	
+  NC_Workstation	

#### 注意:

ドキュメントタイプの詳細については、Tally System TS.Census 2のマニュアルを参照してください。これらのテーブルとフィールドすべてが付録で説明されています。

## TS.Census 2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) [ 献 43]」章の「生成用ルール [ 献 44]」の節を参照してください。

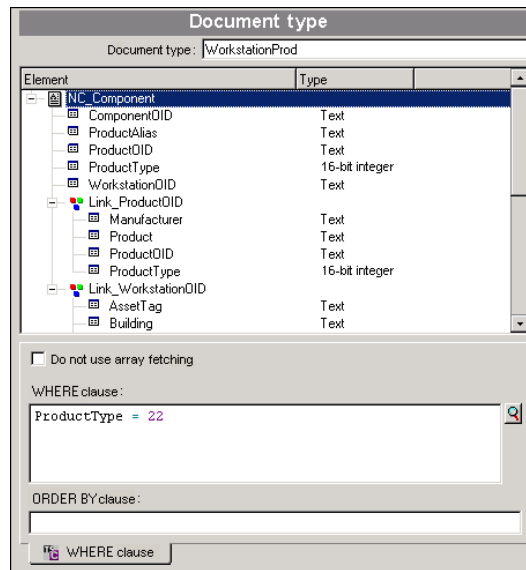
このコネクタに対し、これらのルールは**WHERE**句や**ORDERBY**句で作成されており、ソースデータベースのレコードをフィルタを適用できます。これらの句の詳細については「コネクタのルール（ディレクティブ） [ 献 43]」章の「生成用ルール [ 献 44]」節、「**WHERE**句と**ORDER BY**句 [ 献 45]」を参照してください。

### TS.Census 2コネクタの生成用ルールの例

22に等しい製品タイプを持つNC\_Componentテーブルからレコードを取得するには、以下の**WHERE**句を記述します。

```
[ProductType]=22
```

## 図 6.7. TS.Census 2コネクタの生成用ルール



### 付属シナリオ - TS.Census 2コネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- TS Censusシナリオ [ 献 427]

---

## TS.Census 3コネクタ

オプションコネクタ

■ 生成 (ソース)

TS.Census 3ゲートウェイコネクタを使用することで、Tally System TS.Census 3から取得したデータの処理が可能になります。このアプリケーションは、Tally System TS.Census 3ソフトウェアスイートにあります。

Tally System TS.Census 3はITポートフォリオに関する情報を収集します。この情報を使用することで付属シナリオで、ターゲットデータベースにレコードを作成できます。

### TS.Census 3の既知の制約点

TS.Census 3コネクタを、同一バージョンレベルのDLLファイルを使用するODBCドライバとODBC Administratorと併せて使用することを強く推奨します。

---

例：

TS.Census 3コネクタは、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場合、動作しません。

---

### TS.Census 3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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 6.10. TS.Census 3コネクタの設定

■ Name and describe the connector [ 献 16]
■ Select a connection type [ 献 18]
■ Configuring the connection [ 献 19]
■ Advanced configuration (詳細モード) [ 献 27]
■ Configure the reconnection parameters (詳細モード) [ 献 29]
■ Determine server delay (詳細モード) [ 献 31]
■ Configure schedule pointers (詳細モード) [ 献 33]
■ Configure the cache (詳細モード) [ 献 36]
■ Advanced options (詳細モード) [ 献 37]
■ Define document types (詳細モード) [ 献 39]

## 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

## Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

## Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

## Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

## Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

## Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

## Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

## Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

## Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## 発行ドキュメントタイプ

TS.Census 3コネクタは、**NC\_Component**と**NC\_Workstation**の2タイプのドキュメントを発行します。

**NC\_Component**ドキュメントタイプは、TS.Census 3が実行するコンピュータ-コンポーネントスキャン (ハードウェア、ソフトウェアの両方) に対応します。

**NC\_Workstation**ドキュメントタイプは、スキャンが収集したコンピュータ関連情報に対応します。

Element	Type
+ NC_Component	
+ NC_Workstation	

### 注意:

ドキュメントタイプの詳細については、Tally System TS.Census 3のマニュアルを参照してください。これらのテーブルとフィールドすべてが付録で説明されています。

## TS.Census 3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) [ 献 43]」章の「生成用ルール [ 献 44]」の節を参照してください。

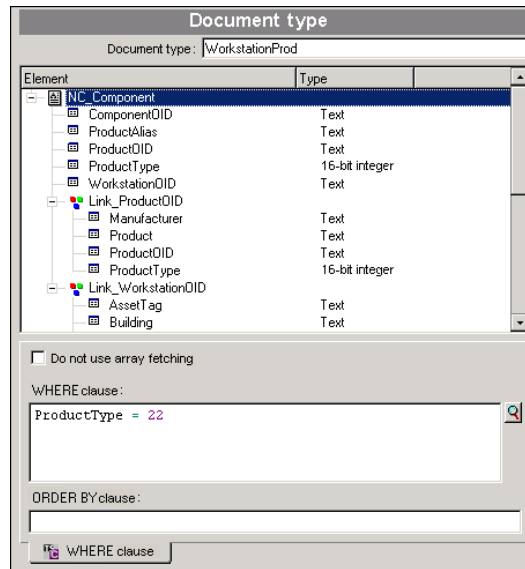
このコネクタに対し、これらのルールは**WHERE**句や**ORDERBY**句で作成されており、ソースデータベースのレコードをフィルタを適用できます。これらの句の詳細については「コネクタのルール (ディレクティブ) [ 献 43]」章の「生成用ルール [ 献 44]」節、「**WHERE**句と**ORDER BY**句 [ 献 45]」を参照してください。

### TS.Census 3コネクタの生成用ルールの例

22に等しい製品タイプを持つ**NC\_Component**テーブルからレコードを取得するには、以下の**WHERE**句を記述します。

```
[ProductType]=22
```

## 図 6.8. TS.Census 3コネクタの生成用ルール



### 付属シナリオ - TS.Census 3コネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- TS Censusシナリオ [ 献 427]

## CA Unicenter AMO 3コネクタ

オプションコネクタ

- 生成 (ソース)

CA Unicenter AMOゲートウェイコネクタでは、Unicenter AMOネットワークインベントリツールで取得し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。CA Unicenter AMO - Asset Management付属シナリオ (amoac.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。

## CA Unicenter AMO 3コネクタの既知の制約点

CA Unicenter AMOコネクタを、同一バージョンレベルのDLLファイルを使用するODBCドライバとODBC Administratorと併せて使用することを強く推奨します。

---

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。

---

## CA Unicenter AMO 3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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 6.11. Unicenter AMO 3コネクタの設定

■ Name and describe the connector [ 献 16]
■ Select a connection type [ 献 18]
■ Configuring the connection [ 献 19]
■ Advanced configuration (詳細モード) [ 献 27]
■ Configure the reconnection parameters (詳細モード) [ 献 29]
■ Determine server delay (詳細モード) [ 献 31]
■ Configure schedule pointers (詳細モード) [ 献 33]
■ Configure the cache (詳細モード) [ 献 36]
■ Advanced options (詳細モード) [ 献 37]
■ Define document types (詳細モード) [ 献 39]

### 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。



### Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「**Configuring the connection** [ 献 19]」の節を参照してください。

### Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「**Advanced configuration (詳細モード)** [ 献 27]」の節を参照してください。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「**Configure the reconnection parameters (詳細モード)** [ 献 29]」の節を参照してください。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「**Determine server delay (詳細モード)** [ 献 31]」の節を参照してください。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「**Configure schedule pointers (詳細モード)** [ 献 33]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「**Configure the cache (詳細モード)** [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「**Advanced options (詳細モード)** [ 献 37]」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「**Define document types (詳細モード)** [ 献 39]」の節を参照してください。

## CA 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 (ディレクティブ) [ 献 43]」章の「**生成用ルール** [ 献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。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) [ 献 43]」章の「**生成用ルール** [ 献 44]」節、「**WHERE**句と**ORDER BY**句 [ 献 45]」を参照してください。

## 発行ドキュメントタイプ

CA Unicenter AMOコネクタが発行するドキュメントタイプを以下に挙げます。

- Computer
- Motor
- Group
- User

## 付属シナリオ - CA Unicenter AMOコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- Unicenter AMOシナリオ [ 献 432]

---

## CA Unicenter AMO 4コネクタ

オプションコネクタ

- 生成 (ソース)

---

CA Unicenter AMOゲートウェイコネクタでは、Unicenter AMOネットワークインベントリツールで取得し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。CA Unicenter AMO - Asset Management付属シナリオ (amoac.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せん。

## CA Unicenter AMO 4コネクタの既知の制約点

CA Unicenter AMOコネクタを、同一バージョンレベルのDLLファイルを使用するODBCドライバとODBC Administratorと併せて使用することを強く推奨しません。

---

例:

AC Unicenter AMOコネクタは、ODBC Administratorバージョン3.0とMicrosoft Access ODBCドライバ4.00を使用している場合、動作しません。

---

## CA Unicenter AMO 4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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 6.12. CA Unicenter AMO 4コネクタの設定

<a href="#">Name and describe the connector [ 献 16]</a>
<a href="#">Select a connection type [ 献 18]</a>
<a href="#">Configuring the connection [ 献 19]</a>
<a href="#">Advanced configuration (詳細モード) [ 献 27]</a>
<a href="#">Configure the reconnection parameters (詳細モード) [ 献 29]</a>
<a href="#">Determine server delay (詳細モード) [ 献 31]</a>
<a href="#">Configure schedule pointers (詳細モード) [ 献 33]</a>
<a href="#">Configure the cache (詳細モード) [ 献 36]</a>
<a href="#">Advanced options (詳細モード) [ 献 37]</a>
<a href="#">Define document types (詳細モード) [ 献 39]</a>

## 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

## Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

## Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

## Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

## Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

## Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

## Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## CA Unicenter AMO 4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) [ 献 43]」章の「生成用ルール [ 献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。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) [ 献 43]」章の「生成用ルール [ 献 44]」節、「WHERE句とORDER BY句 [ 献 45]」を参照してください。

## 発行ドキュメントタイプ

CA Unicenter AMO 4コネクタが発行するドキュメントタイプを以下に挙げます。

- Application
- Computer
- Motor
- Group
- User
- DirectAccessTables

このドキュメントタイプには、コネクタが発行したその他すべてのドキュメントタイプのコレクションが含まれます。このコレクションを、ターゲットコネクタが取り込むドキュメントタイプに直接的にマップできます。

コレクションからドキュメントへのマッピングの詳細については、『ユーザガイド』、「*Mapping document types (ドキュメントタイプのマッピング)*」

の章にある、「*Mapping source elements to destination elements* (ソース要素のターゲット要素へのマッピング)」 / 「コレクションからドキュメントへのマッピング」の節を参照してください。

## 付属シナリオ - CA Unicenter AMO 4コネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- Unicenter AMOシナリオ [ 献 432]

---

## Winpark Actimaコネクタ

オプションコネクタ

- 生成 (ソース)

Winpark Actimaコネクタで、Winpark Actimaソフトウェアスイート付属のネットワークインベントリツールを使用することで取得し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。Winpark Actima - Asset Management付属シナリオ (wpkam.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。

### 既知の制約点

Winpark Actimaコネクタを、同一バージョンレベルのDLLファイルを使用するODBCドライバとODBC Administratorと併せて使用することを強く推奨します。

---

例:

Winpark Actimaコネクタは、ODBC Administratorバージョン3.0とMicrosoft Access ODBCドライバ4.00を使用している場合、動作しません。

---


Paradox 5.0による接続が失われた場合、接続を解除してから再接続してください。これで通常、問題が解決されます。

### Winpark Actima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 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

## 表 6.13. Winpark Actimaコネクタの設定

 Name and describe the connector [ 献 16]
 Select a connection type [ 献 18]
 Configuring the connection [ 献 19]
 Advanced configuration (詳細モード) [ 献 27]
 Configure the reconnection parameters (詳細モード) [ 献 29]
 Determine server delay (詳細モード) [ 献 31]
 Configure schedule pointers (詳細モード) [ 献 33]
 Configure the cache (詳細モード) [ 献 36]
 Advanced options (詳細モード) [ 献 37]
 Define document types (詳細モード) [ 献 39]

### 前提条件

コネクタの設定前に、ODBCデータベースへの有効な接続が、ODBC Administrator内で宣言する必要があります。

有効なODBC宣言には、適切なログインとパスワードを使用する必要があります。

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Select a connection type

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Select a connection type [ 献 18]」の節を参照してください。

### Configuring the connection

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configuring the connection [ 献 19]」の節を参照してください。

### Advanced configuration (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced configuration (詳細モード) [ 献 27]」の節を参照してください。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

### Determine server delay (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Determine server delay (詳細モード) [ 献 31]」の節を参照してください。

### Configure schedule pointers (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure schedule pointers (詳細モード) [ 献 33]」の節を参照してください。

### Configure the cache (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the cache (詳細モード) [ 献 36]」の節を参照してください。

### Advanced options (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Advanced options (詳細モード) [ 献 37]」の節を参照してください。

### Define document types (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Define document types (詳細モード) [ 献 39]」の節を参照してください。

## Winpark Actima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) [ 献 43]」章の「生成用ルール [ 献 44]」の節を参照してください。

このコネクタに対し、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をフィルタを適用できます。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) [ 献 43]」章の「生成用ルール [ 献 44]」節、「**WHERE**句と**ORDER BY**句 [ 献 45]」を参照してください。

## Winpark Actimaコネクタが発行するドキュメントタイプ

Winpark Actimaは、以下のドキュメントタイプを発行します。

- 購入
- 予算
- 契約
- 依頼
- DirectAccessTables
- エンティティ
- 請求書
- サプライヤ
- 作業指示
- ソフトウェア
- 周辺機器

- ポスト
- ネットワーク
- サイト
- ユーザ

## 付属シナリオ - Winpark Actimaコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- Winpark Actimaシナリオ [ 献 433]



## 7 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** [ 献 363]  
このコネクタは、SAP R/3サーバが公開するBAPIメソッドに対応するドキュメントタイプを発行します。このコネクタで、このデータの同期処理が可能です。
- **SAP IDocコネクタ** [ 献 366]  
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ドキュメントタイプを発行します。このIDOCファイルで、SAP R/3サーバとのデータの送受信が可能になります。
- **SAP ALEコネクタ** [ 献 369]  
このコネクタはALE通信層を使用して、IDOCにあるフラットテキストファイルに対応するドキュメントタイプを発行します。

