



Peregrine | Connect-It
ユーザガイド

© Copyright 2002 Peregrine Systems, Inc.

All rights reserved.

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

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

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

本書で説明されているソフトウェアは、ライセンス契約または非開示契約に基づいて提供されます。これらの契約の条項に従う場合に限り、本ソフトウェアを使用またはコピーすることができます。本書に記載されている事項が予告なく変更されることがありますが、Peregrine Systems, Incは予告の義務を負いません。本書の最終バージョンの日付を確認するには、Peregrine Systems, Inc.のカスタマサポートまでお問合せください。

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

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

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

本書の内容は、ライセンス契約に基づくプログラムのバージョン3.1.0に適用されます。

Connect-It

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



目次

はじめに	27
Connect-Itの使用目的	27
Connect-Itの対象ユーザ	28
本書の使用方法	28
1. 概要	33
Connect-Itでのデータ処理	33
2. インストール	35
必要な動作環境	35
最初のインストール	36
Windows 32ビット上のインストール	36
UNIX上のインストール	38
ベースコネクタ	39
オプションコネクタ	39
データキット	40
追加コネクタ	42
認証証明書	42
認証証明書の入力	42
ファイルの構成	43
関連したファイル	44

3. シナリオビルダ	47
メインウィンドウ	47
シナリオ図のビュー	50
シナリオ図のビューを使用する	50
シナリオ図のビューに関する規則	51
メニュー	51
シナリオビルダのメニューバー	52
シナリオビルダのメインウィンドウ内のショートカットメニュー	57
ログ	60
Connect-Itログ	60
ドキュメントログ	61
オプション	73
アクセスビリティ	74
表示	74
確認	77
コネクタ	78
文書	80
ログ	80
4. 統合シナリオのインプリメンテーション	81
シナリオインプリメンテーションウィザードを使用する	81
シナリオの手動インプリメンテーション	82
シナリオ図内にコンポーネントを配置する	82
コンポーネントをリンクする	83
コンポーネントを削除する	85
生成用または取り込み用ドキュメントタイプ	86
生成用または取り込み用ドキュメントタイプを作成する	89
生成用または取り込み用ドキュメントタイプのDTDをコピーする	90
生成用ドキュメントタイプのデータを参照する	90
ドキュメントタイプ編集用ウィンドウでのショートカットメニュー	92
5. ドキュメントタイプのマッピング	95
マッピングボックス	96
マッピングボックスの設定	99
マッピングの編集	100
マッピングの作成方法	100
既存マッピングの編集方法	102
マッピングの削除方法	103
マッピング説明の記述方法	103
複数のマッピングの並べ替え	103
ソースとターゲットのデータをプレビューする	104
マッピングを非アクティブにする	105

マッピングをコピー/貼り付けする	105
ターゲット要素をソース要素にマップする	106
フィールドからフィールドへの直接マッピング	106
コレクションからコレクションへのマッピング	107
コレクションからドキュメントへのマッピング	109
フィールドからコレクションへのマッピング	110
ドキュメントタイプ要素のパス	112
ドキュメント要素のパス	113
Blobタイプのフィールドのマッピング	114
スクリプトを使用したフィールドからフィールドへのマッピング	115
スクリプトの書き方	116
ショートカット (コンテキスト) メニュー	119
スクリプトの作成	122
エディタの使用	123
文字列テーブル	125
マップテーブル	126
ダイナミックマップテーブル	127
グローバル関数と変数	128
ユーザフォーマット	128
PIF関数	130
テキストエディタ	130
シナリオと、シナリオに関連するファイルの移動	134
オンラインのプログラム用参考ガイド	134
スクリプトの認証	134
6. ピボットドキュメントタイプ	137
ピボットドキュメントタイプの機能	138
自動的に作成されるマッピング	138
ピボットドキュメントタイプを使用して統合シナリオを作成する	139
内部コネクタを表示する	139
[マッピング] タブと [ピボットマッピング] タブ	140
ピボットドキュメントタイプのリスト	140
「使用可能なドキュメントタイプ」と「ピボットドキュメントタイプ」間 のマッピングを変更する	141
ターゲットコネクタを変更する	142
7. 統合シナリオのテストとデバッグ	143
生成用ドキュメントタイプのテスト	143
ドキュメントログの使用	144
キャッシュファイルを使用する	144
キャッシュファイルの内容を消去する	146
キャッシュファイルの内容と外部アプリケーション内の内容の同期をと る	146

オフラインで作業する	147
1コネクタの始動時に現在のシナリオの全コネクタを開く	147
テストモード	148
一時停止を実行する	148
マッピング定義に関するアドバイス	148
8. シナリオ文書	149
シナリオ文書の内容	149
シナリオ文書をHTMLフォーマットで表示する	152
シナリオ文書をHTMLフォーマットで表示する	152
シナリオ文書のプロパティ	152
シナリオ文書のプロパティを編集する	153
シナリオ文書の作成	155
シナリオ文書の作成	155
DBKフォーマットの文書	155
9. 統合シナリオをプロダクションモードにする	157
スケジュールの作成	157
スケジューラの編集	158
スケジュールの編集	163
Connect-Itサービスの作成	168
Connect-Itコンソール	168
WindowsでConnect-Itサービスを作成する	170
UNIXでConnect-Itサービスを作成する	171
Connect-Itサービスの起動または停止	171
Connect-Itサービスの削除	172
コマンドライン	172
シナリオのトラッキングをConnect-Itのコンソールで管理する	173
コネクタの設定の変更	173
スケジューラの編集	173
Connect-Itログの参照	173
サービスのログファイルの参照	174
シナリオの性能を最適化する	174
実行結果を使ってドキュメント処理の時間を評価する	174
ドキュメントの生成を改善する	176
ドキュメントの取り込みを改善する	182
Connect-Itエンジンを使ってドキュメント処理を改善する	183
Asset Managementコネクタを使用するシナリオの性能を向上させる	185
データベース型コネクタを使用するシナリオの性能を向上させる	188
10. 処理レポート	189
処理レポートの内容	190

既製シナリオ内で処理レポートを使用する	191
InfraTools Desktop Discovery - Asset Managementシナリオでの処理レポートの使用	191
11. Connect-ItのJava開発キット (JDK)	195
Java開発キットの内容	195
JCA規格の実装	196
CCIクラスの実装	196
SPIクラスの実装	200
JCAの拡張：イベントクラス	202
UDC (Unified Document Content) 規格	204
UDC規格の実装	206
Javaコネクタの作成	206
前提条件	207
接続の契約を定義する	207
トランザクションの契約を定義する	208
セキュリティの契約を定義する	209
データを定義する	210
データへの相互作用を定義する	210
コネクタの自動記述	212
Java仮想マシン (JVM) を設定する	223
Javaコネクタの導入	223
イベントコネクタを作成する	225
トランザクションの契約を定義する - イベントコネクタ	225
データを定義する - イベントコネクタ	226
データへの相互作用を定義する - イベントコネクタ	226
自動記述 - イベントコネクタ	226
Javaコネクタの導入 - イベントコネクタ	227
JCAコネクタの作成 - 例	227
接続の契約を定義する - 用例コネクタ	228
トランザクションの契約を定義する - 用例コネクタ	229
セキュリティの契約を定義する - 用例コネクタ	229
データを定義する - 用例コネクタ	229
データへの相互作用を定義する - 用例コネクタ	230
設定ウィザードを改善する	230
DSDファイルの作成	231
DSDファイルの内容	231
コネクタ設定のプロパティの定義	232
DSD - 頻発する問題点	236
12. コネクタの設定	237
コネクタを設定する	238
コネクタに名前を付け、役割を説明する	238

[名前]	239
[説明]	239
接続タイプを選択する	240
接続を選択する	241
接続を設定する (ODBC接続)	241
接続を選択する (Oracleネイティブ接続)	243
接続を選択する - Sybaseネイティブ接続	244
接続を選択する - ネイティブMySQL	246
高度な設定	247
データベース記述ファイル	247
SQL92をサポート	248
詳細オプション	248
再接続のパラメータを設定する	250
接続が失われた場合自動接続する	251
接続テストのクエリ	251
サーバとの時間差を指定する	252
[サーバとの時間差を自動的に計算する] オプションが選択されている	253
[サーバとの時間差を自動的に計算する] オプションが選択されていない	253
タイムゾーンの使用	253
スケジュールのポイントを設定する	255
スケジュールのポイントを使用する	255
現在の日付を上限として使用する	256
トランザクションを設定する	257
キャッシュを設定する	258
ピボットドキュメントタイプの使用	259
13. コネクタのルール (ディレクティブ)	261
生成用ルール	262
生成用ルールを定義する	262
WHERE句とORDER BY句	263
取り込み用ルール	264
コネクタの取り込み用ルールを定義する	264
整合性	265
14. Peregrine Systemsコネクタ	275
Action Request Systemコネクタ	275
Action Request Systemコネクタの互換性	275
Action Request Systemコネクタの既知の制約点	276
Action Request Systemコネクタの設定	277
生成用ルール - Action Request Systemコネクタ	280
Action Request Systemコネクタの取り込み用ルール	281

追加情報 - Action Request Systemコネクタ	281
用例シナリオ - Action Request Systemコネクタ	281
Asset Managementコネクタ	282
Asset Managementコネクタの互換性	282
Asset Managementコネクタの制約点	284
Asset Managementコネクタの設定	284
Asset Managementコネクタの生成用ルール	286
Asset Managementコネクタの取り込み用ルール	287
用例シナリオ - Asset Managementコネクタ	287
InfraTools Desktop Discoveryコネクタ	289
InfraTools Desktop Discoveryコネクタの互換性	289
InfraTools Desktop Discoveryコネクタの設定	291
InfraTools Desktop Discoveryコネクタのルール	299
追加情報	299
用例シナリオ - Infratools Desktop Discoveryコネクタ	299
InfraTools Managementコネクタ	300
InfraTools Managementコネクタの互換性	300
InfraTools Managementコネクタの設定	300
InfraTools Managementコネクタの生成用ルール	303
InfraTools Managementコネクタの取り込み用ルール	304
用例シナリオ - InfraTools Managementコネクタ	304
InfraTools Network Discoveryコネクタ	304
InfraTools Network Discoveryコネクタの互換性	304
InfraTools Network Discoveryコネクタの設定	305
InfraTools Network Discoveryコネクタの生成用ルール	306
InfraTools Network Discoveryコネクタの使用可能なドキュメントタイプ	308
追加情報 - InfraTools Network Discoveryコネクタ	308
用例シナリオ - InfraTools Network Discoveryコネクタ	309
Peregrine Desktop Inventory (PDI) コネクタ	310
PDIコネクタの設定	310
生成用ドキュメントタイプ	314
用例シナリオ - PDIコネクタ	314
ゲートウェイ3.xコネクタ	314
ゲートウェイコネクタのODBC互換性	315
ゲートウェイコネクタの既知の制約点	316
Peregrine Systems3.xゲートウェイにサポートされているアプリケーションのバージョン	316
ゲートウェイコネクタの設定	319
ゲートウェイ3.xコネクタの生成用ルール	320
ゲートウェイ3.xコネクタの使用可能なドキュメントタイプ	321
用例シナリオ - ゲートウェイ3.xコネクタ	321
ServiceCenterコネクタ	321
ServiceCenterコネクタの互換性	321

ServiceCenterコネクタの設定	323
ServiceCenterコネクタの生成用ルール	325
追加情報	326
用例シナリオ - ServiceCenterコネクタ	328
15. アプリケーションコネクタ	329
Lotus Notesコネクタ	329
Lotus Notesコネクタの互換性	329
Lotus Notesコネクタの既知の制約点	330
Lotus Notesコネクタの設定	330
Lotus Notesコネクタの生成用ルール	332
Lotus Notesコネクタの取り込み用ルール	333
Lotus Notesコネクタに関する追加情報	335
用例シナリオ - Lotus Notesコネクタ	337
MQSeriesコネクタ	337
互換性	338
MQSeriesコネクタ（読み取り）の設定	338
MQSeriesコネクタ（書き込み）の設定	347
MQSeriesコネクタの生成用ルール	353
MQSeriesコネクタの取り込み用ルール	355
用例シナリオ - MQSeriesコネクタ	357
NTセキュリティコネクタ	358
NTセキュリティコネクタの互換性	358
NTセキュリティコネクタの設定	358
NTセキュリティコネクタの使用可能なドキュメントタイプ	359
NTセキュリティコネクタの生成用ルール	360
用例シナリオ - NTセキュリティコネクタ	361
16. プロトコルコネクタ	363
データベースコネクタ	363
データベースコネクタの互換性	364
データベースコネクタの既知の制約点	364
データベースコネクタの設定	367
データベースコネクタの生成用ルール	369
データベースコネクタの取り込み用ルール	369
追加情報	370
用例シナリオ - データベースコネクタ	370
Eメールコネクタ	370
Eメールコネクタの互換性	371
Eメールコネクタの既知の制約点	371
Eメール（受信）コネクタの設定	372
Eメール（送信）コネクタの設定	375
Eメール（受信）コネクタの生成用ドキュメントタイプ	377

Eメール（送信）コネクタの取り込み用ドキュメントタイプ	380
添付ファイルの処理	381
用例シナリオ - Eメールコネクタ	383
LDAPコネクタ	383
LDAPコネクタの互換性	383
LDAPコネクタ使用上の注意	384
LDAPコネクタの設定	384
取り込み用ルール	387
LDAPコネクタの生成用ルール	389
追加情報	393
用例シナリオ - LDAPコネクタ	396
コマンドラインコネクタ	396
コマンドラインコネクタの互換性	396
コマンドラインコネクタの設定	396
使用可能なドキュメントタイプ	398
取り込み用ルール（ディレクティブ）	399
用例シナリオ - コマンドラインコネクタ	399
テキストコネクタ	399
テキストコネクタの既知の制約点	400
テキストコネクタの設定（読み取り）	401
テキストコネクタの設定（書き込み）	407
書式設定ファイル（DSCファイル）の作成	413
テキストコネクタ（書き込み）の取り込み用ルール	419
用例シナリオ - テキストコネクタ	420
XMLコネクタ	420
XMLコネクタの既知の制約点	421
XMLコネクタの設定（読み取り）	421
XMLコネクタの設定（書き込み）	432
XMLコネクタ（書き込み）の取り込み用ルール - FTPとファイル	439
追加情報 - XMLコネクタ	440
用例シナリオ - XMLコネクタ	442
17. インベントリコネクタ	445
Intel LANDeskコネクタ	445
Intel LANDeskコネクタの互換性	446
既知の制約点	446
Intel LANDeskコネクタの設定	446
Intel LANDeskコネクタの生成用ルール	448
使用可能なドキュメントタイプ	448
用例シナリオ - PDA Intel LANDeskコネクタ	449
Tivoli Inventoryコネクタ（バージョン3.1と3.6）	449
Tivoli Inventory Managementコネクタの互換性	450
Tivoli Inventory Managementコネクタの既知の制約点	450

Tivoli Inventory Managementコネクタの設定	451
使用可能なドキュメントタイプ	453
Tivoli Inventory Managementコネクタの生成用ルール	454
用例シナリオ - Tivoli Inventory Managementコネクタ (バージョン3.1と3.6)	455
Tivoli Inventoryコネクタ (バージョン4.0)	455
Tivoli Inventory Management 4.0コネクタの互換性	456
既知の制約点	456
Tivoli Inventory Management 4.0コネクタの設定	457
Tivoli Inventory Management 4.0コネクタの生成用ルール	458
Tivoli Inventory Management 4.0コネクタの取り込み用ルール	459
用例シナリオ - Tivoli Inventoryコネクタ (バージョン4.0)	459
SMS 1.xコネクタとSMS 2.xコネクタ	459
SMSコネクタの互換性	459
コネクタ使用上の重要な情報	460
SMSコネクタの設定	461
SMSコネクタの生成用ルール	462
用例シナリオ - SMS 1.xとSMS 2.xコネクタ	463
Unicenter AMOコネクタ	464
Unicenter AMOコネクタの互換性	464
Unicenter AMOコネクタの既知の制約点	464
Unicenter AMOコネクタの設定	465
Unicenter AMOコネクタの生成用ルール	466
使用可能なドキュメントタイプ	466
用例シナリオ - Unicenter AMOコネクタ	467
18. Javaコネクタ	469
SCAutoリスニングコネクタ	469
イベント受信モードのSCAutoリスニングコネクタ	470
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	470
SCAutoリスニングコネクタの制約点	470
SCAutoリスニングコネクタの設定	471
追加情報	472
用例シナリオ - SCAutoリスニングコネクタ	474
XMLリスニングコネクタ	474
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)	475
相互作用モードのXMLリスニングコネクタ	475
XMLリスニングコネクタの互換性	475
XMLリスニングコネクタの制約点	475
XMLリスニングコネクタの設定	476
XMLリスニングコネクタの生成用ルール	477
XMLリスニングコネクタの使用可能なドキュメントタイプ	477
用例シナリオ - XMLリスニングコネクタ	477

Tivoli Enterprise Consoleコネクタ (送信)	477
前提条件	478
TEC adapter (送信) コネクタの設定	478
追加情報	481
付属の用例シナリオ	481
Tivoli Enterprise Consoleコネクタ (受信)	481
TECコネクタ (受信) の互換性	482
イベント送信用のTECの設定	482
TECコネクタ (受信) の設定	482
追加情報	483
付属の用例シナリオ	487
Webサービスコネクタ	487
Webサービスコネクタの制約点	488
Webサービスコネクタの設定	489
Webサービスコネクタの取り込み用ルール	490
使用可能なドキュメントタイプ	490
用例シナリオ - Webサービスコネクタ	491
JDBCコネクタ	492
JDBCコネクタの互換性	492
JDBCコネクタの設定	492
JDBCコネクタの生成用ルール	494
JDBCコネクタの取り込み用ルール	494
用例シナリオ - JDBCコネクタ	494
19. 用例シナリオ	495
Unicenter AMOシナリオ	495
amo /amo2am.scn	495
InfraTools Desktop Discoveryシナリオ	496
idd /iddac36 /iddac.scn	496
idd /iddac40 /iddac.scn	496
idd /iddac41 /iddac.scn	496
idd /idditsm /iddam.scn	496
idd /idditsm /iddhd.scn	496
idd /idds /iddsc.scn	497
rsm /idd2rsm.scn	497
Intel LANDesk シナリオ	497
ild /ild2am.scn	497
ild /ild65am.scn	497
InfraTools Managementシナリオ	498
im /htsec2im.scn	498
InfraTools Network Discoveryシナリオ	498
ind /indac36 /indac.scn	498
ind /indac40 /indac.scn	499