## 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, .soまたは.o) を使用する必要があります。これらのファイルは以下の手順で入手できます。

- 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入手する必要があります。

- 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 2.1.2**をクリックします。

このダウンロードで、SAPコネクタが機能するのに必要なファイルを含む圧縮ファイル (.zipまたは.tgz) を入手できます。



#### 注意:

SAP JCoの他のバージョンがサポートされます。

<http://h20229.www2.hp.com/index.html>にある、Connect-Itの本バージョンの互換対応表をご確認ください。

- 6 圧縮されているファイルをディスク上に展開します。

- 7 `sapjco.jar`アーカイブをConnect-Itインストールフォルダの**lib**サブフォルダに展開します。
- 8 ネイティブのJavaライブラリをインストールします。この手順は使用するオペレーティングシステムによって異なります。

以下のライブラリをConnect-Itインストールフォルダの**bin32**または**bin**サブフォルダにコピーします。

- `librfc32.dll`および`sapjcorfp.dll` (Microsoft Windowsオペレーティングシステムの場合)
- `librfc32.so`および`libsapjcorfc.so` (LinuxおよびSun Solarisオペレーティングシステムの場合)
- `librfc32.o`および`libsapjcorfc.o` (IBM AIXオペレーティングシステムの場合)

---

## 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 - 自発生 (要求 - 応答)

---

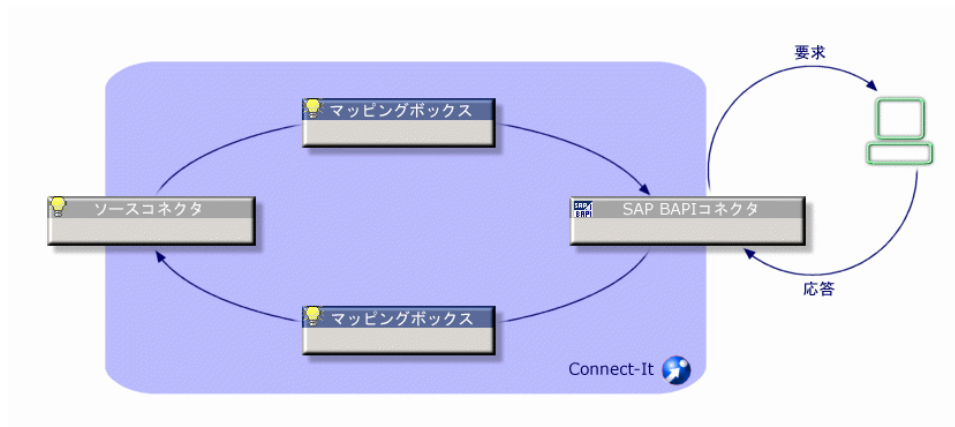
SAP BAPIコネクタはSAP R/3サーバに接続し、ビジネスオブジェクトによってソートされたサービス (BAPI) のリストを取得します。

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サーバヘ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 7.1. 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。

#### 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 7.1. ABAPデータ型とConnect-Itデータ型の対応関係

ABAP type	JCOデータ型	Connect-Itデータ型
C	TYPE_CHAR	テキスト
D	TYPE_DATE	日付
P	TYPE_BCD	倍精度整数 (BigDecimal)
T	TYPE_TIME	日付
X	TYPE_BYTE	Blob
N	TYPE_NUM	整数 (BigInteger)
F	TYPE_FLOAT	倍精度整数 (BigDecimal)
I	TYPE_INT	整数
b	TYPE_INT1	整数
s	TYPE_INT2	整数
g	TYPE_STRING	テキスト
y	TYPE_XSTRING	Blob

## 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。  
以下の表に、コネクタの設定ウィザードの全ページを一覧します。

表 7.2. SAP BAPIコネクタの設定

 Name and describe the connector [ 献 16]
 Configure server connection [ 献 365]
 Configure the JVM [ 献 366]
 Parallelize consumption (詳細モード) [ 献 41]

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### Configure server connection

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 server

SAPサーバ名またはサーバにアクセスするためのルート文字列を指定します。以下に例を挙げます。

```
/H/saprouter1/H/saprouter2/H/sapserver
```

#### System number

SAPサーバと共に使用するシステム番号を指定します。

#### Client

クライアントポートを入力します。

例：800

#### Login

ユーザ名を指定します。

#### Password

前のフィールドで指定したユーザ名に対応するパスワードを入力します。

#### Language

サーバからの情報を表示する際の言語を指定します。

例：英語には`en`、ドイツ語には`de`など。

### Enable Auto Commit/RollBack mode after each document

このオプションは、BAPIユーザコール後に変更を検証するため、**BAPI\_TRANSACTION\_COMMIT**（または**BAPI\_TRANSACTION\_ROLLBACK**）BAPIコールを自動的に有効にします。  
このオプションは、書き込みモードのみで使用できます。

### Configure the JVM

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

### Parallelize consumption（詳細モード）

このページの使用方法については、「コネクタの設定 [ 献 15]」章を参照してください。

## 付属シナリオ - SAP BAPIコネクタ

このコネクタを使用する付属シナリオを以下に挙げます。

- SAP BAPIシナリオ [ 献 412]

---

## 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。

 **警告:**

複数ドキュメントを含むIDocファイルの場合、各ドキュメントは、ヘッダ (**EDI\_DC** セグメント) で区切る必要があります。

## 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。

取り込み用ドキュメントタイプは、IDocファイルヘッダ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。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

 **注意:**

複数のIDocファイルタイプを使用する場合、IDocファイルと同数のコネクタを展開する必要があります。各コネクタには一意の動作（作成または取り込み）、動作と同数のコネクタを作成する必要があります。

表 7.3. SAP IDocコネクタの設定

 Name and describe the connector [ 献 16]		
 Choosing a processing mode [ 献 291]		
 Selecting a connection protocol [ 献 291]		
 Local or network file(s) [ 献 295]	 HTTP Webサイトへの接続 [ 献 291]	 FTP server connection [ 献 293]
 Define post-processing actions [ 献 295]	 Define post-processing actions [ 献 294]	
 IDoc type definition (read and write) [ 献 367]		
 Configure the JVM [ 献 369]		

### Name and describe the connector

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Name and describe the connector [ 献 16]」の節を参照してください。

### IDoc type definition (read and write)

このページではIDocドキュメントタイプとその取得方法を選択できます。

- ローカルのIDocタイプ

**Local file** オプションで、コンピュータに既に存在するIDocタイプを使用できます。

- SAPサーバのIDocタイプ

**On the SAP server** をクリックすると、SAPサーバのIDocタイプを動的に取得できます。

利用できるフィールドは、選択したオプションによって変化します。

### IDOCタイプファイル

**Local IDoc types** オプションを選択した場合は、**Browse** 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 version (write)

使用されるSAPのバージョンを指定します。

### SAPサーバ接続設定

以下のフィールドで、以下のSAPサーバ接続パラメータを指定できます。

- SAP server
- User
- Password
- Language
- Client
- System identifier

### IDoc設定

#### Basic type

生成するSAPドキュメント名です。

*MATMAS01* などです。

#### 拡張

このフィールドで以下の項目を入力できます。

- IDocタイプのCIMタイプ  
バージョン2または3
- SAPサーバCIMタイプ  
使用可能なバージョン : 3.1 Gから4.6 C

#### セグメントリリース

セグメントが依存するSAPバージョンです。最新バージョンのアプリケーションが考慮され、最も古いバージョンのセグメントに適応されます。



## IDocレコードタイプのバージョン

バージョン2のレコードタイプはSAP 3.xに対応します。

バージョン3のレコードタイプはSAP 4.xに対応します。

### Configure the JVM

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。

---

## SAP ALEコネクタ

SAP ALEコネクタは、IDocファイルをSAP IDocコネクタのように処理します。しかし、SAP ALEコネクタはALE (Application Link Enabling) 通信レイヤを使用して、これらのファイルをSAP R/3サーバと送受信します。SAP ALEが受信するイベント (生成ドキュメントタイプに対応) もSAP R/3サーバのRFCを使用できます。

SAP IDocとは異なり、同期データ処理と非同期データ処理の両方がサポートされます (イベントドリブンモード)。

### 前提条件

以下の要素をSAPサーバ用に定義する必要があります。

- 1 論理システム
- 2 RFCターゲット
- 3 tRFCポート
- 4 配布モデル
- 5 ポートプロファイル

### SAP IDoc ALEコネクタの設定

ウィザードでコネクタを設定できます。

以下の表に、コネクタの設定ウィザードの全ページを一覧します。

---

 Nommer et décrire le connecteur [ 献 370]

---

 Select a processing mode [ 献 370]

---

 Configure SAP server configuration [ 献 370]

---

 Configure the reconnection parameters (詳細モード) [ 献 371]

---

## Nommer et décrire le connecteur

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「コネクタの設定 [ 献 15]」の節を参照してください。

## Select a processing mode

このページで、コネクタの動作が読み込みモードであるか書き込みモードであるかを定義できます。

## Configure SAP server configuration

このページで、SAPサーバの接続パラメータを定義します。

### SAP server

SAPサーバの名前です

### System number

SAPサーバと共に使用するシステム番号を指定します。

### Client

クライアントポートを入力します。

### Login

ユーザ名を指定します。

### Password

前のフィールドで指定したユーザ名に対応するパスワードを入力します。

### Language

サーバからの情報を表示する際の言語を指定します。

例：英語には`en`、ドイツ語には`de`など。

### Logical system

論理システムの名前です。

### Gateway host

ゲートウェイのホストコンピュータ名です。

SAPサーバがゲートウェイのホストでない場合にこのフィールドに入力してください。

#### Gateway service

SAP R/3システムの接続ゲートウェイのTCP/IPソケット名です。バックグラウンドでSAP R/3システムに接続する全システムは、エントリポイントでこのソケットを使用します。

#### Program ID

プログラムの識別子です。

### Configure the reconnection parameters (詳細モード)

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the reconnection parameters (詳細モード) [ 献 29]」の節を参照してください。

### Configure the JVM

このページの使用方法については、「コネクタの設定 [ 献 15]」章の「Configure the JVM [ 献 40]」の節を参照してください。



## 8 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、本番環境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。

- **ひな型となる実務シナリオ**

これらのシナリオは、実際に発生する統合処理に対応していますが、本番環境で使用するにはユーザが手直しの必要があります。

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。

本章ではこのような実務シナリオを、アイコン  で区別します。

- **実務シナリオ**

これらのシナリオは、実際に発生する統合処理に対応しているため、本番環境でそのまま使用できます。

本章ではこのような実務シナリオを、アイコン  で区別します。

---

## Altirisシナリオ

本節では、Altiris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。

### altiris\altiris61ac44\altirisac.scn

このシナリオで、AltirisデータベースからAssetCenterデータベースにデータを転送できます。

転送されるデータを以下に挙げます。

- コンピュータ
- インストールされているソフトウェア
- モニタ
- インストールされているカード
- 論理ハードドライブ
- 物理ハードドライブ

### altiris\altiris61ac44\altirisac\_swnorm.scn

このシナリオで、AltirisデータベースからAssetCenterデータベースにデータを転送できます。

このシナリオは、ソフトウェアの正規化に使用します。

### altiris\altiris61ac50\altirisac.scn

このシナリオは、シナリオaltiris\altiris61ac44\altirisac.scn [ 献 374]と同一の機能を持ちます。

### altiris\altiris61ac50\altirisac\_swnorm.scn

このシナリオは、シナリオaltiris\altiris61ac44\altirisac\_swnorm.scn [ 献 374]と同一の機能を持ちます。

---

## 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。

## ars\rsm\rsm4ac36\cmpo.scn



このシナリオで、Action Request SystemデータベースからAssetCenter 3.6データベースにデータを転送できます。

このシナリオが転送するデータを以下に挙げます。

- 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

## ars\rsm\rsm4ac36\sharedat.scn



このシナリオで、Action Request SystemデータベースからAssetCenter 3.6データベースにデータを転送できます。

このシナリオが転送するデータを以下に挙げます。

- 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

---

## Asset Insightシナリオ

本節では、Asset Insight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。

## ist\ist4ac36\istac.scn



このシナリオで、Asset Insight 4データベースからAssetCenter 3.6データベースにデータを転送できます。

転送されるデータを以下に挙げます。

- コンピュータ
- デバイス
- ソフトウェア

## ist\ist4ac43\istac.scn



このシナリオで、Asset Insight 4データベースからAssetCenter 4.3データベースにデータを転送できます。

## ist\ist4ac44\istac.scn



このシナリオは、シナリオist\ist4ac36\istac.scn [ 献 376]と同一の機能を持ちます。

## ist\ist4ac50\istac.scn



このシナリオは、シナリオist\ist4ac36\istac.scn [ 献 376]と同一の機能を持ちます。

---

## Asset Managementシナリオ

本節では、Asset 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。

## ac\ac43\catalog.scn



このシナリオで、カタログ関連データをAssetCenterデータベースにインポートできます。



詳細については、AssetCenter 『照合更新』の「サンプルシナリオ」の章を参照してください。

### ac\ac44\catalog.scn

このシナリオは、シナリオ ac\ac43\catalog.scn [ 献 376]と同一の機能を持ちます。

### ac\ac50\catalog.scn

このシナリオは、シナリオ ac\ac43\catalog.scn [ 献 376]と同一の機能を持ちます。

---

## 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、依頼テーブルのレコードに対応する購入依頼（依頼テーブルのレコードに対応）を、Eメールアドレス（調達責任者のアドレス）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	amRequestDst

## mail\mailac36\newemplo.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部署と従業員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\mailac43\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	amRequestDst

## mail\mailac43\finreque.scn



このシナリオは、依頼テーブルのレコードに対応する購入依頼（依頼テーブルのレコードに対応）を、Eメールアドレス（調達責任者のアドレス）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\mailac43\newemplo.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部署と従業員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