ind /ndac41 /ndac.scn	499
rsm /nd2rsm.scn	499
Action Request Systemシナリオ	499
itsm-am /cmppo.scn	500
itsm-am /sharedat.scn	500
LDAPシナリオ	500
ldap /ac /complete.scn	501
ldap /ac /simple.scn	501
ldap /hotes /hames.scn	501
Eメールシナリオ	501
ac36 /finreque.scn	502
ac36 /finconfi.scn	502
ac36 /hewemplo.scn	502
ac40 /finconfi.scn	502
ac40 /finreque.scn	503
ac40 /hewemplo.scn	503
MQSeriesシナリオ	503
mqseries /acmq /mqtoprgn.scn	503
mqseries /acmq /prgntomq.scn	504
mqseries /scmq /mqsc.scn	504
NTセキュリティシナリオ	505
ntac36 /addcpu.scn	505
ntac36 /adduser.scn	505
ntac40 /addcpu.scn	505
ntac40 /adduser.scn	505
ntac41 /addcpu.scn	506
ntac41 /adduser.scn	506
ServiceCenterシナリオ	506
scac /sc3ac36 /scac.scn	506
適用コンテキスト	508
データの整合性チェックとデータの複製の相違点	510
Connect-It内でのデータの複製の管理	511
scac /sc3ac40 /scac.scn	513
scac /sc3ac41 /scac.scn	513
scac /sc4ac40 /scac.scn	513
scac /sc4ac41 /scac.scn	513
scauto /scacfg /scacfg.scn	513
acscups /dataprop /01-SCcompany2ACamCompany.scn	514
acscups /dataprop /02-SClocation2ACamLocation.scn	514
acscups /dataprop /03-SCdept2ACamEmplDept.scn	514
acscups /dataprop /04-SCcontacts2ACamEmplDept.scn	514
acscups /dataprop /05-SCdevtype2ACamNature.scn	515
acscups /dataprop /06-SCcocmlcat2ACamNature.scn	515
acscups /dataprop /07-SCmodel2ACamModel.scn	515

acscups /dataprop /08-SCdevtype2ACamModel.scn	515
acscups /dataprop /09-SCdevice2AACamPortfolioAndamComputer.scn	515
acscups /dataprop /10-SClaptop2ACamPortfolioAndamComputer.scn	516
acscups /dataprop /11-SCworkstation2ACamPortfolioAndamComputer.scn	516
acscups /dataprop /12-SCpc2ACamAndamComputer.scn	516
acscups /dataprop /13-SCserver2ACamAndamComputer.scn	516
acscups /dataprop /14-SCphone2ACamPortfolioAndamPhone.scn	516
SMSシナリオ	517
sms /sms1am.scn	517
sms /sms1amlt.scn	517
sms /sms2am.scn	517
sms /sms2amlt.scn	518
sms /sms2ac41.scn	518
Tivoli Inventory Managementシナリオ	518
tim /tim2am	518
tim /tim4am	519
tim /tim4sc4	519
ゲートウェイ3.xシナリオ	519
uagw /uagw.scn	519
Webサービスシナリオ	520
ws/ac41/currency.scn	520
Peregrine Desktop Inventoryシナリオ	520
pdi /pdiac41 /pdiac.scn	520
A. AQL言語	521
クエリの例	521
メインテーブルのフィールドを特定の値と比較する	522
メインテーブルのリンクを別のリンクと比較する	522
メインテーブルのリンクを特定の値と比較する	522
メインテーブルにリンクしたテーブルのフィールドの値に応じて比較する	522
階層構造のテーブル	522
2つの条件を組み合わせたクエリ	523
数値、日付、テキスト型フィールドの比較	523
任意管理項目に関するクエリ	523
式でレコードを検索する	523
未入力のフィールドを検索する	524
リンクなしのレコードを検索する	524
エイリアスを使ったクエリ	525
AQLの関数	525
集合型のAQL関数	526
文字列型のAQL関数	527
日付型のAQL関数	527

数値型のAQL関数	529
テスト型のAQL関数	530
AQLのシンタックス	530
表記法	531
クエリのシンタックス	531
FROM句：クエリの対象となるテーブル	533
クエリの要素	534
WHERE句	539
GROUP BY句	539
HAVING句	540
ORDER BY句	541
B. 問題点の報告方法	543
一般情報	543
機能に関する情報	543
技術的な情報	544
問題固有の情報	545
その他の情報	545
Connect-Itに関する問題を報告する場合	546
C. 用語解説	547
Connect-It用語	547
コネクタ	547
ドキュメントとドキュメントタイプ	549
マッピング	554
統合シナリオ	554
ルール（ディレクティブ）	555
ドキュメントログ	555
スケジュール	555
処理レポート	556
認証証明書	556
外部アプリケーション	556
Blob	556
キャッシュ	556
コレクション	557
コンポーネント	557
サービスコンソール	557
取り込み	557
DAT	557
DBK	558
ルール（ディレクティブ）	558
コネクタの導入	558
シナリオビルダ	558

要素	558
ユーザフォーマット	559
IDX	559
MSG	559
スケジュールのポインタ	559
生成	560
スケジューラ	560
実行結果	560
構造体	560
処理後のアクション	561
ピボットドキュメントタイプ	561
ログビューア	561
ビュー	561
DSC	562
主要用語	562
AQL	562
DSE	562
FSF	562
JCA	563
整合性チェック	563
XMLスキーマ	563
マップテーブル	563
トランザクション	564
グローバル変数	564
XSL	564
索引	565

図の一覧表

1.1. Connect-Itで、あるデータベースから別のデータベースヘデータを転送する	33
3.1. シナリオビルダ - メインウィンドウ	48
3.2. Connect-Itログ	60
3.3. シナリオビルダ - ドキュメントログのタブ	62
3.4. ドキュメントログの設定ウィンドウ	62
3.5. 処理中に起こった問題	65
3.6. ドキュメントログ内のトラッキング項目へのフィルタ	66
3.7. トラッキング項目のフィルタ	66
3.8. ドキュメント内の記述文字列	67
3.9. ドキュメントタイプ詳細内のコレクションと、ドキュメント詳細内のコレクション	68
3.10. ServiceCenterコネクタに取り込まれた「pcsoftware」ドキュメントの詳細部分	69
3.11. ノード下にトラッキング項目があることを意味する灰色のトラッキング項目	72
3.12. ドキュメント詳細内のトラッキング項目用フィルタ	72
4.1. シナリオビルダのツールボックス	83
4.2. シナリオのコンポーネントのリンク	84
4.3. リンクの作成またはコネクタの移動	85
4.4. ServiceCenterコネクタの取り込み用ドキュメントタイプの編集	87
4.5. 使用可能なドキュメントタイプのゾーン	88

4.6. 生成用ドキュメントタイプのデータプレビュー用ウィンドウ	91
5.1. マッピング-コンポーネントとソースドキュメントタイプの選択用ウィンドウ	96
5.2. マッピング編集用ウィンドウ	98
5.3. 作業枠上のマッピングスクリプトの要約	99
5.4. コンポーネントとソースドキュメントタイプの選択用ボックス	101
5.5. マッピングスクリプト内のソース要素	107
5.6. コレクション	108
5.7. コレクションからドキュメントへのマッピング	110
5.8. フィールドからコレクションへのマッピング	111
5.9. 複数のフィールドをドラッグ&ドロップで移動させる方法	117
5.10. マップテーブルのエディタ	124
5.11. テキストエディタの設定	133
6.1. ピボットドキュメントタイプの使用	138
6.2. 内部コネクタの表示	140
7.1. コネクタ-キャッシュ使用を示すアイコン	145
7.2. コネクタ-オフラインセッションを示すアイコン	147
9.1. スケジューラエディタ	159
9.2. スケジュールの編集ウィンドウ	163
9.3. マッピングの順番	164
9.4. Connect-Itコンソール	169
9.5. ドキュメントの処理 - 進行状況バー	178
10.1. InfraTools Desktop Discovery - Asset Managementシナリオ図	191
11.1. Java開発キット - CCIクラス	198
11.2. Java開発キット - SPIクラス	201
11.3. Java開発キット - イベントクラス	203
13.1. コネクタ - 生成用ルールと取り込み用ルール	262
13.2. 取り込み用ルール - [整合性] タブ	266
13.3. 取り込み用ルール - [高度な整合性チェック] タブ	271
13.4. 整合性チェック - コレクションのフィルタ	273
13.5. 整合性チェック - ターゲットコレクションの構成要素の削除	273
14.1. PDIコネクタ - 生成用ドキュメントタイプ	314
15.1. MQSeriesコネクタ - 相関IDによるMQSeriesメッセージの識別	357
15.2. NTセキュリティコネクタ - 使用可能なドキュメントタイプ	360
16.1. 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	380
16.2. 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	380
16.3. テキストコネクタ - テキストファイル内のデータの記録法	400
16.4. テキストコネクタ - データ処理	414
16.5. テキストコネクタ - 取り込み用ルール	420
16.6. XMLコネクタ - の取り込み用ルール	440

17.1. Intel LANDeskコネクタ - COMPUTER_SYSTEMドキュメントタイプ	449
17.2. Tivoli Inventory Managementコネクタ - COMPUTER_SYSTEMド キュメントタイプ	453
17.3. Tivoli Inventory Managementコネクタ - 生成用ルール	454
17.4. SMSコネクタ - 生成用ルールの例	463
18.1. Webサービスコネクタ - 機能の概要	488
C.1. Asset Managementコネクタ - 使用可能なドキュメントタイプ群	550
C.2. ドキュメントタイプの要素	550
C.3. ドキュメントタイプのコレクション	552
C.4. ドキュメントタイプのツリー構造	553

表の一覧表

3.1. シナリオビルダ - ツールバーのアイコン	48
3.2. [ファイル]メニューのコマンドの機能	52
3.3. [編集]メニューのコマンドの機能	53
3.4. [表示]メニューのコマンドの機能	53
3.5. [シナリオ]メニューのコマンドの機能	54
3.6. [ツール]メニューのコマンドの機能	55
3.7. [ログ]メニューのコマンドの機能	56
3.8. [管理]メニューのコマンドの機能	56
3.9. [Java]メニューのコマンドの機能	57
3.10. [ヘルプ]メニューのコマンドの機能	57
5.1. 日付型フォーマットの例	129
5.2. 数値のユーザフォーマットの例	130
5.3. テキストエディタ - ツールバー	131
10.1. 処理レポートの内容	190
10.2. Mappingマッピングボックスに生成された処理レポートと、InfraTools Desktop Discoveryコネクタに取り込まれるDirectoryPoolerActionドキュメ ントタイプ間のマッピングの詳細	191
10.3. Asset Managementコネクタに生成される処理レポートと、InfraTools Desktop Discoveryコネクタに取り込まれるDirectoryPoolerActionドキュメ ントタイプ間のマッピングの詳細	192
11.1. Connect-ItのJava開発キットの内容	195

12.1. インベントリコネクタとデータベースコネクタ - 設定の詳細オプション	248
14.1. Action Request Systemコネクタの設定	277
14.2. Asset Managementコネクタの設定	284
14.3. InfraTools Desktop Discoveryコネクタの設定	291
14.4. InfraTools Managementコネクタの設定	300
14.5. InfraTools Network Discoveryコネクタの設定	305
14.6. PDIコネクタの設定	310
14.7. ゲートウェイコネクタの設定	319
14.8. ServiceCenterコネクタの設定	323
14.9. Asset Managementコネクタの処理レポートと、ServiceCenterコネクタのConnectItDelドキュメントタイプ間のマッピング	328
15.1. Lotus Notesコネクタの設定	330
15.2. MQSeriesコネクタ（読み取り）の設定	338
15.3. MQSeriesコネクタ（書き込み）の設定	347
15.4. NTセキュリティコネクタの設定	358
15.5. NTセキュリティコネクタ - 生成用ルールの例	360
16.1. データベースコネクタの設定	367
16.2. Eメール（受信）コネクタの設定	372
16.3. Eメール（送信）コネクタの設定	375
16.4. Eメール（受信）コネクタの生成用ドキュメントタイプ	378
16.5. Eメール（送信）コネクタの取り込み用ドキュメントタイプ	380
16.6. LDAPコネクタの設定	384
16.7. LDAPフィルタ	391
16.8. マップテーブルファイル（ASCII値の文字）	392
16.9. コマンドラインコネクタの設定	396
16.10. テキストコネクタの設定（読み取り）	401
16.11. テキストコネクタの設定（書き込み）	407
16.12. XMLコネクタの設定（読み取り）	421
16.13. XMLコネクタの設定（書き込み）	432
17.1. Intel LANDeskコネクタの設定	446
17.2. Tivoli Inventory Managementコネクタの設定	451
17.3. Tivoli Inventory Management 4.0コネクタの設定	457
17.4. SMS 1.xコネクタとSMS 2.xコネクタの設定	461
17.5. Unicenter AMOコネクタの設定	465
18.1. SCAutoリスニングコネクタの設定	471
18.2. XMLリスニングコネクタの設定	476
18.3. TEC adapterコネクタの設定	478
18.4. TECコネクタの設定	482
18.5. サポートされるプロトコル - Webサービスコネクタ	489
18.6. Webサービスコネクタの設定	489
18.7. 拡張型処理レポートの名前 - Webサービスコネクタ	491
18.8. JDBCコネクタの設定	492

A.1. 集合型のAQL関数	526
A.2. 文字列型のAQL関数	527
A.3. 日付型のAQL関数	527
A.4. 日付型のAQL関数	528
A.5. 数値型のAQL関数	529
A.6. テスト型のAQL関数	530
A.7. 表記法	531
A.8. 論理演算子	536
A.9. 比較演算子	537

Connect-Itの使用目的

Connect-Itは、EAI（Enterprise Application Integration）グループの一部を成す統合プラットフォームです。EAIソリューションを使用すると、企業の内部や外部でデータを取得または提供する様々なアプリケーション（内部用は備品管理ソフト、社内技術サポート、LDAPディレクトリ、外部用はERP、B2B、B2C）を、統合できるようになります。

Connect-Itは、データだけではなく企業アプリケーションのプロセスも統合します。

Connect-Itには以下の用途があります。

- あるデータベースから別のデータベースへデータを転送する。
- 2つの異種のデータベース間でデータを複製（レプリケーション）する。
- Eメール、区切り文字で区切られたテキストファイル、XMLファイルやその他のフォーマットのデータを、データベースにインポートする。
- データベースから、Eメール、区切り文字で区切られたテキストファイル、XMLファイルやその他のフォーマットへデータをエクスポートする。
- NTセキュリティ情報をデータベースにインポートする。
- その他

Connect-Itの対象ユーザ

Connect-Itは、企業内の異種のアプリケーション間の統合を担当するIT技術者を対象としています。

Connect-Itを使用するには以下の知識が必要になります。

- 統合されるアプリケーションに関する高度な知識
- マッピングスクリプトで使用されるBasic言語の知識
- JCA (Java Connector Architecture) 規格の知識 (Java開発キットを使ってコネクタを作成する場合)

本書の使用方法

「概要」の章

この章では、Connect-It機能の概要が説明されています。

「インストール」の章

この章では、Connect-Itのインストール方法が説明されています。

「シナリオビルダ」の章

この章では、統合シナリオ作成用のシナリオビルダのユーザインタフェースが説明されています。

「統合シナリオのインプリメンテーション」の章

この章では、統合シナリオのインプリメンテーション方法が説明されています。インプリメンテーションでは以下の操作を実行します。

- ソースコネクタとターゲットコネクタを選択する
コネクタ (データベース型コネクタ、Eメールコネクタなど) は外部アプリケーションと通信し、Connect-Itシナリオがこれらの外部アプリケーションを統合します。
- コネクタの生成用ドキュメントタイプと、取り込み用ドキュメントタイプを作成する
外部アプリケーションが生成するデータ全体 (データベーステーブル、Eメール、テキストファイル、プロパティフォーマットなど) をXML形式に変換したものが、ドキュメントタイプに当たります。

このインプリメンテーションに関する情報は、次章で説明されるマッピングの手順で補足されます。

「ドキュメントタイプのマッピング」の章

この章では、ソースコネクタの生成用ドキュメントタイプと、ターゲットコネクタの取り込み用ドキュメントタイプ間のマッピングが、説明されています。マッピングは、ソースコネクタの生成用ドキュメントタイプのデータを変換し、ターゲットコネクタがデータを取り込めるようにします。

「ピボットドキュメントタイプ」の章

この章では、ピボットドキュメントタイプが説明されています。ピボットドキュメントタイプは、マッピングなしで、ソースコネクタとターゲットコネクタ間でデータを転送できるようにします。

「統合シナリオのテストとデバッグ」の章

この章では、プロダクションモードで統合シナリオを使用する前に、テストとデバッグを行う方法が説明されています。

「シナリオ文書」の章

この章では、統合シナリオのシナリオ文書を動的に作成する機能が説明されています。このシナリオ文書には、シナリオで使用されるコネクタ、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、マッピングの詳細と、その他多数の情報が含まれています。

「統合シナリオをプロダクションモードにする」の章

この章では、統合シナリオをプロダクションモードにする方法が説明されています。シナリオのスケジュール（シナリオ実行の頻度）、Windowsのサービスとしてシナリオを使用する方法、Unixのデーモンとしてシナリオを使用する方法が解説されています。

「処理レポート」の章

この章では、コネクタが使用する処理レポートが説明されています。このレポートは、コネクタがデータを処理した方法を記述するドキュメントタイプです。データ処理の成功または失敗の情報は、他のコネクタのアクションをトリガするために使用されることもあります。アクションには、処理の停止、ヘルプデスクへの警告メッセージの送信などがあります。

「Connect-ItのJava開発キット（JDK）」の章

この章では、Connect-It付属のJava開発キットが説明されています。この開発キットを使うと、JCA（Java Connector Architecture）規格に基づいてJavaコネクタを作成できるようになります。

「コネクタの設定」の章

この章では、コネクタの設定方法が説明されています。コネクタを設定すると、コネクタは外部アプリケーションと通信できるようになります。この章には、複数のコネクタに共通の設定ウィザードのページがまとめられています。コネクタに特有の設定ページは、各コネクタに関する章で解説されています。

「コネクタのルール（ディレクティブ）」の章

この章ではコネクタのルール（ディレクティブ）が説明されています。ルールとは、コネクタが外部アプリケーションに書き込み、送信する方法、または外部アプリケーションから読み取り、受信する方法を指定する指示に当たります。例：フィルタ（Where句）のルールを使うと、データベース型のコネクタは、データベースから読み取る際にレコードにフィルタを適用できるようになります。