## Enterprise Discoveryシナリオ

本節では、Enterprise 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。

これらのシナリオは別マニュアルに記載されています。『*Enterprise Discovery - Customization (Enterprise Discovery - カスタマイズ)*』マニュアルの「*Getting your data in AssetCenter (AssetCenterにあるデータの取得)*」の章を参照してください。

- ▶ Connect-It - AssetCenterデータベースとの統合

## ed\ed2ac44\edac.scn



このシナリオを使用して、インベントリデータをAssetCenterアプリケーションに転送します。

- ▶ Connect-Itマニュアル、『AssetCenterデータベース統合ソリューション』、「マッピング - 例」

## ed2ac44\edac-hpovcmse.scn



このシナリオで、Enterprise DiscoveryデータベースおよびRadia Service EventsデータベースからAssetCenterデータベースにデータを転送できます。

節「[hpovcm\im42iac44\imac-se.scn](#) [ 献 390]」には、AssetCenterデータベースに行う必要のあるカスタマイズと、シナリオの動作方法が説明されています。

 **注意:**

シナリオをロードすると、警告メッセージが表示されます。このメッセージでは、シナリオのマップテーブルで使用するデータを定義するように求められます。

## [ed\ed2ac44\edac-reconc.scn](#)



このシナリオを使用して、AssetCenterアプリケーションのインベントリデータを照合更新します。

## [ed\ed2ac44\edac-swnorm.scn](#)



このシナリオは、AssetCenterにあるソフトウェアの正規化モジュールにアクセスするのに使用します。

## [ed\ed2ac50\edac.scn](#)



このシナリオを使用して、インベントリデータをAssetCenterアプリケーションに転送します。

## [ed\ed2ac50\edac-reconc.scn](#)



このシナリオを使用して、AssetCenterアプリケーションのインベントリデータを照合更新します。

## [ed\ed2ac50\edac-swnorm.scn](#)



このシナリオは、AssetCenterにあるソフトウェアの正規化モジュールにアクセスするのに使用します。

## ed\ed2sc51\edsc.scn



このシナリオは、インベントリデータをServiceCenterアプリケーションに転送します。

## ed\ed2sc60\edsc.scn



このシナリオは、インベントリデータをServiceCenterアプリケーションに転送します。

## ed\ed2sc61\edsc.scn



このシナリオは、シナリオed\ed2sc60\edsc.scn [ 献 381]と同一の機能を持ちます。

## ed\ed21ac44\edac.scn

シナリオでは、以下のマッピングが使用されます。

- **Scanned Computers** : このマッピングを使用して、インベントリデータを以下の項目にマップします。
  - ソフトウェアアプリケーション (ライセンスと使用状況)
  - コンピュータコンポーネント (ネットワークカード、CPU、論理および物理ディスクドライブ、モニタ、拡張カード)
- **Network Devices** : このマッピングはネットワークコンポーネントを扱います (スイッチ、ルータなど)。
- **Non-Scanned Computers** : このマッピングは、スキャンで戻りデータが無かったコンピュータを扱います (ネットワークインベントリが起動しなかった場合と、エージェントがコンピュータにインストールされていなかった場合)。
- **Network Device Structures** : このマッピングはネットワークデバイスを扱い、バックプレーン、バックプレーンに接続されているカードなどの内部コンポーネントに関する詳細情報の取得に使用されます。コンポーネントの階層構造がキャプチャされ、保持されます。
- **Connections** : このマッピングは、ネットワークデバイスの接続間の詳細 (ネットワークトポロジ) を生成します。

### スキャン検出コンピュータマッピング

ソフトウェアの使用状況に関する部分がこのマッピング用に変更されています。

以前のバージョンのEnterprise Discoveryのソフトウェア使用状況はコンピュータ毎に計算されていました。現行バージョンでは、ソフトウェアの使用状況はコンピュータとユーザ毎に計算されます。

以前のバージョンとの互換性を維持するため、2つのユーザが作成されました。

- **ALL USERS** : 特定のコンピュータの全ユーザです。  
個別のユーザが利用不可能である場合、ソフトウェア使用状況データはALL USERSに割り当てられます。  
個別のユーザが特定のコンピュータに関連付けられている場合、ALL USERSデータは使用できません。
- **SYSTEM USER** : 特定のコンピュータコンピュータの、ユーザに割り当てることができないタスクに関連する使用状況データを取得します。

## ネットワークデバイスマッピング

このマッピングはネットワークデバイスに関連するデータを取り扱います。

以前のバージョンのEnterprise Discoveryでは、edac.scnシナリオは関連情報を取得するためにいくつかのマッピングに分割されていました。これは、特定のコンピュータの競合する情報を取得する際に特に有効でした。

Network Devicesマッピングを考察することで、情報が取得される様子が理解できます。Scanned ComputersマッピングはhwNetworkData.hwNetworkCardsコレクションを使用します。Network Devicesマッピングでは、既に利用できなくなっているNetworkCardsの代わりに、Enterprise DiscoveryのPortテーブルが使用されます。

## 非スキャン検出コンピュータマッピング

このマッピングはネットワークデバイスのマッピングに似ていて、ネットワークデバイスに関連する情報をインポートすることなく、コンピュータ（スキャン検出または非スキャン検出）コンピュータに関連する情報をインポートできます。

## ネットワークデバイス構造体マッピング

このマッピングは、AssetCenterデータベース用にネットワークデバイスの内部構造を用意します。階層の再生成には、AddOnコレクションと親テーブルへのリンクが使用されます。

## 接続マッピング

このマッピングは、ネットワークデバイスに関連する全情報が処理され、AssetCenterデータベースに保存された後に実行されます。このマッピングを使用して、異なるデバイス間の接続状況を表します。

ed\ed21ac44\edac.scn



このシナリオは、シナリオed\ed21ac44\edac.scn [ 献 381]と同一の機能を持ちます。

以下のマッピングが追加されています。

- *Network-Devices-Connections* : このマッピングは、ネットワークデバイス間の接続を処理し、AssetCenterの**Network equipment**という名前の業務サービスの作成に使用されます。

ネットワーク要素は以下のように分類されます。

- コンピュータは、タイプがネットワークデバイスである親を常に持ちます。
- 親子関係はネットワークデバイス間には存在しません。
- 終端ノードは、ネットワークデバイスリソースの子です。

## ed\ed21ac50\edac.scn



このシナリオは、シナリオed\ed21ac44\edac.scn [ 献 381]と同一の機能を持ちます。

このシナリオは、仮想マシンを考慮に入れるようになりました。仮想マシンとその仮想マシンをホストする物理マシン間のリンクは、AssetCenterに伝播されません。

## ed\ed21sc61\edsc.scn



このシナリオは、インベントリデータをServiceCenterアプリケーションに転送します。

シナリオでは、以下のマッピングが使用されます。

- *ICM-Mainframe* : このマッピングは、メインフレームデバイスに関連する情報を処理します。
- *ICM-Telecom* : このマッピングは、テレコミュニケーション装置（電話、PABXなど）に関連する情報を処理します。
- *ICM-Computer* : このマッピングは、コンピュータワークステーションに関連する情報を処理します。
- *ICM-Network Component* : このマッピングは、ネットワークコンポーネント（スイッチ、ルータなど）に関連する情報を処理します。
- *ICM-Office Electronic* : このマッピングは、コンピュータ以外の電子機器（ファクシミリ、ビデオプロジェクタ、コピー機など）に関連する情報を処理します。

物理デバイスに使用するマッピングのほか、以下のマッピングはソフトウェアに関連する情報を処理します。

- *PC Software Uninstall* : このマッピングは、*PC Software Uninstall*マッピングの前に起動する必要があります。
- *PC Software (Install)* : このマッピングは、各コンピュータにリンクされている新規情報、または既存情報を再書き込みします。

## ed\ed21sc62\edsc.scn

このシナリオは、シナリオed\ed21sc61\edsc.scn [ 献 383]と同一の機能を持ちます。



## ed\ed21sd45\edsc.scn



このシナリオは、インベントリデータをService Deskアプリケーションに転送します。

シナリオでは、以下のマッピングが使用されます。

- CIM\_V2.0 DTDで定義されているXMLデータ構造体へのマッピング。このDTDは、同一DEVICEに属するフィールドがPROPERTIES配列にマップされるデータ構造体を定義します。各プロパティは、NAME、TYPE、TYPEおよびVALUEを持ちます。各DEVICEはVALUE.OBJECTS配列を形成するINSTANCEにマップされます。
- 次に、XMLドキュメントはService Deskコネクタが解析を行い、PROPERTIESをService Desk固有のテーブルとフィールドにインポートし、マップします。

Service Desk Data Exchange TaskがService Deskのインポートが完了すると、XMLドキュメントがConnect-Itシナリオで生成されます。このタスクはService Deskインポートマッピングを使用するように設定されています。

## ed\ed22ac50\edac.scn



このシナリオは、シナリオed\ed21ac44\edac.scn [ 献 381]と同一の機能を持ちます。

このシナリオは、仮想マシンを考慮に入れるようになりました。仮想マシンとその仮想マシンをホストする物理マシン間のリンクは、AssetCenterに伝播されません。



---

## iInventoryシナリオ

本節では、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。

### iinv5ac43.scn\iinvac.scn



このシナリオで、Microsoft Access（ハードウェアとソフトウェアの情報）データベースからAssetCenter 4.3のコンピュータテーブルにデータを転送できます。

### iinv5ac44.scn\iinvac.scn



このシナリオは、シナリオiinv5ac43.scn\iinvac.scn [ 献 385]と同一の機能を持ちます。

### iinv5ac50.scn\iinvac.scn



このシナリオは、シナリオiinv5ac43.scn\iinvac.scn [ 献 385]と同一の機能を持ちます。

### iinv5sc51.scn\iinvsc.scn



このシナリオを実行する前に、pcsoft-uninst.unlファイルとpcsoftware.unlファイルをServiceCenterにインポートする必要があります。これらのファイルは、Connect-Itのインストール（インストールフォルダのdatakit\scフォルダ）にあります。

このシナリオで、iInventory LANAuditorが取得したデータを使用することで、ServiceCenter 5.1にあるコンピュータ（とそのソフトウェア）に関する情報を更新できます。

### iinv5sc60.scn\iinvsc.scn



このシナリオで、iInventory LANAuditorが取得したデータを使用することで、ServiceCenter 6にあるコンピュータ（とそのソフトウェア）に関する情報を更新できます。

### iinv5sc61.scn\iinvsc.scn



このシナリオは、シナリオldsk\ld8sc51\ldsksc.scn [ 献 387]と同一の機能を持ちます。

### iinv5sc62.scn\iinvsc.scn



このシナリオは、シナリオiinv5sc60.scn\iinvsc.scn [ 献 385]と同一の機能を持ちます。

---

## LANDeskシナリオ

本節では、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。

### ldsk\ld8ac43\ldskac.scn



このシナリオで、LANDesk 8データベースからAssetCenter 4.3データベースにデータを転送できます。

- コンピュータ
- デバイス
- ソフトウェア

### ldsk\ld8ac44\ldskac.scn



このシナリオは、シナリオldsk\ld8ac43\ldskac.scn [ 献 386]と同一の機能を持ちます。

## ldsk\ld8ac50\ldskac.scn



このシナリオは、シナリオldsk\ld8ac43\ldskac.scn [ 献 386]と同一の機能を持ちます。

## ldsk\ld8sc51\ldsksc.scn



このシナリオを実行する前に、**pcsoft-uninst.unl**ファイルと**pcsoftware.unl**ファイルを**ServiceCenter**にインポートする必要があります。これらのファイルは、**Connect-It**のインストール（インストールフォルダの**datakit\sc**フォルダ）にあります。

このシナリオで、**LANDesk8**が取得したデータを使用することで、**ServiceCenter 5.1**にあるコンピュータ（とそのソフトウェア）に関する情報を更新できます。

## ldsk\ld8sc6\ldsksc.scn



このシナリオは、シナリオldsk\ld8sc51\ldsksc.scn [ 献 387]と同一の機能を持ちます。

## ldsk\ld8sc61\ldsksc.scn



このシナリオは、シナリオldsk\ld8sc51\ldsksc.scn [ 献 387]と同一の機能を持ちます。

## ldsk\ld8sc62\ldsksc.scn



このシナリオは、シナリオldsk\ld8sc51\ldsksc.scn [ 献 387]と同一の機能を持ちます。

## ldskws\ld8ac50\LDStartTasks.scn



このシナリオは、受信した**LANDesk**情報に応じて以下のテーブルを更新します。

- amESDPackage

- amESDDelivMethod

## ldskws\ld8ac50\LDUpdateRepository.scn



このシナリオは、AssetCenter amESDTaskスケジュールされたタスクテーブルにある情報に基づく、LANDesk Software Distribution内にスケジュールされたタスクを作成します。

## ldskws\ld8ac50\LDUpdateTask.scn



このシナリオは、AssetCenter amESDTaskスケジュールされたタスクテーブルにある情報に基づく、LANDesk Software Distribution内のスケジュールされたタスクを更新します。

---

## OpenView Configuration Managementシナリオ

本節では、Configuration 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。

### hpovcm\im42iac44\imac.scn

このシナリオで、OpenView Inventory ManagerデータベースからAssetCenterデータベースにデータを転送できます。

転送されるデータを以下に挙げます。

- コンピュータ
- 論理ハードドライブ
- 物理ハードドライブ
- ネットワークカード
- 内部デバイス

### hpovcm\im42iac44\imac\_swnorm

このシナリオで、OpenView Inventory ManagerデータベースからAssetCenterデータベースにデータを転送できます。

このシナリオは、ソフトウェアの正規化に使用します。

## hpovcm\im42iac44\imac-usage.scn

このシナリオで、OpenView Inventory ManagerデータベースとUsage ManagerデータベースからAssetCenterデータベースにデータを転送できます。

このシナリオで、コンピュータとユーザに関するデータをAssetCenterデータベースに転送できます。

### 特別な動作

Radia Usage Managerテーブルでは、各**ComputerNames**レコードはコンピュータとユーザを表します。これは、同一コンピュータに複数ユーザが存在する場合、Raidaインベントリは、そのコンピュータのユーザ数と同数のレコードを生成することを意味します。