コネクタの章

以下の章

- 「Peregrine Systemsコネクタ」
- 「アプリケーションコネクタ」
- 「プロトコルコネクタ」
- 「インベントリコネクタ」
- 「Javaコネクタ」

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

「AQL言語」の章

この章では、AQL言語が説明されています。この言語は、Asset Managementコネクタの生成用ルール内で使用されます。

コネクタの生成用ルールについては、「コネクタのルール(ディレクティブ)」章の「生成用ルール」の節を参照してください。

「問題点の報告方法」の章

この章では、Pegrine Systemsヘルプデスクに問題を報告する方法が説明されています。Pegrine Systemsの全製品に共通する情報に加えて、「[Connect-Itに関する問題を報告する場合](#)」の節では、Connect-It用の報告方法が解説されています。

「用語解説」の章

この用語解説では、Connect-It特有の専門用語の一部が説明されています。

1 | 概要

Connect-Itでのデータ処理

図 1.1. Connect-Itで、あるデータベースから別のデータベースヘデータを転送する



Connect-Itは、コネクタを使用して外部アプリケーションと相互に作用します。あるデータベースから別のデータベースヘデータを転送する場合、

- ソースコネクタはXML形式のドキュメントを生成します。
生成される各ドキュメントは、ソースアプリケーション内の1データ集合に一致します。
- マッピングボックスが、ソースコネクタにより生成されたドキュメントの構造を再構築し、ターゲットコネクタがドキュメントを取り込めるようにします。
- ターゲットコネクタがXML形式のドキュメントを取り込みます。
取り込まれた各ドキュメントは、ターゲットアプリケーション内の1データ集合に一致します。

データベース型のコネクタでは、各ドキュメントは1データベーステーブルのレコード（または別のテーブルへのリンク）に対応しています。例：AssetCenter データベースでは、1ドキュメントは資産のテーブルのデータを含みます。

他のコネクタの場合、データ集合は、区切り文字で区切られたテキスト、Eメールメッセージやセキュリティ情報などに当たります。

コネクタの選択、コネクタ間の関係、そしてソースドキュメントとターゲットドキュメント間のマッピングの定義が、統合シナリオを構成します。テストとデバッグを経た後、シナリオはスケジューラに関連付けられプロダクション段階に入ります。

どの統合過程でも起こるように、ドキュメントの一部または全体が拒否されることもあります。しかし処理レポートやドキュメントログを使用すると、統合シナリオ全体を定義し直すことなく、拒否されたドキュメントを再処理することができます。

2 | インストール

本章ではConnect-It統合プラットフォームのインストール方法を説明します。内容は以下の通りです。

- 必要な動作環境
- 最初のインストール
- 認証証明書
- 追加コネクタのインストール
- ファイルの構成
- アンインストール

必要な動作環境

Connect-ItはWindowsで使用可能です。

Connect-ItをUnix上（Linux、Solaris、AIX）にインストールすることは可能ですが、この場合Connect-Itは非グラフィックモードでのみ機能します。

Connect-Itは、シナリオに関連するアプリケーションにアクセスできるコンピュータ上に、インストールされなければなりません。一般的に、シナリオに関連するアプリケーションのクライアント部分を、コンピュータに完全インストールする必要はありません。しかし最初のテスト段階などでは、外部アプリケーション

ン内へ書き込まれるデータ結果を確認するために、コンピュータにアプリケーションをインストールしておく便利です。

ドキュメントのトラッキング管理は、以下の要素をハードディスクに保存します。

- ドキュメント (「.dat」ファイル)
- トラッキング項目とその内容を説明するメッセージ (「.msg」ファイル)
- 迅速なアクセスを可能にするドキュメントの完全なインデックス (「.idx」ファイル)

ログに割り当てられるメモリ容量は、シナリオビルダの **[編集 / オプション]** メニューのコマンドで指定できます。ログに割り当てられるデフォルトのメモリ容量は5MBです。

ドキュメントログとドキュメントのトラッキング管理についての詳細は、「シナリオビルダ」の章の「ログ」の「ドキュメントログ」を参照してください。

最初のインストール

Connect-Itは、Windows 32ビットまたはUNIX上にインストール可能です。

Windows 32ビット上のインストール

Connect-Itをインストールするには、

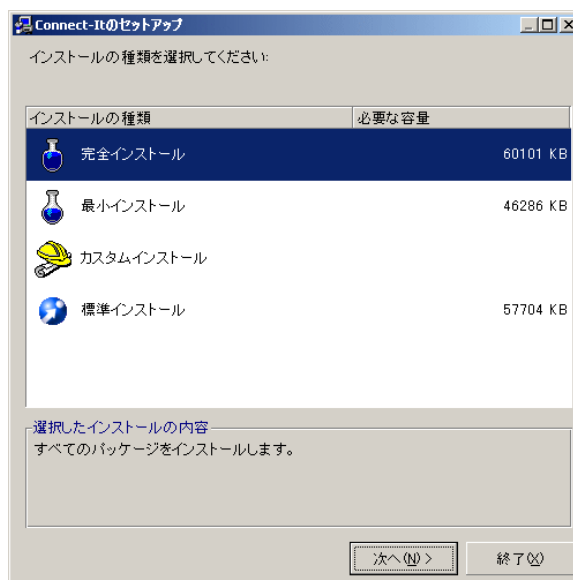
- 1 必要な動作環境があるか確認します。
- 2 Windowsを管理者の権限で起動します。

CD ROMを挿入すると、インストールプログラムが自動的に起動します。

自動的に起動しない場合はConnect-Itインストール先フォルダを開き、「setup.exe」プログラムを起動します。

インストールプログラムでは以下の4種のインストールから選択できます。

- 完全インストール
- 最小インストール
- カスタムインストール
- 標準インストール



最小インストール

最小インストールには以下のコンポーネントが含まれています。

- シナリオビルダ (Connect-It)
- Connect-Itコンソール。これによりシナリオに関連したサービスを作成できます。
- LOGファイルを読み取るためのログビューア
- ベースコネクタ
- Connect-Itデータキットには以下の要素が含まれています。
 - インストールされるコネクタに関連する既製シナリオ
 - 既製シナリオの使用に必要な関連ファイル

標準インストール

標準インストールには、最小インストールに加えてConnect-ItのPDFとCHM形式のマニュアルが含まれています。

完全インストール

完全インストールには、標準インストールと以下のコンポーネントが含まれています。

- オプションコネクタ

- これは認証証明書が許可している場合に使用可能です。
- オプションコネクタに関連する既製シナリオ

カスタムインストール

カスタムインストールでは、最小インストールを実行した上で、必要なオプションコネクタを追加することができます。

オプションコンポーネントをインストールすると、コンポーネントに関連する既製シナリオもインストールされます。

UNIX上のインストール

Connect-ItをUNIX上にインストールするには、以下のコマンドラインを実行して付属の「.tgz」ファイルを圧縮解除します。

```
tar xvfz cnit-[オペレーティングシステム]_[バージョン]_[ビルド]_[言語の接頭文字].tgz
```

使用可能な接頭文字は、フランス語版では「fr」、英語版では「en」、イタリア語版では「it」、スペイン語版では「es」、ドイツ語版では「de」です。

UNIX上のConnect-Itのインストールは、常に完全インストールになっています。このインストールには以下のコンポーネントが含まれています。

- ベースコネクタ
- オプションコネクタ
- データキット
- 文書
- 用例シナリオ

注意:

コネクタの中には、Windows 32ビット環境でしか機能しないものもあります。コネクタの互換性に関する詳細情報については、ペレグリンシステムズのサポート用Webページ<http://support.peregrine.com>を参照してください。

UNIX下のダイナミックライブラリ (.so)

Connect-Itでは、Connect-Itのインストール先フォルダの「bin」フォルダにあるダイナミックライブラリ (.so) を使用します。パス[Connect-Itのインストール先フォルダ]/binを、環境変数LD_LIBRARY_PATHに追加する必要があります。コマンドC (csh) のインタープリタには、以下のコマンドラインを実行します。

```
setenv LD_LIBRARY_PATH=[Connect-Itのインストール先フォルダ]/bin
```

コマンドK (ksh) や Bourne (sh) のインタプリタには、「.profile」ファイル内で以下のコマンドラインを実行します。

```
LD_LIBRARY_PATH=[Connect-Itのインストール先フォルダ]/bin export LD_LIBRARY_PATH
```

ベースコネクタ

ベースコネクタは、Connect-Itのインストールモードに関わらず必ずインストールされるコネクタです。

ベースコネクタは以下の通りです。

- Asset Managementコネクタ
- Action Request Systemコネクタ
- コマンドラインコネクタ
- InfraTools Desktop Discoveryコネクタ
- InfraTools Network Discoveryコネクタ
- InfraTools Managementコネクタ
- NTセキュリティコネクタ
- ゲートウェイ3.xコネクタ
- ServiceCenterコネクタ
- テキストコネクタ
- XMLコネクタ

オプションコネクタ

オプションコネクタは、Connect-Itの完全インストールを選択した場合、またはカスタムインストールでオプションコネクタを選択した場合にインストールされます。

オプションコネクタは以下の通りです。

- データベースコネクタ
- Eメールコネクタ
- XMLリスニングコネクタ
- Intel LANDeskコネクタ
- LDAPコネクタ
- Lotus Notesコネクタ
- MQSeriesコネクタ
- SCAutoリスニングコネクタ

- SMS 1.xコネクタとSMS 2.xコネクタ
- Tivoli Enterprise Consoleコネクタ（送信）
- Tivoli Enterprise Consoleコネクタ（受信）
- Tivoli Inventoryコネクタ（バージョン3.1と3.6）
- Unicenter AMOコネクタ
- Webサービスコネクタ
- JDBCコネクタ

データキット

次の表はデータキットの内容を説明しています。

データキット

ファイル名	機能
iddフォルダ *.fsf	Desktop Discoveryによるコンピュータのスキャンの例です。スキャンにはコンピュータ自体 (Hardware)、コンピュータにインストールされたソフトウェア (Software) とコンピュータに関連する資産とユーザ (UserAndAssets) についての情報が含まれています。
master.sai、 french.sai	このファイルにより、Desktop Discoveryは大多数のアプリケーションの情報を検索できます。ファイルには、各アプリケーションごとにバージョン、ソフトウェア会社、ライセンス番号などの情報が含まれています。この情報がない場合は、「.fsf」ファイル内にソフトウェアのファイル名だけが列挙されます。
iddsdt.cdt	このファイルは、Desktop Discoveryスキャナ生成プログラム (Scanner Generator) により生成されたデフォルトのスキャナに対応するドキュメントタイプを含んでいます。
datakit.cdt	このファイルはサンプルのスキャン (インベントリ) に対応するドキュメントタイプを含んでいます。
scac36とscac41フォルダ modelfeat.scr modelfeat.txt	このスクリプトファイルにより、Asset Managementアプリケーション内で任意管理項目を作成できるようになっています。任意管理項目は、ServiceCenter-AssetCenter複製 (レプリケーション) シナリオ内のServiceCenterの標準フィールドに関連付けられています。
acフォルダ stdfeatサブフォルダ	スクリプトファイル (sdtfeat.scr) をインポートすると、Asset Managementアプリケーション内の任意管理項目テーブル内にレコードが作成されます。これらの任意管理項目は、Asset Managementコネクタが発行する使用可能なドキュメントタイプ内に現れます。

データキット

tcフォルダ	このフォルダにあるファイルにより、TeleCenterコネクタを含む既製のシナリオが使用可能になります。
indフォルダ acサブフォルダ	このフォルダは、Asset Managementアプリケーション内でカテゴリと任意管理項目を作成するインポートスクリプトを含んでいます。
doctransフォルダ	このフォルダは、シナリオ文書の自動生成に必要なファイルを含んでいます。

追加コネクタ

追加コネクタは、デフォルトではシナリオビルダと共にインストールされません。これは、ペレグリンシステムズから提供されたCD-ROMを用いて、別々にインストールする必要があります。各追加コネクタ用にペレグリンシステムズから認証証明書が発行されます。

認証証明書

Connect-Itのインストールが終了した後、ペレグリンシステムズから入手した認証証明書をシナリオビルダ内に入力する必要があります。

認証証明書は以下の内容を含むテキストファイルです。

- 使用が許可されているオプションコネクタのリスト
- 各オプションコネクタの有効期限
- 暗号化された認証キー

認証証明書の入力

認証証明書でコネクタを有効にするには、

- 1 Connect-Itのシナリオビルダを起動します。
- 2 **[管理 / 認証証明書の編集]**メニューを選択します。
- 3 表示されるダイアログボックスに認証証明書を入力します。
- 4 **[OK]**をクリックします。

このアクションは、Connect-Itインストール先フォルダ内にlicense.txtを作成します。

 注意:

UnixバージョンのConnect-Itにはグラフィカルインタフェースがありません。認証証明書を入力するには、以下の手順に従います。

- 1 ペレグリンシステムズから提供されるライセンスファイルの内容をコピーします。このファイルの内容は変更しないでください。
- 2 このファイルに「license.txt」と名前を付けてから保存します。
- 3 「license.txt」をConnect-Itのインストール先フォルダにコピーします。

ファイルの構成

Connect-Itインストール先フォルダのファイルの構成は、以下の表の通りです。

フォルダ名	主なファイル
bin32	<ul style="list-style-type: none"> 実行可能ファイル (conitgui.exe、conitsvc.exe、console.exe、logview.exe) ダイナミックライブラリ (DLL) Connect-It専用のテキストエディタの設定ファイル (codeedit.cfg) 既製シナリオとConnect-Itの機能に必要な多種のファイル
config	全シナリオ内のコネクタに関連したファイル <ul style="list-style-type: none"> .mpt (マップテーブル) .str (文字列テーブル) .bas (Basic関数とグローバル変数)
datakit	Connect-Itツールで使用されるデータキット
doc	「.pdf」と「.chm」フォーマットの文書
fsf	「.fsf」ファイルのバックアップコピーの保存用フォルダ
scenario	<ul style="list-style-type: none"> Connect-Itの用例シナリオを含んだSCNファイル シナリオ用のコネクタに関連したファイル
lib	JARファイルには次の内容が含まれていません。 <ul style="list-style-type: none"> Peregrine Systemsに開発されたJavaクラス Javaサードパーティクラス (共有クラス)
wizards	Connect-Itの機能に必要なファイル

関連したファイル

数種のファイル (「.mpt」、「.str」、「.bas」ファイル) は、シナリオに関連付けられています。シナリオを別のフォルダへ移す場合は、これらのファイルもフォルダと共に移す必要があります。移動先では、シナリオファイルとの移動前の位置関係を再現しなければなりません。例：複数の用例シナリオに共通のファイルは「shared」フォルダにあります。このフォルダはシナリオの移動先にも作成されなければなりません。

 注意:

ファイルマネージャを使用してシナリオを移動させるよりも、可能な限りシナリオビルダ内で直接移動先に保存することをお勧めします。これにより、シナリオに関連した全ファイルは、移動後も適切に参照されるようになります。

3 | シナリオビルダ

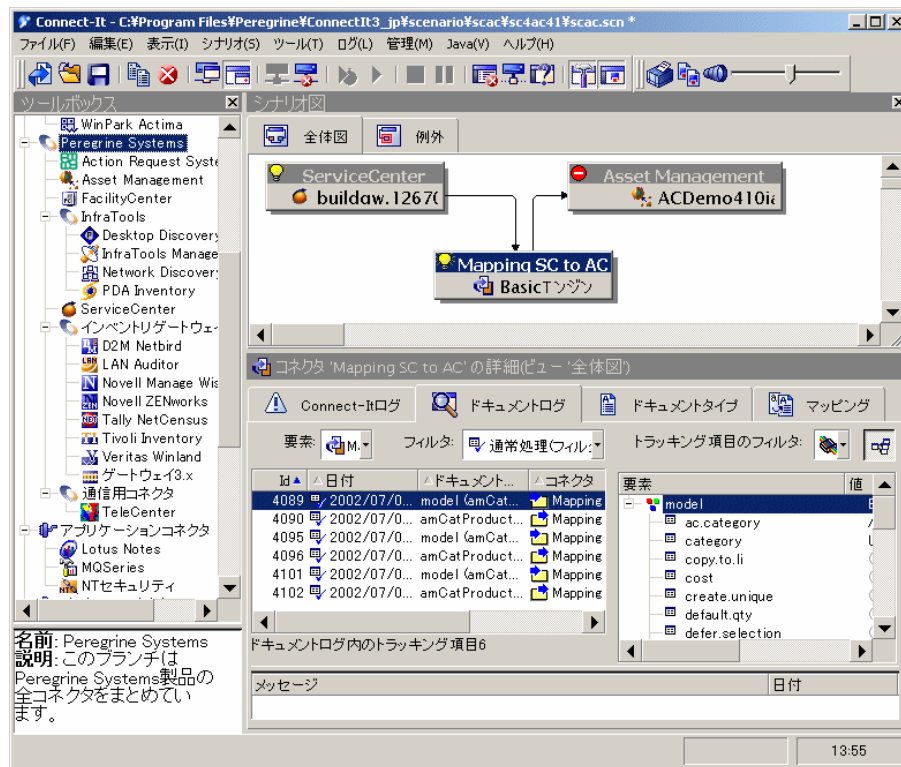
シナリオビルダは統合シナリオを作成するためのグラフィカルインターフェイスです。本章ではシナリオビルダの主要部分と、Connect-Itのオプションウィンドウについて説明します。

メインウィンドウ

シナリオビルダのメインウィンドウは3つの部分から成っています。




- シナリオのコンポーネントを選択するためのツールボックス
- シナリオ内のコンポーネントとコンポーネント間のリンクがグラフィックで表示されるシナリオ図部分
- 以下のタブを含む詳細画面
 - Connect-Itログのタブ
 - ドキュメントログのタブ
 - シナリオ図内で選択したコンポーネントに関連するタブ：生成用、または取り込み用ドキュメントタイプ、マッピング、ピボットマッピング。


















図 3.1. シナリオビルダ - メインウィンドウ



頻繁に行う操作は、メニューバーを使用せずにツールバーで実行できます。ツールバー上で右クリックすると、ツールバーの位置と表示方法を設定できます。

表 3.1. シナリオビルダ - ツールバーのアイコン

アイコン	機能
	ウィザードを起動し、1つのソースコネクタと1つのターゲットコネクタを使った最小シナリオを作成できるようにします。
	既存シナリオを開きます。
	シナリオを保存します。