AssetCenterでは、アプリケーションの使用状況に関する情報は**ソフトウェアのインストールまたは使用**テーブルに記録され、**使用**タブで表示できます。特定アプリケーションに複数のユーザが存在しても構いません（各ユーザは部署と従業員テーブルに記録されます）。このため、それぞれのユーザによるアプリケーションの使用状況を表す**ソフトウェアのインストールまたは使用**テーブルには複数の行が存在することになります。

両アプリケーション間における動作の差異の照合更新に使用されるフィルタは、**amSoftInstall**構造体で定義されます。

```
seType=1
```

このフィルタは使用されるアプリケーションの定義に使用されます（**インストールタイプ**は使用）。

### パフォーマンス

データ処理速度を改善するため、以下の動作が定義されています。

- コンピュータの使用状況に関する情報がUsage Managerデータベースに存在せず、コンピュータがAssetCenterデータベースに存在しない場合、無視されます。
- Radia Usage Managerに情報のないアプリケーションは、AssetCenterには挿入されません。
- 詳細な照合更新は、ソフトウェアに関する**AddOn**コレクションには実装されていません。

詳細な照合更新が有効である場合、Inventory Managerデータベースでは検出され、User Managerデータベースでは検出されないソフトウェアは、タイプ6"absent"に設定されます。

- "ProductVersion"、"ProductName"、および"CompanyName"のみを考慮する**rcaFilePropertyNames**フィルタが、Usage Managerコネクタの**ComputersName**生成ドキュメントタイプに適用されます。
- Usage Managerデータベースのインデックスが作成されています。

- **rcaWindowsFileUsage**テーブルの**WindowsComputerUser\_id**列のインデックス

MS SQL Serverデータベースの例：

```
CREATE
INDEX [idxCIT_rcaWindowsFileUsage_WindowsComputerUser_id] ON
[dbo].[rcaWindowsFileUsage] ([WindowsComputerUser_id])
ON [PRIMARY]
```

- **rcaFileSignatureProperties**テーブルの**FileSignature\_id**列のインデックス

MS SQL Serverデータベースの例：

```
CREATE
INDEX [idxCIT_rcaFileSignatureProperties_FileSignatureId] ON [dbo].
[rcaFileSignatureProperties] ([FileSignature_id])
ON [PRIMARY]
```

## hpovcm\im42iac44\imac-se.scn

このシナリオで、OpenView Inventory ManagerデータベースおよびRadiaデータベースからAssetCenterデータベースにデータを転送できます。

### AssetCenterデータベースのカスタマイズ

付属シナリオが正しく機能するよう、AssetCenterデータベースをカスタマイズする必要があります。



#### 警告:

カスタマイズする前に、AssetCenterデータベースのバックアップコピーを作成することを推奨します。

変更内容を以下に挙げます。

- 以下のフィールドを**amSoftInstall**テーブルに追加し、アンインストール日を挿入できるようにします。
  - SQL名：dtUninstalledByRadia
  - ラベル：Radiaによるアンインストール日
  - 説明：Radiaによるアンインストール日
  - タイプ：日付+時刻
  - 履歴の保持：はい
- 以下のフィールドを**amSoftInstall**テーブルに追加し、インストール日を挿入できるようにします。
  - SQL名：dtInstalledByRadia

- ラベル：Radiaによるインストール日
- 説明：Radiaによるインストール日
- タイプ：日付+時刻
- 履歴の保持：はい
- 以下のフィールドを**amSoftInstall**テーブルに追加し、Radiaがアプリケーションをインストールしたかどうかを示すことができますようにします。
  - SQL名：deployedByRadia
  - ラベル：Radiaによる展開
  - 説明：Radiaによる展開（1=はい 0=いいえ）
  - タイプ：ブール
- 以下のフィールドを**amSoftInstall**テーブルに追加し、使用されているサービス名を示すことができますようにします。
  - SQL名：RadiaService
  - ラベル：Radiaサービス
  - 説明：アプリケーションの展開に使用したRadiaサービス（カンマ区切り）
  - タイプ：テキスト（サイズ150）
  - このフィールドのインデックスを作成：オプションを選択
- 以下のフィールドを**amSoftInstall**テーブルに追加し、動作中のコンポーネントを示すことができますようにします。
  - SQL名：RadiaCmpt
  - ラベル：Radiaコンポーネント
  - 説明：使用するRadiaコンポーネント
  - タイプ：テキスト（サイズ50）

シナリオの実行後、コンピュータやソフトウェアのインストール画面にある**アプリケーション**タブに情報が表示されます。

 **注意:**

カスタマイズは、付属シナリオを使用する前に実行する必要があります。

### シナリオの動作

付属シナリオを使用して、HP OpenView Service Eventsデータが入力されているインベントリデータをAssetCenterデータベースに統合します。

不必要なレコードが挿入されることを防ぐため、以下のルールを使用します。

- インベントリツールデータベース（Enterprise Discovery、Inventory Managementなど）にあるコンピュータ関連の任意の情報は、AssetCenterデータベースモードの*Update or insert*を使用してマップされます。
- Service Eventsコネクタから返される任意の情報は、AssetCenterデータベースモードの*Update only*を使用してマップされます。

- コンピュータに関連する情報が2つのソースデータベース（インベントリツールまたはサービスイベント）のいずれか一方にのみ存在し、そのコンピュータに関連するレコードがAssetCenterデータベースに存在しない場合、AssetCenterデータベースにレコードは挿入されません。
- コンピュータに関連する情報が2つのソースデータベース（インベントリツールまたはサービスイベント）のいずれか一方にのみ存在し、そのコンピュータに関連するレコードがAssetCenterデータベースに存在しない場合、ソースデータベースにある新しい情報を使用して、レコードが更新されます。

## データの照合更新

インベントリデータをService Eventsコネクタからのデータと照合更新するため、共通モデルをAssetCenterで定義する必要があります。共通モデルを使用することで、インベントリやService Eventsコネクタから返されるコンピュータ関連の情報を正しく識別し、AssetCenterデータベースに挿入します。

データの照合更新を行うため、以下の命名規則が定義されています。

- ED / ACやHP OVCM IM / ACの統合では、ソフトウェアのインストールモデルは、"アプリケーション名" + " " + "バージョン"です。
- HP OVCMツールでは、HP OVCM展開アプリケーション（AppEventテーブルのapp\_nameフィールド）のShort Descriptionフィールドは、インベントリツールによって決められている"アプリケーション名" + " " + "バージョン"命名規則を使用する必要があります。

統合シナリオで、app\_nameフィールドはModel Nameフィールドにマップされます。

HP OVCMツールを使用して展開を作成する際に命名規則が適用されていない場合、照合更新は実行されません。

## 照合更新の状況

照合更新に関連する状況として以下の3つの状況があります。

- HP OVCM展開ツールを使用した単一の展開が、AssetCenterの単一ソフトウェアのインストールに対応する状況。

展開のShort DescriptionがAssetCenterモデル名を使用した照合更新に使用されます。

- HP OVCM展開ツールを使用した単一の展開が、AssetCenterの複数ソフトウェアのインストールに対応する状況。

この状況は、インベントリツールがソフトウェアスイートに含まれる各アプリケーションをインベントリする場合に発生することがあります。たとえば、インベントリツールが"Microsoft Office 2003"に含まれる各アプリケーション"Excel 2003"、"Word 2003"、"PowerPoint 2003"などをカウントし、"Microsoft Office 2003"スイート自体を参照しない場合です。HP OVCM展開ツールでは、これらのアプリケーションの展開は"Microsoft Office 2003"と呼ぶ単一の展開に対応します。



この場合、*RadiaService*<インベントリツール>*Application* マップテーブル（ここで、<インベントリツール>は、使用しているEnterprise DiscoveryまたはInventory Managerのいずれかのインベントリツールに対応します）に、HP OVCMツールが展開したアプリケーションに対応するキーと、コンポーネントリストの値を入力する必要があります。

"Microsoft Office 2003"キーの場合であれば、以下の値を定義します。

"Microsoft Office 2003"   "Excel 2003"   "Word 2003"   "PowerPoint 2003"
--------------------------------------------------------------------------

このマップテーブルはradia\_product.mptファイルにあります。

マップテーブルが定義すむと、Connect-ItはHP OVCM "Microsoft Office 2003"展開を、AssetCenterに保存されているすべてのソフトウェアのインストールと照合更新します。次にConnect-Itは、HP OVCMデータベースからのデータをスイートに属す各ソフトウェアのインストールに伝播します。

- HP OVCM展開ツールを使用した複数の展開が、AssetCenterの単一ソフトウェアのインストールに対応する状況。

この状況は、インベントリツールがソフトウェアスイートのみを検出し、そのスイートに含まれるアプリケーションを検出しない場合に発生することがあります。たとえば、インベントリツールが"Microsoft Office 2003"をカウントし、インストールされているアプリケーション ("Excel 2003"、"Word 2003"、"PowerPoint 2003"など) をカウントしない場合です。HP OVCM展開ツールでは使用されている展開の数は、アプリケーションに等しくなることがあります。

この場合、*RadiaService*<インベントリツール>*Application* マップテーブル（ここで、<インベントリツール>は、使用しているEnterprise DiscoveryまたはInventory Managerのいずれかのインベントリツールに対応します）に、HP OVCMツールが展開したアプリケーションに対応するキーと、ソフトウェアスイートの値を入力する必要があります。

"Microsoft Office 2003"キーの場合であれば、以下の値を定義します。

"PowerPoint 2003"   "Microsoft Office 2003"
---------------------------------------------

"Word 2003"   "Microsoft Office 2003"
---------------------------------------

"Excel 2003"   "Microsoft Office 2003"
----------------------------------------

次にConnect-Itは"PowerPoint 2003"、"Word 2003"およびその他の展開を、AssetCenterで検出された"Microsoft Office 2003"ソフトウェアのインストールと照合更新します。

次にConnect-Itは、HP OVCMデータベースからのデータを対応するソフトウェアのインストールに伝播します。



#### 注意:

既存のRadiaインフラストラクチャとの統合を行う際、付属の事前定義されたマップテーブルを使用できます（Radiaデータベースには既にデータが入力されているとします）。この場合、事前定義されている命名規則は既存の命名規則とは対応しません。

この場合、展開の*Short Description*フィールドに対応するキーと、AssetCenterデータベースでの対応する値とを使用して、`radia_product.mpt`にあるマップテーブルを更新します。

### hpovcm\im42iac50\imac.scn

このシナリオは、シナリオhpovcm\im42iac44\imac.scn [ 献 388]と同一の機能を持ちます。

照合更新は、FullName要素ではなくBarcode要素に対して実行されるようになりました。

Nature要素のsysComputer値は、CPU値で置き換えられました。

### hpovcm\im42iac50\imac\_swnorm

このシナリオは、シナリオhpovcm\im42iac44\imac\_swnorm [ 献 388]と同一の機能を持ちます。

### hpovcm\im42iac50\imac-usage.scn

このシナリオは、シナリオhpovcm\im42iac44\imac-usage.scn [ 献 389]と同一の機能を持ちます。

### hpovcm\im42iac50\imac-se.scn

このシナリオは、シナリオhpovcm\im42iac44\imac-se.scn [ 献 390]と同一の機能を持ちます。

### hpovcm\im42sc61\imsc.scn

このシナリオを実行する前に、`pcsoft-uninst.unl`ファイルと`pcsoftware.unl`ファイルをServiceCenterにインポートする必要があります。これらのファイルは、Connect-Itのインストール（インストールフォルダの`datakit\sc`フォルダ）にあります。

このシナリオで、OpenView Inventory ManagerデータベースからServiceCenterデータベースにデータを転送できます。

 **注意:**

シナリオをロードすると、警告メッセージが表示されます。このメッセージでは、シナリオのマッピングテーブルで使用するデータを定義するように求められます。

## hpovcm\im42sc62\imsc.scn

このシナリオは、シナリオhpovcm\im42sc61\imsc.scn [ 献 394]と同一の機能を持ちます。

## Mercuryシナリオ

本節では、Mercury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。

これらのシナリオには、独立したマニュアルがあります。

『ServiceCenter-MAM/BAC統合ソリューション』を参照してください。

## merq\mam242sc61



シナリオ :

- **scmam-ci.scn**

このシナリオを使用して、ServiceCenterからMAMにデータを転送します。シナリオは、構成アイテム（デバイス）をServiceCenterからエクスポートし、MAMにインポートします。

- **mamsc-ci-service.scn**

このシナリオを使用して、ServiceCenterからMAMにデータを転送します。このシナリオは、Business ServicesとBusiness ServicesとMAM間の関係をエクスポートし、ServiceCenterにデバイスとしてインポートします。

- **mamsc-ci.scn**

このシナリオを使用して、MAMからServiceCenterにデータを転送します。このシナリオは構成アイテム（デバイス）と構成アイテムとMAM間の関係をエクスポートし、ServiceCenterにインポートします。構成アイテムの照合更新にはMACアドレスが使用されます。

## merq\bac51sc61



シナリオ :

- **scbac-ci.scn**  
このシナリオを使用して、ServiceCenterからBACにデータを転送します。  
シナリオは、構成アイテム（デバイス）をServiceCenterからエクスポートし、BACにインポートします。
- **scbac-changes.scn**  
このシナリオを使用して、ServiceCenterからBACにデータを転送します。  
このシナリオは、変更と関連構成アイテムをServiceCenterからエクスポートし、BACにインポートします。このシナリオは、変更に関するServiceCenter URLをエクスポートし、BAC内でServiceCenter Webページを表示できるようにします。
- **bacsc-incident.scn**  
このシナリオを使用して、BACからServiceCenterにデータを転送します。  
このシナリオは、ServiceCenterでサポートチケットを自動的に開きます。
- **bacsc-ci.scn**  
このシナリオを使用して、BACからServiceCenterにデータを転送します。  
このシナリオは構成アイテム（デバイス）と構成アイテムとBAC間の関係をエクスポートし、ServiceCenterにインポートします。

## merq\bac51sc62



このディレクトリにあるシナリオは、merq\bac51sc61 [ 献 396]シナリオと同一の機能を持ちます。

## merq\mam242sc61



このシナリオは、シナリオmerq\mam242sc61 [ 献 395]と同一の機能を持ちます。

---

## 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\ac44mq\mqtoprgn.scn



このシナリオで、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できます。

- 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 (amPOOrder)
- 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\ac44mq\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\sc51mq\mqsc.scn