アイコン	機能
	Connect-Itのテキスト要素をコピーします。 <ul style="list-style-type: none"> • ツールボックス内のコネクタの名前 • ログ内のメッセージ • その他
	シナリオ内で選択したコンポーネントを削除します。
	選択したコンポーネントの設定ウィザードが起動します。
	コンポーネントの設定ウィザードで、高度な設定ページを表示します。(例: ピボットドキュメントタイプの使用のページ)
	シナリオの全てのコネクタを開きます。
	シナリオの全てのコネクタを閉じます。
	シナリオをスケジュールモードで起動します。
	シナリオをテストします (非スケジュールモード)。
	シナリオを停止します。
	シナリオを一時停止します。
	【Connect-It ログ 】タブと【 ドキュメントログ 】タブ内の項目を削除します。
	作業のオフライン/オンラインモードを変更します。
	ターゲットコネクタを外部アプリケーションと相互作用させずに、シナリオをテストします。 トランザクションをサポートするデータベース型のコネクタでは、エラーメッセージが表示されます。エラーメッセージは、このオプションを使用しない場合に取得されるメッセージに相当します。
	【 ツールボックス 】枠を表示/非表示にします。
	【 シナリオ図 】枠を表示/非表示にします。
	シナリオ図を印刷します。 (印刷前のプレビュー用ウィンドウで、印刷をパラメータ設定できます。)
	シナリオ図をクリップボードにコピーします。

アイコン**機能**

このアイコンの横のレバーでシナリオ図を縮小 / 拡大します。

シナリオ図のビュー

【シナリオ図】枠には、2つのタブ【全体図】と【例外】があります。【全体図】タブは、シナリオの全コンポーネント（ソースコネクタ、ターゲットコネクタ、リンクとマッピングボックス）を表示します。【例外】タブにはデフォルトでは何も表示されませんが、このタブはシナリオのエラーの管理用に用意されています。


【シナリオ図】枠内でビューを使用すると、一部のマッピングで使用されるコンポーネントのみを表示できるようになります。

例：複数のソースコネクタを使用するシナリオで、1つのソースコネクタのみを含むマッピングを選択すると、このソースコネクタのみが表示されるようになります。

シナリオ図のビューを使用する

あるシナリオ用にビューを作成すると、このシナリオをシナリオビルダ内で開くたびに、ビューが表示されます。

シナリオ図のビューの作成方法


- 1 シナリオを開きます。
- 2 【シナリオ/ビューの管理】を選択します。
- 3 表示されるウィンドウで、をクリックします。
- 4 ビューに名前を付けます。ビューの名前はシナリオ内で固有の名前でなければなりません。
- 5 シナリオ図のビューで表示するマッピングを選択します。
- 6 【作成】をクリックします。
ビューエディタを閉じると、【シナリオ図】枠内にビューの名前がついたタブが表示されます。

シナリオ図のビューの変更方法

- 1 シナリオを開きます。
- 2 【シナリオ/ビューの管理】を選択します。

- 3 ビューを選択します。
- 4 ビューの名前やマッピングの選択内容を変更します。
- 5 **[変更]**をクリックします。
- 6 **[閉じる]**をクリックします。

シナリオ図のビューの削除方法

- 1 シナリオを開きます。
- 2 **[シナリオ/ビューの管理]**を選択します。
- 3 ビューを選択します。
- 4 をクリックします。
このビュー用のタブが**[シナリオ図]**枠に表示されなくなります。

シナリオ図のビューに関する規則

シナリオ図のビューは以下の規則に従います。

- **[全体図]**以外のビュー内のコンポーネントを、ドラッグアンドドロップで移動させることはできません。
- ビュー内の1つのマッピングを削除しても、対応するSCNファイルの<LAYER>要素はなくなりません。
- マッピング、コネクタとリンクの編集は、**[全体図]**ビューのみで実行可能です。
- ビュー内のコンポーネントをクリックすると、ドキュメントログ内に関連するトラッキング項目のみが表示されるように、フィルタを設定できます。
- ビューは、Connect-Itの他の機能には影響しません。

メニュー

本節では以下の内容について説明します。

- シナリオビルダのメニューバー
- シナリオビルダのショートカットメニュー

シナリオビルダのメニューバー

[ファイル]メニュー

[ファイル]メニューのコマンドのリストは以下の通りです。

表 3.2. [ファイル]メニューのコマンドの機能

コマンド	機能
新規作成	[シナリオの設定]ウィザードを起動します。このウィザードでは、ソースコネクタとターゲットコネクタを設定し選択します。ウィザードは2つのコネクタをリンクするマッピングボックスを作成します。
開く	既に作成されたシナリオファイル(「.scn」ファイル)を開きます。
保存...	シナリオを保存します。
名前を付けて保存...	開いたシナリオの元の名前とは別の名前を付けて、シナリオを保存します。
シナリオの履歴	現在のシナリオ用の情報を入力するためのウィンドウを表示します。
シナリオ図を印刷する	シナリオ図の印刷用パラメータを入力するためのウィンドウを表示します。
シナリオ図をコピーする	シナリオ図の画像をクリップボードにコピーします。この画像をJasc Paint Shop ProやAdobe Photoshopなどのグラフィック用アプリケーション内に貼り付けることができます。
シナリオ文書を作成する	シナリオ文書をXML、DBKまたはHTMLフォーマットで作成できます。 この機能の使用方法については、「シナリオ文書」章の「シナリオ文書の作成」の節を参照してください。
HTMLシナリオ文書を表示する	現在のシナリオ文書をインターネットブラウザ内に表示します。 この機能の使用方法については、「シナリオ文書」章の「シナリオ文書をHTMLフォーマットで表示する」の節を参照してください。

コマンド	機能
シナリオ文書のプロパティ	この機能の詳細については、「シナリオ文書」の章の「シナリオ文書のプロパティを編集する」の節を参照してください。シナリオ文書のプロパティのウィンドウが起動します。
終了	シナリオビルダを終了します。
ファイルのリスト	最近編集したシナリオを表示します。リストの1要素をクリックすると、シナリオを直接開くことができます。

[編集] メニュー

[編集] メニューのコマンドのリストは以下の通りです。

表 3.3. [編集] メニューのコマンドの機能

コマンド	機能
切り取り	
コピー	Connect-Itログまたはドキュメントログの項目を、コンピュータのクリップボードにコピーします。
貼り付け	
削除	シナリオから選択したコンポーネントを削除します。
オプション...	シナリオビルダのオプションウィンドウを表示します。

[表示] メニュー

[表示] メニューのコマンドのリストは以下の通りです。

表 3.4. [表示] メニューのコマンドの機能

コマンド	機能
ツールボックス	コネクタのリストを含む枠を表示 / 非表示にします。
シナリオ図	シナリオ図を表示 / 非表示にします。
ツールバー	ツールバーを表示 / 非表示にします。



コマンド	機能
図のバー	シナリオ図用のツールバーを表示 / 非表示にします。

[シナリオ] メニュー

[シナリオ] メニューのコマンドのリストは以下の通りです。

表 3.5. [シナリオ] メニューのコマンドの機能


コマンド	機能
全コネクタを開く	シナリオの全てのコネクタとマッピングボックスを開きます。
全コネクタを閉じる	シナリオの全てのコネクタとマッピングボックスを閉じます。
内部コネクタを表示する	ピボットドキュメントタイプを使用している場合に、Connect-It が自動的に作成する内部コネクタを表示します。
キャッシュの同期をとる	シナリオ内で開かれた全コネクタのキャッシュファイルの同期をとります。
キャッシュを削除する	シナリオ内で開かれた全コネクタのキャッシュファイルを削除します。
ビューの管理	ビューエディタを表示します。 ビュー管理の詳細については、本章の「シナリオ図のビュー」の節を参照してください。
スケジューラ	スケジューラの編集ウィンドウを表示します。
スケジュール	スケジュールの編集ウィンドウを表示します。
文字列テーブル...	文字列テーブルの編集ウィンドウを表示します。
マップテーブル...	マップテーブルの編集ウィンドウを表示します。
グローバル関数...	グローバル関数と変数の編集ウィンドウを表示します。
ユーザフォーマット	マッピングスクリプトで使用される日付と数値のフォーマットを作成できます。

コマンド	機能
オフラインで作業する	オフラインで作業できるようにします。 ツールバーの  をクリックすると、このオプションをオンにすることができます。 オフライン作業の詳細については、「統合シナリオのテストとデバッグ」章の「オフラインで作業する」の節を参照してください。
シナリオのテストモード	シナリオをテストする際に、ターゲットコネクタが外部アプリケーションにデータを転送しないように設定します。 ツールバーの  をクリックすると、このオプションをオンにすることができます。
全スケジューラを起動する	シナリオのソースコネクタの生成用ドキュメントタイプに関連するスケジューラを起動します。
停止	シナリオを停止します。
一時停止	シナリオの実行中に一時停止します。

[ツール] メニュー

[ツール] メニューのコマンドのリストは以下の通りです。

表 3.6. [ツール] メニューのコマンドの機能

コマンド	機能
設定...	選択したコンポーネントの設定ウィザードが起動します。
高度な設定	コンポーネントの設定ウィザードで高度な設定ページを表示します ([F2] キー)。 ツールバーの  をクリックすると、このオプションをオンにすることができます。
開く	選択したコンポーネントを開きます。
閉じる	選択したコンポーネントを閉じます。
キャッシュ / キャッシュの同期をとる	選択したコネクタのキャッシュファイルの同期をとります。
キャッシュ / キャッシュを削除する	選択したコネクタのキャッシュファイルを削除します。