このシナリオは、

- PEREGRINE.INキューにあるMQSeriesメッセージから、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から、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
ServiceCenterの連絡先	ServiceCenterにある連絡先

### mqseries\sc60mq\mqsc.scn



このシナリオは、シナリオmqseries\sc51mq\mqsc.scn [ 献 397]と同一の機能を持ちます。

### mqseries\sc61mq\mqsc.scn



このシナリオは、シナリオmqseries\sc51mq\mqsc.scn [ 献 397]と同一の機能を持ちます。

---

## 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



このシナリオで、LDAPディレクトリからAsset ManagementアプリケーションのDepartments and Employeesテーブルにデータを転送できます。

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



このシナリオで、LDAPディレクトリからAsset ManagementアプリケーションのDepartments and Employeesテーブルにデータを転送できます。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\names.scn



このシナリオで、LDAPディレクトリからLotus Notesデータベースにデータを転送できます。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

## Network Node Managerシナリオ

本節では、NNM統合用に用意されているシナリオについて簡潔に説明します。

### nnm755ac50\nnmac



#### 前提条件

- 統合シナリオは、NNMプログラムと同じサーバ上で実行する必要があります
- Python 2.5がインストールされている必要があります（さらに、python.exeが環境変数で宣言されている必要があります）

## シナリオの動作方法

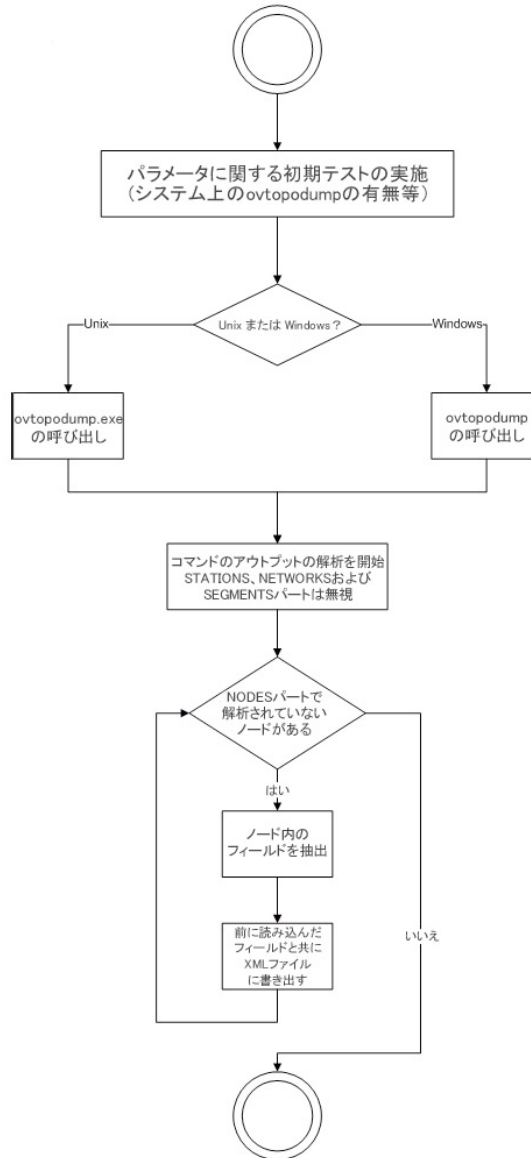
本シナリオは、`ovtopodump`コマンドラインツールを使用します。

`l` (ロング) および `r` (再帰) オプションの両方を `ovtopodump` で使用することで、**NNM** オブジェクトの詳細なリストを取得し、ツールが全ノードタイプ **NNM** ブランチを対象とすることを指定します。

スクリプトは、**Windows** および **Unix** 環境の両方にある **Python** スクリプトを経由して呼び出されます。**Python** スクリプトは、ファイルを **XML** フォーマットで書き出します。**XML** ファイルはその後、**Connect-It** が処理します。



スクリプトのロジックを以下に挙げます。



シナリオは以下に挙げる2つの部分に分けられます。

- Pythonスクリプトがコマンドラインコネクタを経由して呼び出されます
- XMLファイルとAssetCenter間のマッピングです。XMLファイルはXMLコネクタが処理します。

## Mapping Node-> amComputer

ターゲット	マッピング/ソース	コメント
Name	[label]	
ComputerDesc	[description]	
IpxSpxAddress	If [IPXAddress] = "00000000:000000000000" Then PifIgnoreNodeMapping Else RetVal = [IPXAddress] End If	
IpxSpxServer	[IPXServerName]	
TcpIpHostName	[hostname]	
TcpIpAddress	If [interfaces.interface(0).IPAddress] <> "" Then RetVal = [interfaces.interface(0).IPAddr] Else PifIgnoreNodeMapping End If	
PhysicalAddress	If [interfaces.interface(0).physicalAddress] <> "" Then RetVal = RightPart([interfaces.interface(0).physicalAddress], "x", 1) Else PifIgnoreNodeMapping End If	
Portfolio		
Folder [UUID]		
Folder Portfolio.Asset		
ExternalAssetID [ID]		
Name	RetVal = PifStrVal("MODEL_UNKNOWN_COMPUTER_MODEL")	
Portfolio.Asset.Model.Nature		
Code	"CPU"	
NetworkCards	If [interfaces.interface.physicalAddress] = "<none>" Then Ignore the current interface End If	[interfaces.interface]
Description	[interfaces.interface.description]	Descriptionが空欄の場合、レコードは無視されます。
PhysAddress	RetVal = RightPart([interfaces.interface.physicalAddress], "x", 1)	
TcpIpAddress	[interfaces.interface.IPAddr]	

ターゲット	マッピング/ソース	コメント
SubnetMask	[interfaces.interface.IPMask]	



#### 注意:

太字のフィールド (TcpIpHostName、PhysAddress) は照合更新キーを持ちます。

## nnm755sc62\nnmsc



このシナリオで、ノードタイプデータをNNMアプリケーションからServiceCenterアプリケーションに転送できます。

## NTセキュリティシナリオ

本節では、NT Security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\ntac36\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\ntac36\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\ntac42\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 4.2データベースへインポートします。

## ntsec\ntac42\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 4.2データベースへインポートします。

## ntsec\ntac43\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 4.3データベースへインポートします。

## ntsec\ntac43\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 4.3データベースへインポートします。

## ntsec\ntac44\addcpu.scn



このシナリオは、シナリオntsec\ntac43\addcpu.scn [ 献 404]と同一の機能を持ちます。

## ntsec\ntac44\adduser.scn



このシナリオは、シナリオntsec\ntac43\adduser.scn [ 献 404]と同一の機能を持ちます。

## ntsec\ntac50\addcpu.scn



このシナリオは、シナリオntsec\ntac43\addcpu.scn [ 献 404]と同一の機能を持ちます。

## ntsec\ntac50\adduser.scn



このシナリオは、シナリオntsec\ntac43\adduser.scn [ 献 404]と同一の機能を持ちます。

---

## OpenView Network Discoveryシナリオ

本節では、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。

### pnd\pnd51ac43\pndac.scn



このシナリオで、OpenView Network Discoveryデータベースに関連するデータをAssetCenter 4.3のポートフォリオ品目テーブルに転送できます。

### pnd\pnd51sc51\pndsc.scn



このシナリオで、OpenView Network Discoveryデータベースに関連するデータをServiceCenter 5.1に転送できます。

### pnd\pnd51sc60\pndsc.scn



このシナリオは、シナリオpnd\pnd51sc51\pndsc.scn [ 献 405]と同一の機能を持ちます。

### pnd\pn52dac43\pndac.scn

このシナリオは、シナリオpnd\pnd51ac43\pndac.scn [ 献 405]と同一の機能を持ちます。



### pnd\pn52dac44\pndac.scn



このシナリオは、シナリオpnd\pnd51ac43\pndac.scn [ 献 405]と同一の機能を持ちます。

## pnd\pnd52ac50\pndsc.scn



このシナリオは、シナリオpnd\pnd51ac43\pndac.scn [ 献 405]と同一の機能を持ちます。

## pnd\pnd52sc51\pndsc.scn



このシナリオは、シナリオpnd\pnd51sc51\pndsc.scn [ 献 405]と同一の機能を持ちます。

## pnd\pnd52sc60\pndsc.scn



このシナリオは、シナリオpnd\pnd51sc51\pndsc.scn [ 献 405]と同一の機能を持ちます。

## pnd\pnd52sc61\pndsc.scn



このシナリオは、シナリオpnd\pnd51sc51\pndsc.scn [ 献 405]と同一の機能を持ちます。

---

## OpenView Desktop Inventoryシナリオ

本節では、OpenView 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\pdi7ac36\pdia.scn



このシナリオで、OpenView Desktop Inventoryが取得したデータを使用することで、AssetCenter 3.6データベースにあるコンピュータテーブル (amComputer) に関する情報を更新できます。

## pdi\pdi7ac42\pdiaac.scn



このシナリオで、OpenView Desktop Inventoryが取得したデータを使用することで、AssetCenter 4.2データベースにあるコンピュータテーブル (amComputer) に関するデータを更新できます。

## pdi\pdi7ac43\pdiaac.scn



このシナリオで、OpenView Desktop Inventoryが取得したデータを使用することで、AssetCenter 4.3データベースにあるコンピュータテーブル (amComputer) に関する情報を更新できます。

## pdi\pdi7sc51\pdisc.scn



このシナリオで、OpenView Desktop Inventoryが取得したデータを使用することで、ServiceCenter 5.1にあるコンピュータ (とそのソフトウェア) に関する情報を更新できます。

このシナリオを実行する前に、pcsoft-uninst.unlファイルとpcsoftware.unlファイルをServiceCenterにインポートする必要があります。これらのファイルは、Connect-Itのインストール (インストールフォルダのdatakit\scフォルダ) にあります。

## pdi\pdi7sc60\pdisc.scn

Ce scénario a les mêmes fonctions que le scénario [pdi\pdi7sc51\pdisc.scn](#) [ 献 407].



## pdi\pdi8ac43\pdiaac.scn

Ce scénario a les mêmes fonctions que le scénario [pdi\pdi7ac43\pdiaac.scn](#) [ 献 407].



## pdi\pdi8ac44\pdiaac.scn



このシナリオは、シナリオ `pdi\pdi7ac43\pdiaac.scn` [ 献 407] と同一の機能を持ちます。

シナリオでの特定マッピング問題については「**Asset Management**コネクタ [ 献 59]」の節にある「**AssetManagement**コネクタが生成するドキュメントタイプ [ 献 68]」を参照してください。

▶ **Connect-It - AssetCenter**データベースとの統合

## `pdi\pdi8ac44\pdiaac-swnorm.scn`

このシナリオは**AssetCenter**ソフトウェアライセンスモジュールのユーザ用であり、ソフトウェアモデルの照合更新を行えます。

詳細については、節「**AssetManagement**コネクタが生成するドキュメントタイプ [ 献 68]」を参照してください。

## `pdi\pdi8ac50\pdiaac.scn`



このシナリオは、シナリオ `pdi\pdi8ac44\pdiaac.scn` [ 献 407] と同一の機能を持ちます。

## `pdi\pdi8ac50\pdiaac-swnorm.scn`



このシナリオは、シナリオ `pdi\pdi8ac44\pdiaac-swnorm.scn` [ 献 408] と同一の機能を持ちます。

## `pdi\pdi8sc51\pdisc.scn`



Ce scénario a les mêmes fonctions que le scénario `pdi\pdi7sc51\pdisc.scn` [ 献 407].

## `pdi\pdi8sc60\pdisc.scn`



Ce scénario a les mêmes fonctions que le scénario `pdi\pdi7sc51\pdisc.scn` [ 献 407].



pdi\pdi8sc61\pdisc.scn



このシナリオは、シナリオpdi\pdi7sc51\pdisc.scn [ 献 407]と同一の機能を持ちます。

pdi\pdi8sc62\pdisc.scn



このシナリオは、シナリオpdi\pdi7sc51\pdisc.scn [ 献 407]と同一の機能を持ちます。

---

## Decision Centerシナリオ

本節では、Decision Center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。

rds\rds50sc51\rdssc.scn



このシナリオで、ServiceCenterデータからのデータをBiPortalで使用できるように準備します。

rds\rds52ac4\rdzac.scn



このシナリオで、BiPortalで使用できるようにAssetCenterデータベースのデータを用意します。

---

## SAP BAPIシナリオ

本節では、SAP IDOC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。

sap\sapac43\masterdata\lGetCompany.scn



このシナリオで、SAPデータベースからAssetCenterデータベースにデータを転送できます。

このシナリオが転送するデータを以下に挙げます。

- 会社
- 場所

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
GetSupplier	amCompanyDst

### sap\sapac43\masterdata\IGetCostcenter.scn



このシナリオで、SAPデータベースからAssetCenterデータベースにデータを転送できます。

このシナリオが転送するデータを以下に挙げます。

- コストセンタ

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
GetCostCenterSrc	amCostCenterDst

### sap\sapac43\masterdata\IGetPeople.scn



このシナリオで、SAPデータベースからAssetCenterデータベースにデータを転送できます。

このシナリオが転送するデータを以下に挙げます。

- 部署と従業員
- コストセンタ

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
HRMD_A01	amEmplDeptDst

## sap\sapac43\purchaseprocess\IGetInvoice.scn



このシナリオで、SAPデータベースからAssetCenterデータベースにデータを転送できます。

このシナリオが転送するデータを以下に挙げます。

- 請求書

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
GetInvoiceSrc	amInvoiceDst

## sap\sapac43\purchaseprocess\IGetOrderNbr.scn



このシナリオで、SAPデータベースからAssetCenterデータベースにデータを転送できます。

## sap\sapac44\masterdata



以下のシナリオは、AssetCenterバージョン4.3用のシナリオと同じ機能を持ちます。

- IGetCompany.scn
- IGetCostcenter.scn
- IGetPeople.scn

## sap\sapac44\purchaseprocess



以下のシナリオは、AssetCenterバージョン4.3用のシナリオと同じ機能を持ちます。

- IGetOrderNbr.scn
- IGetInvoice.scn

## sap\sapac50\masterdata



以下のシナリオは、AssetCenterバージョン4.3用のシナリオと同じ機能を持ちます。

- IGetCompany.scn
- IGetCostcenter.scn
- IGetPeople.scn

## sap\sapac50\purchaseprocess



以下のシナリオは、AssetCenterバージョン4.3用のシナリオと同じ機能を持ちます。

- IGetOrderNbr.scn
- IGetInvoice.scn

---

## SAP BAPIシナリオ

本節では、SAP BAPI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。

### sap\sapac43\purchaseprocess\BAPI\_CREATE\_REQUEST.scn



このシナリオで、SAPデータベースからAssetCenterデータベースにデータを転送できます。

このシナリオが転送するデータを以下に挙げます。

- 依頼
- サプライヤ
- 従業員

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
amPOrder	NewRequestDst

## sap\sapac43\purchaseprocess\BAPI\_CREATE\_PO.scn



このシナリオで、SAPデータベースからAssetCenterデータベースにデータを転送できます。逆方向にもデータを転送できます。

このシナリオが転送するデータを以下に挙げます。

- 発注
- 発注明細

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
amPOrderSrc	NewPurchaseOrderDst
ProcessReportNewPurchaseOrderSrc	amPOrderDst

## sap\sapac43\purchaseprocess\BAPI\_RECEIVE.scn



このシナリオで、SAPデータベースからAssetCenterデータベースにデータを転送できます。逆方向にもデータを転送できます。

このシナリオが転送するデータを以下に挙げます。

- 受領伝票
- 受領明細

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
amReceiptSrc2	BAPI_GOODSMVT_CREATEDst2
ProcessReportBAPI_GOODSMVT_CREATESrc1	amReceiptDst2

## sap\sapac44\purchaseprocess\BAPI\_CREATE\_REQUEST.scn



このシナリオは、シナリオ

sap\sapac43\purchaseprocess\BAPI\_CREATE\_REQUEST.scn [ 献 412]と同一の機能を持ちます。

sap\sapac44\purchaseprocess\BAPI\_CREATE\_PO.scn



このシナリオは、シナリオ  
sap\sapac43\purchaseprocess\BAPI\_CREATE\_PO.scn [ 献 413]と同一の機能  
を持ちます。

sap\sapac44\purchaseprocess\BAPI\_RECEIVE.scn



このシナリオは、シナリオsap\sapac43\purchaseprocess\BAPI\_RECEIVE.scn  
[ 献 413]と同一の機能を持ちます。

sap\sapac50\purchaseprocess\BAPI\_CREATE\_REQUEST.scn



このシナリオは、シナリオ  
sap\sapac43\purchaseprocess\BAPI\_CREATE\_REQUEST.scn [ 献 412]と同一  
の機能を持ちます。

sap\sapac50\purchaseprocess\BAPI\_CREATE\_PO.scn



このシナリオは、シナリオ  
sap\sapac43\purchaseprocess\BAPI\_CREATE\_PO.scn [ 献 413]と同一の機能  
を持ちます。

sap\sapac50purchaseprocess\BAPI\_RECEIVE.scn



このシナリオは、シナリオsap\sapac43\purchaseprocess\BAPI\_RECEIVE.scn  
[ 献 413]と同一の機能を持ちます。

---

## SMSシナリオ

本節では、SMS 2.xとSMS 2003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\sms2ac43\smsac.scn



このシナリオで、SMSデータベース（ITポートフォリオ）からAssetCenter 4.3データベースのamComputerテーブルに情報を転送できます。

ソース	ターゲット
System_DATASrc	amComputerDst

## sms\sms2003ac43\smsac.scn



このシナリオで、SMSデータベース（ITポートフォリオ）からAssetCenter 4.3データベースのamComputerテーブルに情報を転送できます。

## sms\sms2003ac44\smsac.scn



このシナリオは、シナリオsms\sms2003ac43\smsac.scn [ 献 415]と同一の機能を持ちます。

## sms\sms2003ac50\smsac.scn



このシナリオは、シナリオsms\sms2003ac43\smsac.scn [ 献 415]と同一の機能を持ちます。

## sms\sms2003sc51\smssc.scn



このシナリオを実行する前に、pcsoft-uninst.unlファイルとpcsoftware.unlファイルをServiceCenterにインポートする必要があります。これらのファイルは、Connect-Itのインストール（インストールフォルダのdatakit\scフォルダ）にあります。

このシナリオで、SMS2003が取得したデータを使用することで、ServiceCenter 5.1にあるコンピュータ（とそのソフトウェア）に関する情報を更新できます。

sms\sms2003sc6\smssc.scn



このシナリオは、シナリオsms\sms2003sc51\smssc.scn [ 献 415]と同一の機能を持ちます。

sms\sms2003sc61\smssc.scn



このシナリオは、シナリオsms\sms2003sc51\smssc.scn [ 献 415]と同一の機能を持ちます。

sms\sms2003sc62\smssc.scn



このシナリオは、シナリオsms\sms2003sc51\smssc.scn [ 献 415]と同一の機能を持ちます。

---

## 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。

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コネクタが発行する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。

照合更新時に、Asset Managementコネクタが発行する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。

## 入カイベントの作成

シナリオを適切に機能させるためには、以下のUNL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。このアプリケーションは、Connect-Itインストールフォルダの`datakit\sc`フォルダ内にあります。

- `cmdb.unl`,
- `pcsoft-uninstall.unl`
- `pcsoftware.unl`

これらのファイルで、ICMタイプイベントなどの、構成アイテム (CI) にリンクされているイベントを更新できます。これらのファイルで、`model`および`modelvendor`ファイルに適用される検証ルールの更新も、標準ServiceCenterデータキットの整理と同様に行えます。



**警告:**

これらのファイルを、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、*Hewlett-Packard Development Company, L.P.*のサポートから入手できます。

## 入カイベントの同期をとる

ServiceCenter内の入カ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せん。

## アプリケーションのコンテキスト

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、いかなる環境においても、本番用にインストールされたServiceCenterアプリケーションには適用できません。このシナリオは、本マニュアルでの一例としてのみ提供、説明しています。

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、両者に含まれるデータはまったく異なります。

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。

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つの規則は、可能な限り適用する必要があります。

- 一意のリポジトリの定義。

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システムの記録の保存。

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間のデータの統合。

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い必要があります。

## データの照合更新とデータの複製の相違点

データの照合更新とデータの複製との間には、大きな相違点があります。

## 照合更新

照合更新とは、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に存在しているか?」

- 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、このプロセスは照合更新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。

この結果照合更新のプロセスは、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。

ServiceCenterコネクタとは異なり、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、照合更新のプロセスが（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**で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 **modification date** 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せん。

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\sc51ac43\scac.scn - scac\sc51ac43\acsc.scn`



このシナリオを実行する前に、`acsc.unl`ファイルをインポートする必要があります。このファイルが、ServiceCenterにあるデータをインポートする際に必要となるイベントを作成します。

2つのシナリオ（`scac.scn`および`acsc.scn`）は、インベントリデータと関連データ（連絡先、場所、モデルなど）をServiceCenter 5.1データベースとAssetCenter 4.3データベース間で複製します。

### 注意:

データベーススキーマが異なるため、双方向のデータ複製は推奨しません。各機能カテゴリの管理をどちらかのアプリケーションでのみ行い、各シナリオの対応するマッピングのみを有効にすることを推奨します。たとえば、従業員と場所をAssetCenterで更新し、インベントリデータをServiceCenterで更新する場合、`scac.scn`シナリオの従業員/場所の複製を無効にし、`acsc.scn`シナリオのインベントリデータの複製を無効にします。

## scac\sc51ac44\scac.scn - scac\sc51ac44\acsc.scn



このシナリオは、シナリオscac\sc51ac43\scac.scn - scac\sc51ac43\acsc.scn [ 献 422]と同一の機能を持ちます。

## scac\sc6ac43\scac.scn - scac\sc6ac43\acsc.scn



このシナリオは、シナリオscac\sc51ac43\scac.scn - scac\sc51ac43\acsc.scn [ 献 422]と同一の機能を持ちます。

この場合では、ServiceCenterバージョン6.0とAsset Managementバージョン4.3の間で複製が行われます。

## scac\sc6ac44\scac.scn - scac\sc6ac44\acsc.scn



このシナリオは、シナリオscac\sc51ac43\scac.scn - scac\sc51ac43\acsc.scn [ 献 422]と同一の機能を持ちます。

## scac\sc61ac44\



- acsc-ci.scn  
このシナリオで、AssetCenter構成アイテムのServiceCenterへの統合が可能になります。
- acsc-incident.scn  
AssetCenterで照合更新提案が行われた際、このシナリオを使用して、ServiceCenterのインシデントを作成します。
- scac-ci.scn  
このシナリオで、ServiceCenter構成アイテムのAssetCenterへの統合が可能になります。
- scac-wo.scn  
このシナリオを使用して、ServiceCenterの変更依頼に対応するAssetCenterの作業指示を作成します。

## scac\sc61ac50\



このフォルダにあるシナリオには、`scac\sc61ac44\` [ 献 423]フォルダのシナリオと同じ関数があります。

## scac\sc62ac50\



このフォルダにあるシナリオには、`scac\sc61ac44\` [ 献 423]フォルダのシナリオと同じ関数があります。

`citDeviceParent`要素を含むマッピングは、`citRelationship`要素でアップデートされています。

## 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\sca-sc\sca-sc.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。

## sc\sc51mail\scincident-mail.scn



インシデントチケット作成時にこのシナリオはEメールを送信します。`probsummary`テーブルにあるデータが読み込まれ、`operator`レコードに含まれているEメールアドレスを使用して、受信者にEメールが自動的に送信されます。Eメールには、インシデントの説明と、インシデントチケットにリンクされているファイルが添付ファイルとして含まれます。このシナリオは、午前8時から午後6時まで1時間ごとに実行するようにスケジュールされており、レコードの挿入や更新を行います。



## sc\sc60mail\scincident-mail.scn



このシナリオは、シナリオsc\sc51mail\scincident-mail.scn [ 献 424]と同一の機能を持ちます。

## sc\sc61mail\scincident-mail.scn



このシナリオは、シナリオsc\sc51mail\scincident-mail.scn [ 献 424]と同一の機能を持ちます。

## sc\sc62mail\scincident-mail.scn



このシナリオは、シナリオsc\sc51mail\scincident-mail.scn [ 献 424]と同一の機能を持ちます。

---

## ServiceCenter Webサービスシナリオ

本節では、ServiceCenter 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。

### scws\scws61\scws-im.scn



このシナリオは、インシデント管理レコードの作成や更新を行います。

#### 必須フィールド

生成ドキュメントタイプや取り込みドキュメントタイプのための一部のフィールドは必須です。このようなフィールドは、公開WSDLによって必須として定義されています。

作成モードのこのシナリオでは、以下の複合要素が必須です。

- createIncidentRequest
- model
- instance
- dkeys

■ アイコンは、必須であることを表します。

フィールドに値が入力されていない場合、警告がドキュメントログに書き込まれます。

 **注意:**

このシナリオは更新を実行しません。

### Strict error handling

Webサービスが失敗ステータスの応答を送信したとき、このオプションで、取り込みドキュメントのエラーを記録して生成を停止できます。

このオプションはデフォルトでは有効ではありません。

## scws\scws61\scws-cm.scn



このシナリオは、インシデント管理レコードの更新を行います。

このシナリオは2つのマッピングを実行します。最初のマッピングは挿入

(RetrieveComputer-CreateComputerDst) を実行し、2番目のマッピングは更新 (RetrieveComputer-UpdateComputerDst1) を実行します。最初に、シナリオは既存コンピュータのIDの取得を試み、見つからなければIDを作成します。次に更新を実行します。

RetrieveComputer-CreateComputerDstマッピングでは、挿入と更新の定義に以下のマッピングを使用します。

```
If [RetrieveComputerResponse.returnValue] = 0 Then
PifIgnoreDocumentMapping
End If
```

このマッピングでは、Webサービスが返す値がコンピュータIDが存在することを示す場合（値が0に等しい場合）、挿入は実行されません。Webサービスが返す値が1に等しい場合、これはコンピュータID値の戻しに失敗したことを意味するため、挿入が行われて空の値のコンピュータIDが作成されます。

RetrieveComputer-UpdateComputerDst1マッピングでは、挿入と更新の定義に以下のマッピングを使用します。

```
If [RetrieveComputerResponse.returnValue] <> 0 Then
PifIgnoreDocumentMapping
End If
```

このマッピングでは、Webサービスが返す値がコンピュータID値の戻しに失敗したことを示す場合、つまりコンピュータが存在しない場合、更新は実行されません。それ以外の場合、更新は実行されます。

## 必須フィールド

生成ドキュメントタイプや取り込みドキュメントタイプのための一部のフィールドは必須です。このようなフィールドは、公開WSDLによって必須として定義されています。これらのフィールドが必須であるかどうかの性質は、使用するモード（作成や更新）によって異なります。

フィールドに値が入力されていない場合、警告がドキュメントログに書き込まれます。

## Strict error handling

Webサービスが失敗ステータスの応答を送信したとき、このオプションで、取り込みドキュメントのエラーを記録して生成を停止できます。

---

## TS Censusシナリオ

本節では、TS Census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。

### tsc\tsc2ac36\tscac.scn



このシナリオで、TSCensus 2.xデータベースからAssetCenter 3.6データベースにデータを転送できます。

転送されるデータを以下に挙げます。

- コンピュータ
- ソフトウェア
- デバイス

### tsc\tsc2ac43\tscac.scn



このシナリオで、TSCensus 2.xデータベースからAssetCenter 4.3データベースにデータを転送できます。

### tsc\tsc3ac36\tscac.scn



このシナリオで、TSCensus 3.0データベースからAssetCenter 3.6データベースにデータを転送できます。

## tsc\tsc3ac43\tscac.scn



このシナリオで、TSCensus 3.0データベースからAssetCenter 4.3データベースにデータを転送できます。

## tsc\tsc3ac44\tscac.scn



このシナリオは、シナリオtsc\tsc3ac43\tscac.scn [ 献 428]と同一の機能を持ちます。

## tsc\tsc3ac50\tscac.scn



このシナリオは、シナリオtsc\tsc3ac43\tscac.scn [ 献 428]と同一の機能を持ちます。

## tsc\tsc3sc51\tscsc.scn



このシナリオを実行する前に、pcsoft-uninst.unlファイルとpcsoftware.unlファイルをServiceCenterにインポートする必要があります。これらのファイルは、Connect-Itのインストール（インストールフォルダのdatakit\scフォルダ）にあります。

このシナリオで、TSCensusが取得したデータを使用することで、ServiceCenter 5.1にあるコンピュータ（とそのソフトウェア）に関する情報を更新できます。

## tsc\tsc3sc6\tscsc.scn



このシナリオは、シナリオtsc\tsc3sc51\tscsc.scn [ 献 428]と同一の機能を持ちます。

## tsc\tsc3sc61\tscsc.scn



このシナリオは、シナリオtsc\tsc3sc51\tscsc.scn [ 献 428]と同一の機能を持ちます。

tsc\tsc3sc62\tscsc.scn



このシナリオは、シナリオtsc\tsc3sc51\tscsc.scn [ 献 428]と同一の機能を持ちます。

---

## Tivoli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。

tivoli\tim\tim4ac42\timac.scn



このシナリオで、Tivoli Inventory Management 4.0データベースからAssetCenter 4.2データベースにデータを転送できます。

転送されるデータを以下に挙げます。

- コンピュータ
- ソフトウェア

tivoli\tim\tim4ac43\timac.scn



このシナリオで、Tivoli Inventory Management 4.0データベースからAssetCenter 4.3データベースにデータを転送できます。

転送されるデータを以下に挙げます。

- コンピュータ
- ソフトウェア

tivoli\tim\tim4ac44\timac.scn



このシナリオは、シナリオtivoli\tim\tim4ac43\timac.scn [ 献 429]と同一の機能を持ちます。

tivoli\tim\tim4ac50\timac.scn



このシナリオは、シナリオtivoli\tim\tim4ac43\timac.scn [ 献 429]と同一の機能を持ちます。

## tivoli\tcm\tcm42sc51\tcmsc.scn



このシナリオを実行する前に、pcsoft-uninst.unlファイルとpcsoftware.unlファイルをServiceCenterにインポートする必要があります。これらのファイルは、Connect-Itのインストール（インストールフォルダのdatakit\scフォルダ）にあります。

このシナリオで、Tivoli Inventory Management 4.2データベースからAssetCenter 5.1データベースにデータを転送できます。

転送されるデータを以下に挙げます。

- コンピュータ
- ソフトウェア

## tivoli\tcm\tcm42sc60\tcmsc.scn



このシナリオは、シナリオtivoli\tcm\tcm42sc51\tcmsc.scn [ 献 430]と同一の機能を持ちます。

## tivoli\tcm\tcm42sc61\tcmsc.scn



このシナリオは、シナリオtivoli\tcm\tcm42sc51\tcmsc.scn [ 献 430]と同一の機能を持ちます。

## tivoli\tcm\tcm42sc62\tcmsc.scn



このシナリオは、シナリオtivoli\tcm\tcm42sc51\tcmsc.scn [ 献 430]と同一の機能を持ちます。

## tivoli\tcm\tcm42ac43\tcmac.scn



このシナリオで、Tivoli Configuration Manager 4.2データベースからAssetCenter 4.3データベースにデータを転送できます。

転送されるデータを以下に挙げます。

- コンピュータ
- ソフトウェア

## tivoli\tcm\tcm42ac44\tcmac.scn



このシナリオは、シナリオtivoli\tcm\tcm42ac43\tcmac.scn [ 献 430]と同一の機能を持ちます。

## tivoli\tcm\tcm42ac44\tcmac-swnorm.scn



このシナリオはAssetCenterソフトウェアライセンスモジュールのユーザ用であり、ソフトウェアモデルの照合更新を行えます。

詳細については、節「AssetManagementコネクタが生成するドキュメントタイプ [ 献 68]」を参照してください。

## tivoli\tcm\tcm42ac50\tcmac.scn



このシナリオは、シナリオtivoli\tcm\tcm42ac44\tcmac.scn [ 献 431]と同一の機能を持ちます。

## tivoli\tcm\tcm42ac50\tcmac-swnorm.scn



このシナリオは、シナリオtivoli\tcm\tcm42ac44\tcmac-swnorm.scn [ 献 431]と同一の機能を持ちます。

## tivoli\tcm\_sd\tcmsd42ac43

このフォルダに含まれているシナリオには、独立したマニュアルがあります。

## tivoli\tcm\_sd\tcmsd42ac44

このフォルダに含まれているシナリオには、独立したマニュアルがあります。

## tivoli\tcm\_sd\tcmsd421ac44

このフォルダに含まれているシナリオには、独立したマニュアルがあります。

## tivoli\tcm\_sd\tcmsd421ac50

このフォルダに含まれているシナリオには、独立したマニュアルがあります。

---

## Tivoli Enterprise Data Warehouseシナリオ

付属シナリオを以下に挙げます。

- Tivoli\tedw\_sc51tedw\sc\_cm3r\_tedw.scn
- Tivoli\tedw\_sc51tedw\sc\_incidents\_tedw.scn
- Tivoli\tedw\_sc51tedw\sc\_probsummary\_tedw.scn

これらのシナリオには、独立したマニュアルがあります。

---

## 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\amo3ac43\amoac.scn



このシナリオで、Unicenter AMO 3データベースからAsset Managementデータベースにデータを転送できます。

転送されるデータを以下に挙げます。

- コンピュータ
- ソフトウェア
- 関連付けられたデバイス

### amo\amo3ac44\amoac.scn



このシナリオは、シナリオamo\amo3ac43\amoac.scn [ 献 432]と同一の機能を持ちます。



## amo\amo4ac44\amoac.scn



このシナリオは、シナリオamo\amo3ac43\amoac.scn [ 献 432]と同一の機能を持ちます。

## amo\amo4ac50\amoac.scn



このシナリオは、シナリオamo\amo3ac43\amoac.scn [ 献 432]と同一の機能を持ちます。

---

## 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/wsac43/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。

---

## Winpark Actimaシナリオ

本節では、Winpark Actima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。

### wpk\wpkac43\wpkac.scn



このシナリオで、Winpark ActimaデータベースからAssetCenter 4.3データベースにデータを転送できます。

転送されるデータを以下に挙げます。

- 部署と従業員

- 会社
- 契約
- コストセンタ
- 場所
- コンピュータ
- デバイス
- モデル
- 電話
- ネットワーク情報とその他不特定の資産

### wpk\wpkac44\wpkac.scn



このシナリオは、シナリオwpk\wpkac43\wpkac.scn [ 献 433]と同一の機能を持ちます。

### wpk\wpkac50\wpkac.scn



このシナリオは、シナリオwpk\wpkac43\wpkac.scn [ 献 433]と同一の機能を持ちます。

# A AQLクエリ

本章では、AQLでクエリを書く方法について説明します。

## 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データベースにあるデータにアクセスするのに使うクエリ言語で、SQLと互換性があります。AQLをクエリに使用すると、データベースエンジンによりSQL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データベースに関するクエリを記述するには、SQLよりもAQLの方が適しています。以下に理由を挙げます。

## 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\_IDModelId**) と**完全名** (SQL名: **FullName**) にインデックスを強制して、モデルの完全名を検索する場合は、次のAQLを記述します。

```
SELECT FIRST_ROWS IModelId, FullName FROM amModel
```

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

```
SELECT /*+ FIRST_ROWS INDEX_ASC(M1 Model_IDModelId) */ M1.IModelId, M1.FullName FROM amModel M1
```

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

```
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。

クエリはAQL (Advanced Query Language) で記述します。AQLとは、AssetCenterデータベースにあるデータにアクセスするための、AssetCenterで使用する内部言語です。

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
- 最適なクエリの記述のためのAQLとAssetCenterデータベースに固有の特異性

「[AQLのシンタックス \[ 献 453\]](#)」と「[AQLの関数 \[ 献 465\]](#)」の節に、この節の補足説明があります。

### 警告:

AssetCenter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名を使用してAQLクエリを記述します。これらの名前の包括的なリストについては、データベース構造が記載されている*Database.txt*ファイルを参照してください。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、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: **ModelId**)
- モデルの完全名 (SQL名: **FullName**)
- ブランド (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 (**amModel**) テーブルとブランド (**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のインタフェースクエリエディタを使えば、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の階層リスト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、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, amEmplDept e, amLocation l WHERE a.lUserId *= e.lEmplDeptId AND e.lLocaId *= l.lLocaId
```

このコードは複数の外部結合を次々に使うので、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各テーブル内に主キーが("0")のレコードがあるので、外部SQL結合を呼び出す必要はありません。このため、AssetCenterは、通常の（外部ではない）結合を使用したSQLクエリを生成します。

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

上記のクエリでは、ユーザ（と場所のリンクが、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つのインスタンスにのみ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（値がNULL）の場合は、値は"0"に設定されます。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。"Location=0"または"llocald=0"などのようにします。

## 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、AssetCenterデータベースのカスタマイズを活用しつつ、クエリが簡単になります。

例：

部署と従業員のテーブル名の表記文字列が次のように構成されているとします。

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

AQLクエリは次のようになります。

```
SELECT self FROM amEmplDept
```

これは、次のクエリに相当します。

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

## CurrentUser

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

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

### 式として使う

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

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

### リンクとして使う

"CurrentUser"は、まず全テーブルを検索し、次に部署と従業員テーブル内にある現在のユーザのレコードに検索を絞るリンク、と見なされます。

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

例：データベースに接続しているユーザがアクションをトリガした場合、別のメッセージタイプのアクションが状況に応じて発生するように設定できます。メッセージタイプのアクションは、接続中のユーザに自動的に警告メッセージを

送信するとします。これを指定するには、アクションの詳細画面で次のように入力します。

## システムリストデータ

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

例：

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

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

**タイプ**（SQL名：seType）フィールドはシステムリストデータです。データベースに保存される値は次の通りです。

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

### 注意:

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、場所名の前に親場所名が付いた形式で保存されます。

```
FullName = '/東京/府中支社/府中ビル/5階/'
```

### "sLvl" フィールド

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

ルートはレベル0になります。



次のクエリは、「アジア地域営業部」のレコードとその付属部署を選択します。

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

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

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

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

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

## AQLの簡単な表記法

ここでは、AQLの記述を簡略化するための表記法を紹介します。

### 外部キー

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

例えば次のクエリ

```
WHERE location = 0
```

これは、次のクエリに相当します。

```
WHERE ILocaId = 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"location"リンクのSQL名で、"ILocaId"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`fu_`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。

- AssetCenterが、クエリ内で指定されているインデックスの使用を強制するモード。このようなインデックスが存在する場合、結果が検索で表示されます。
- AssetCenterが、クエリ内で指定されているインデックスを使用しないモード。この場合、データベースエンジンがデータのソート方法を決定します。



### 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 lModelId, 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"（ブランド）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"Brand"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。

### Configure list 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。手順を以下に挙げます。

- 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**Configure list**を選択します。
- 4 *Columns and sort*タブページで**Force indexes**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、クエリのファイナライズとプレビューを行えます。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 - compositionモード



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。

- *Tools/ Queries* 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。
  - *Tools/ Queries* 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、購入依頼の検証、税金式など、AssetCenterのクエリを呼び出す機能を使う。
- AssetCenterアプリケーションのExportコンポーネント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**Configure list**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。**Tools / Queries**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、**Configure list**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。

- **Columns and sort**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）。
- **Force indexes**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。
- **Filter (WHERE clause)**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、**Tools / Queries**メニューを選択します。表示されるウィンドウには、次の**Filter (WHERE clause)**と**Preview**タブページがあります。

- **Filter (WHERE clause)**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。
- **Preview**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、**Not shared**チェックボックスをオフにします。



注意:

管理者は、*Not shared*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**Create**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 : Filter (WHERE clause)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、フィールドに関連し、計算式を使用し、定数と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 *Field 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*Field 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 **Modify**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 **Modify**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、*Filter (WHERE clause)*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*Preview*タブページに移動します。

- 2  アイコンをクリックします。クエリ結果がレコードのリストの形式で表示されます。クエリで戻されたレコード数は、ウィンドウの右下に表示されます。

 **注意:**


Preview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

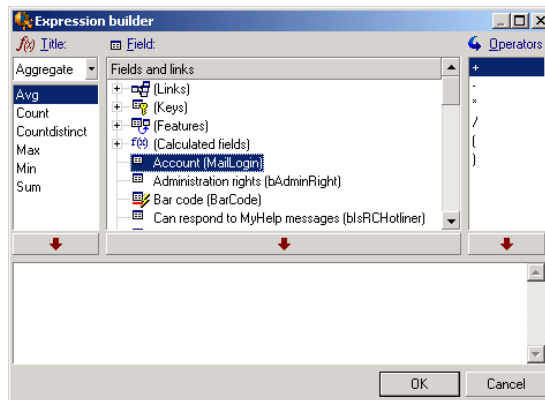
## 式を記述する

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Filter (WHERE clause)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。

- "Function"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。
- "Field"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。
- "Operators"列には、式で使用できる演算子のリストが表示されます。

"Function"、"Field"、"Operator"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*Filter (WHERE clause)*タブページに画面が切り替り、定義した式が表示されます。


## 定数

定数  $k$  は、選択基準に割り当てる固定の値です。例えば、メーカーが3Comのすべての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 [ 献 459]]  
[WHERE句 [ 献 460]]  
[GROUP BY句 [ 献 461]]  
[HAVING句 [ 献 462]]  
[ORDER BY句 [ 献 462]]

### サブクエリ

AQLでは、フィールドの代わりにサブクエリを使うことができます。

#### 注意:

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

(**SELECT** [DISTINCT] <式>  
[FROM句 [ 献 459]]  
[WHERE句 [ 献 460]]  
[GROUP BY句 [ 献 461]]  
[HAVING句 [ 献 462]]  
)

#### 警告:

サブクエリは括弧で囲む必要があります。

使用例：

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

## UNION型のクエリ

**UNION**を使うと、次のように複数のクエリの結果をグループ化できます。

**SELECT** <選択リスト>

[FROM句 [ 献 459]]

[WHERE句 [ 献 460]]

[GROUP BY句 [ 献 461]]

[HAVING句 [ 献 462]]

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

**SELECT** <選択リスト>

[FROM句 [ 献 459]]

[WHERE句 [ 献 460]]

[WHERE句 [ 献 460]]

[GROUP BY句 [ 献 461]]

[HAVING句 [ 献 462]]...

[ORDER BY句 [ 献 462]]

## 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**Table (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、 <b>AQL</b> 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、 <b>AQL</b> でリンクを処理する場合のみ使います。
LIKE	"="演算子と同様に機能し、ワイルドカード文字を使うこともできます。
NOT LIKE	次のワイルドカード文字を使用できます。 パーセント "%" :すべての文字列に置き換わります。 アンダースコア "_" :任意の1文字に置き換わります。 次の指定は、データベースエンジンによって変わります (SQL Anywhere、SQL Server、Sybaseはサポートしますが、WorkGroups用のOracleはサポートしません)。 [abc...]は任意の並び文字 (間にスペースがない) を定義します。 [a-c]は、文字列値の範囲を定義します。 DB2では、LIKE X演算子のXにSQL列名が含まれる場合は、この演算子を使うことはできません。この演算子で利用できるのは定数のみです。例えば、次のようなクエリはDB2で機能しません。 <b>SELECT COL1, COL2 FROM TABLE1 WHERE COL1 LIKE COL2</b>
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、出カクエリで出力する列名を指定する場合に特に便利です。

### 注意:

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

## FROM句

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

AQLでは、テーブル名のエイリアスを利用できます。

### シンタックス

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

## クエリの開始テーブル

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

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

たとえば、以下のサブクエリでは、**AQL**は`amAsset`テーブルの**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。

<b>SELECT</b> 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(lAstId) FROM amAsset GROUP BY Model.Brand
```

## HAVING句

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

### シンタックス

**HAVING** <検索条件>

### WHERE句との相違点

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

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

### 例

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

次の**WHERE**句を使ったクエリでも同じ結果が得られます。

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

**HAVING**句を使ったクエリの例

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

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

## ORDER BY句

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

項目の並べ替え順には、次のタイプがあります。

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

### シンタックス

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

## INSERT句

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

### シンタックス

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

この句はAssetCenter API AmDbExecAqlに含まれています。

AssetCenter APIの詳細については、マニュアル『プログラマーズリファレンス』の「関数の説明」の章を参照してください。

### 例

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

#### INSERT句を使用しないウィザードのコード

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

#### INSERT句を使用したウィザードのコード

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

## UPDATE句

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

### シンタックス

**UPDATE** <テーブル名> [テーブルのエイリアス] **SET** (<フィールド名> [, <フィールド名>...] [FROM句 [ 献 459]] [WHERE句 [ 献 460]]

### 例

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

#### UPDATE句を使用しないアクションのコード

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

#### UPDATE句を使用したアクションのコード

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

## DUPLICATE句

DUPLICATEは、データベースのテーブルのレコードを複製します。

この関数はAssetCenter特有です。

詳細については、マニュアル『はじめに』の「レコードの処理」の章の「レコードを複製する」を参照してください。

### シンタックス

**DUPLICATE** <テーブル名> [テーブルのエイリアス] **SET** (<フィールド名> [, <フィールド名>...] [FROM句 [ 献 459]] [WHERE句 [ 献 460]]

## DELETE句

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

### シンタックス

**DELETE** [FROM句 [ 献 459]] [WHERE句 [ 献 460]]



## 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. AQL - 集合型のAQL関数

関数	Description
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. AQL - 文字列型の関数

関数	Description
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. AQL - 日付型関数

関数	Description
Year( <日付> )	日付または日付+時刻型のフィールドの年を表す数字を返します (1997など)。
Month( <日付> )	日付または日付+時刻型のフィールドの月を表す数字を返します (1-12)。
Day( <日付> )	日付または日付+時刻型のフィールドで、その月の何日目かを表す数字を返します (1-31)。
DayOfYear( <日付> )	日付または日付+時刻型のフィールドで、その年の何日目かを表す数字を返します (1-366)。
WeekDay( <日付> )	日付または日付+時刻型のフィールドの曜日を表す数字を返します。 この数字は、サーバの設定によって変わります。例えば、SybaseまたはMicrosoft SQL Serverのデフォルトの設定は (1=日, 2=月, ..., 7=土)、Oracleのデフォルトの設定は (1=月, ..., 7=日) です。
Hour( <時間> )	時刻または日付+時刻型のフィールドの1日の時間を表す数字を返します (0-23)。
Minute( <時間> )	時刻または日付+時刻型のフィールドの分を表す数字を返します (0-59)。
Second( <時間> )	時刻または日付+時刻型のフィールドの秒を表す数字を返します (0-59)。
Getdate()	サーバの現在のシステム日付を返します。
AddDays( <日付>, <数値> )	日付または日付+時刻型のフィールドに特定の日数を加えます。

関数	Description
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. AQL - 日付型関数の例

Description	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()
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. AQL - 数値型の関数

関数	Description
Abs( <数値> )	「数値」の絶対値を返します。
Ceil( <数値> )	「数値」以上で一番小さい整数を返します。
Floor( <数値> )	「数値」以下で一番大きい整数を返します。
Mod( <a>, <b> )	"a" を "b" で割った余りを返します( $a = qb + r$ 、ただし $q$ は整数で $0 \leq r < q$ )。
Round( <a>, <n> )	"a" を小数点以下第" $n$ "位で四捨五入します。
Trunc( <a>, <n> )	"a" を小数点以下第" $n$ "位で切り捨てます。

適用例

Abs (2.516) = 2.

Ceil (2.516) = 3.

Floor (2.516) = 2.

Mod (6,4) = 2.

Round (31.16, 1) = 31.20.

Round (31.16, -1) = 30.00.

Round (31.16, -1) = 30.00.

Trunc (31.16, 1) = 31.1.

## テスト型のAQL関数

表 A.9. AQL - テスト型関数

関数	Description
IsNull( <a>, <b> )	"a"が"Null"の場合、"a"が"b"で置換されます。"a"と"b"のタイプは互換である必要があります。

# インデックス

- アーカイバル, 27
- アプリケーションコネクタ, 147
- インベントリコネクタ, 315-343
- キャッシュ
  - 設定, 36
- クエリ
  - AssetCenter, 437
  - クエリエディタ, 448
  - フィールドとリンク, 455
  - 定数, 456
- クエリエディタ, 448
- ゲートウェイ (コネクタ), 328
- コネクタ
  - Action Request System, 147
  - Altiris, 315
  - Asset Insight, 318
  - Asset Management, 59
  - BizDoc, 95
  - CA Unicenter AMO 3, 351
  - CA Unicenter AMO 4, 354
  - Desktop Administration 6.x, 71
  - Enterprise Discovery, 76
  - Eメール, 225
  - Hewlett-Packard, 59
  - iInventory, 332
  - JMS, 280
  - LANDesk for Inventory 8, 323
  - LANDesk Software Distribution, 326
  - LDAP, 239
  - Lotus Notes, 154
  - Management Portal, 143
  - Mercury, 189
  - MQSeries, 163
  - Name and describe the connector, 16
  - NTセキュリティ, 185
  - OpenView Desktop Inventory 7.x, 79
  - OpenView Inventory Manager, 137
  - OpenView Network Inventory, 89
  - OpenView Network Inventory 8.x, 84
  - OpenView Service Desk (受信), 133
  - OpenView Service Desk (送信), 130
  - OpenView Service Events, 139
  - OpenView Usage Manager, 141
  - SAP, 361
  - SAP ALE, 369
  - SAP BAPI, 363
  - SAP IDoc, 366
  - SCAutoリスニング, 99
  - ServiceCenter, 107
  - ServiceCenter Webサービス, 127

- SMS 2.x, 328
- SMS 2003, 328
- Tivoli CM for Inventory 4.2, 339
- Tivoli Enterprise Console , 194
- Tivoli Enterprise Console (送信) , 189
- Tivoli Enterprise Data Warehouse, 432
- Tivoli Inventory (バージョン4.0) , 336
- TS.Census 2, 344
- TS.Census 3, 348
- Webサービス, 284
- Winpark Actima, 357
- XML, 289
- XMLリスニング, 214
- インベントリ, 38
- コマンドライン, 252
- データベース, 205
- ルール (ディレクティブ) , 43
- 区切りテキストコネクタ, 256
- 取り込み用ルール, 46
- 生成用ルール, 44
- 接続タイプの定義, 18
- 設定, 15 , 15
- コミット, 35
- コレクション
  - LDAP, 250
  - XMLリスニングコネクタ, 219
  - マッピング, 68
- コレクションの照合更新, 68
- システムリストデータ, 443
- シナリオ
  - Action Request System, 374
  - Asset Insight, 375
  - Asset Management, 376
  - Decision Center, 409
  - Enterprise Discovery, 379
  - Eメール, 377
  - LANDesk, 386
  - LDAP, 398
  - Mercury, 395
  - MQSeries, 396
  - Network Node Manager, 399
  - NTセキュリティ, 403
  - OpenView Desktop Inventory, 406
  - OpenView Inventory Manager, 388
  - OpenView Network Discovery, 405
- SAP
  - BAPI, 412
  - IDOC, 409
  - ServiceCenter, 416
  - ServiceCenter Webサービス, 425
  - SMS, 414
  - Tivoli Inventory, 429
  - TS Census, 427
  - Unicenter AMO, 432
  - Webサービス, 433
  - Winpark Actima, 433
- スケジュールポイント, 33
  - 使用, 34
- セッション, 30
- ドキュメント
  - ルール (ディレクティブ) , 43
- ドキュメントタイプ
  - 取り込み用ルール, 47
  - 定義, 39
- トランザクション, 35
- プロキシ, 292
- プロキシサーバ, 292
- プロトコルコネクタ, 205
- マッピング
  - コレクション-コレクション間マッピング, 68
  - 取り込み用ルール, 47
- リンク
  - たどる (Asset Managementコネクタ) , 67
- ルール (ディレクティブ) , 43
  - WHERE句とORDER BY句, 45
  - 取り込み用ルールの定義, 47
  - 照合更新, 47
  - 生成用ルールの定義, 44
- 階層構造のテーブル, 444
- 外結合, 67
- 外部アプリケーション, 15
- 外部キー, 445
- 互換性
  - LDAPコネクタ, 239
  - NTセキュリティコネクタ, 186
  - XMLリスニングコネクタ, 215
  - コマンドラインコネクタ, 253
  - データベースコネクタ, 205

- 再接続, 29
  - 自動的, 29
- 最適化済みSQL, 436
- 主キー, 439
- 処理後のアクション, 87, 82
  - UrlFileInfo.Path, 295, 263
  - XMLコネクタ, 295
  - 区切りテキストコネクタ, 259
- 照合更新, 53, 47
  - Reconciliation scripts, 55
  - Reconciliation type, 48
  - キーセット, 52
  - コレクション, 53
  - 大文字小文字, 52
  - 動作, 53
- 照合更新キー, 48 (参考 ServiceCenterコネクタ)
  - キーセット, 52
- 詳細モード, 16
- 接続, 18
  - DB2, 24
  - MySQL, 23
  - ODBC, 19
  - Oracle, 20
  - Sybase, 21
  - 接続タイプ, 18
  - 設定, 19
- 接続テスト, 30
- 設定ウィザード, 15
  - Determine server delay, 31
  - Reconnection parameters, 29
  - 接続タイプの定義, 18
- 大文字小文字, 52
- 添付ファイル - Lotus Notes, 157
- 配列の取り出し, 46
- 付属シナリオ, 373-433
- 複製 (レプリケーション)
  - 矛盾, 53
- 論理演算子, 457

**A**

- AddOn, 62
- Advanced configuration, 27
- Advanced options, 37

- AQL, 435
  - ORDER BY, 446
  - インデックスの強制, 447
  - クエリ, 454
  - クエリエディタ, 448
  - シンタックス, 453
  - 簡略表記, 445
  - 結合, 438
  - 表記法, 453

## **B**

- BIOS, 255
- Blob
  - ServiceCenterコネクタ, 117
  - 生成用ルール, 46

## **C**

- CurrentUser, 442

## **D**

- DELETE, 464
- DSC, 272
- DSE, 243
- DTD
  - BizDocコネクタ, 97
  - Eメールコネクタ, 230
  - MQSeriesコネクタ, 167
  - OpenView Desktop Inventoryコネクタ, 83
  - OpenView Network Inventoryコネクタ, 87
  - XMLコネクタ, 295
- DUPLICATE, 464

## **E**

- ERPコネクタ

## **F**

- FROM, 459
- FullName, 444

## **G**

- getdate(), 67

GROUP BY, 461

## H

HAVING, 462

Hewlett-Packardコネクタ

HP Softwareコネクタ, 59-76

## I

INSERT, 463

## J

Java, 40

Java仮想マシン, 40

JVM, 40

## L

Learning mode

MQSeriesコネクタ, 171

XMLコネクタ, 299

## M

MAPI, 236

## N

NNM, 399

NULL, 441

## O

ORDER BY, 462, 45

表記文字列, 445

OVO Windows, 93

## R

Reconciliation scripts, 55

## S

scdb.cfg, 115

SELECT, 454

AssetManagementコネクタ, 64

表記文字列, 445

Self, 441

send2cit.sh, 196

Server

Determine server delay, 31

sLvl, 444

SQL92, 27

Substring, 436

Sybase

インデックス, 448

sysdate, 67

## T

TEDW, 432

Tivoli CM for Software Distribution 4.2, 339

Tivoli CM for Software Distribution Status  
4.2, 344

## U

UNION, 455

UNIX

Asset Managementコネクタ, 62

Eメールコネクタ - 処理のオプション, 227

ServiceCenterコネクタ - 使用方法, 109

XMLコネクタ - end-of-lineマーカーの変換  
ルール, 313

コマンドラインコネクタ, 253

UPDATE, 464

UrlFileInfo

構造体 - BizDocコネクタ, 99

構造体 - XMLコネクタ, 314

構造体 - 区切りテキストコネクタ, 278

UrlFileInfo.Path

XMLコネクタ, 295

区切りテキストコネクタ, 263

## W

WHERE, 460, 45

Lotus Notesコネクタ, 158

## X

XSD

BizDocコネクタ, 97

MQSeriesコネクタ, 167

OpenView Desktop Inventoryコネクタ,  
83

OpenView Network Inventoryコネクタ,  
87



XMLコネクタ, 295