コマンド	機能
ドキュメントタイプを編集する	[ドキュメントタイプの選択] ウィザードが起動します。選択したコンポーネントの生成用、または取り込み用ドキュメントタイプを選択します。
マッピングを編集する	[マッピングの選択] ウィザードが起動します。選択したマッピングボックスのマッピングを選択できます。
生成する	ドキュメントタイプが作成されているコンポーネントのドキュメントを生成します。

[ログ] メニュー

[ログ] メニューのコマンドのリストは以下の通りです。

表 3.7. [ログ] メニューのコマンドの機能

コマンド	機能
ドキュメントログを設定する...	ドキュメントログを設定するためのウィンドウを表示します。
ドキュメントログを更新する	ドキュメントログ内の内容を更新します。
ドキュメントログを再び読み込む	[ドキュメントログ] タブ内に、ドキュメントログファイルに保存されたトラッキング項目全体を表示します。
表示されたトラッキング項目を削除する	[ドキュメントログ] タブから、表示されているトラッキング項目全体を削除します。
格納されたトラッキング項目を削除する	ドキュメントログファイルに保存されているトラッキング項目を削除します。

[管理] メニュー

[管理] メニューのコマンドのリストは以下の通りです。

表 3.8. [管理] メニューのコマンドの機能

コマンド	機能
認証証明書を編集する...	認証証明書の編集ウィンドウを表示します。

[Java] メニュー

[Java] メニューのコマンドのリストは以下の通りです。

表 3.9. [Java] メニューのコマンドの機能

コマンド	機能
JMVを設定する	Javaクラスのパスを指定するためのウィンドウが表示されます。
コネクタを導入する	作成するJavaコネクタの導入用ウィンドウが表示されます。

[ヘルプ] メニュー

[ヘルプ] メニューのコマンドのリストは以下の通りです。

表 3.10. [ヘルプ] メニューのコマンドの機能

コマンド	機能
オンラインヘルプ	Connect-Itのオンラインヘルプを表示します。
ご存知でしたか?	[ご存知でしたか?] ボックスを表示します。
バージョン情報...	使用中のConnect-Itのバージョンに関する一般的な情報を含む [バージョン情報...] ウィンドウを表示します。

シナリオビルダのメインウィンドウ内のショートカットメニュー

シナリオビルダのメインウィンドウ内では、以下の場合に右クリックメニュー（ショートカットメニュー）が表示されます。

- カーソルがシナリオ図部分にある場合
- カーソルがConnect-Itログのタブ内にある場合
- カーソルがドキュメントログのタブ内にある場合

カーソルがシナリオ図部分にある場合

カーソルがシナリオ図部分にある場合に表示されるショートカットメニューの
コマンドのリストは、以下の通りです。

コマンド	機能
コネクタを設定する...	選択したコンポーネントを設定するための 【コネクタの設定】ウィザードが起動しま す。
コネクタを開く	選択したコンポーネントを開きます。
コネクタを閉じる	選択したコンポーネントを閉じます。
キャッシュ/キャッシュの同期をとる	選択したコネクタのキャッシュファイルの 同期をとります。
キャッシュ/キャッシュを削除する	選択したコネクタのキャッシュファイルを 削除します。
ドキュメントタイプを編集する	【ドキュメントタイプの選択】ウィザード が起動します。選択したコンポーネントの 生成用、または取り込み用ドキュメントタ イプを選択します。
マッピングを編集する	【マッピングの選択】ウィザードが起動し ます。選択したマッピングボックスのマッ ピングを選択できます。
生成する	ドキュメントタイプが作成されているコン ポーネントのドキュメントを生成します。
削除	選択したコンポーネントを削除します。
トラッキング項目の表示	選択したコンポーネントに関連するトラッ キング項目をドキュメントログ内に表示し ます。
ツールボックスを表示する	ツールボックスを表示 / 非表示にします。
シナリオ図を表示する	シナリオ図を表示 / 非表示にします。

カーソルがConnect-Itログのタブ内にある場合

コマンド	機能
1レベル表示	トラッキング項目を1レベル分表示します。
全レベル表示	トラッキング項目を全レベル表示します。
全てのレベルを非表示	トラッキング項目の全てのレベルを非表示 にします。
ツールボックスを表示する	ツールボックスを表示 / 非表示にします。
シナリオ図を表示する	シナリオ図を表示 / 非表示にします。

カーソルがドキュメントログのタブ内にある場合

ドキュメントログ内では2つのショートカットメニューがあります。

- トラッキング項目上にカーソルがある場合
- ソースドキュメントの詳細の1要素上にカーソルがある場合

トラッキング項目上にカーソルがある場合

コマンド	機能
このXML文書をコピーする	コンピュータのクリップボードに、トラッキング項目に対応するXML文書をコピーします。
このXMLのDTDをコピーする	コンピュータのクリップボードに、トラッキング項目に対応するXMLのDTD（文書型定義）をコピーします。
このXML文書を開く	オペレーティングシステムで指定されている、XMLファイルに関連したアプリケーションでXML文書を開きます。（例：Internet Explorer）
ツールボックスを表示する	ツールボックスを表示 / 非表示にします。
シナリオ図を表示する	シナリオ図を表示 / 非表示にします。

ソースドキュメントの詳細の1要素上にカーソルがある場合

コマンド	機能
このXML文書をコピーする	コンピュータのクリップボードにXML文書をコピーします。
このXMLのDTDをコピーする	コンピュータのクリップボードに、トラッキング項目に対応するXMLのDTD（文書型定義）をコピーします。
このXML文書を開く	オペレーティングシステムで指定されている、XMLファイルに関連したアプリケーションでXML文書を開きます。（例：Internet Explorer）
パスをコピーする	コンピュータのクリップボードに、選択した要素のパスをコピーします。
ツールボックスを表示する	ツールボックスを表示 / 非表示にします。
シナリオ図を表示する	シナリオ図を表示 / 非表示にします。

ログ

シナリオビルダには2つのログがあります。

- Connect-Itログ
- ドキュメントログ

Connect-Itログ

ユーザはConnect-Itで実行された全アクションをログ内で確認できます。

例：コネクタが開いたことを知らせるメッセージが表示されます。

シナリオビルダの【Connect-Itログ】タブを選択すると、Connect-Itログにアクセスできます。

ログでは各アクションがアイコンで表示されます。アクションのメッセージには、場合によってアクションの詳細を説明するサブメッセージがあります。サブメッセージには更にサブメッセージが付くこともあります。

各メッセージには、アクションが起動した日付が付きます。





ログ内で右クリックするとショートカットメニューが表示され、1レベル表示、全レベル表示または非表示を選択できます。


図 3.2. Connect-Itログ

メッセージ	日付
セッションを開いています...	2001/10/08 16:18:29
コネクタ 'Mapping (Basic エンジン)' 用セッションの開始	2001/10/08 16:18:30
スクリプトをコネクタ 'Mapping (Basic エンジン)' 用にコンパイルしています...	2001/10/08 16:18:30
コネクタ 'Asset Management (ACDemo360 JPN)' 用セッションの開始	2001/10/08 16:18:30
整合性の解析	2001/10/08 16:18:30
整合性の解析 'amAsset'	2001/10/08 16:18:30
インデックス 'Categ_FullName' は 'amAsset.Product.Category' と共に整合性チェックに...	2001/10/08 16:18:30
インデックス 'Prod_CatgIdBrand' は整合性チェック用に 'amAsset.Product' (キー: 'Cat...	2001/10/08 16:18:30
インデックス 'EmplDept_NameFirst' は整合性チェック用に 'amAsset.User' (キー: 'Name...	2001/10/08 16:18:30
インデックス 'Loca_FullName' は 'amAsset.Location' と共に整合性チェックに使用されて...	2001/10/08 16:18:30
インデックス 'FVAst_AstIdFeat[]' は 'amAsset.FeatValues.Feature' と共に整合性チェックに...	2001/10/08 16:18:30
インデックス 'FVAst_PublisherFeat[]' は 'amAsset.FeatValues' と共に整合性チェックに使用さ...	2001/10/08 16:18:30
インデックス 'Soft_PublisherName[]' は 'amAsset.SoftInstalls.Software' と共に整合性チェ...	2001/10/08 16:18:30
インデックス 'SoftInst_StationI[]' は整合性チェック用に 'amAsset.SoftInstalls' (キー: 'IS...	2001/10/08 16:18:30

Connect-Itログ内で取得されるメッセージの例については、「コネクタのルール（ディレクティブ）」章の「取り込み用ルール」節の「整合性」の「識別キーに関するエラーメッセージ」を参照してください。

Connect-Itログ内で使用されるアイコン

トラッキング項目	意味	メッセージのタイプ	例
	アクションについての情報	Connect-Itのアクションを説明します。	コネクタの開始
	アクションの詳細	アクションを詳細に説明します。	整合性チェックの段階で特定のインデックスが使用されている。
	警告	データの不適切な処理の原因となる問題点を報告します。	識別キーとして選択されたフィールドは重複を許可している（一意性がない）。
	重大な問題	アクションが失敗した理由を説明します。	コネクタの設定が無効である。外部アプリケーションとの通信が不可能。

ログから全メッセージを削除するには、[ログ]メニューの[表示されたトラッキング項目を削除する]オプションを選択するか、またはをクリックします。

Connect-Itログを削除するとドキュメントログも削除されます。

ドキュメントログ

ドキュメントに以下の操作が実行された後、ドキュメントログでドキュメントの詳細を参照、確認できます。

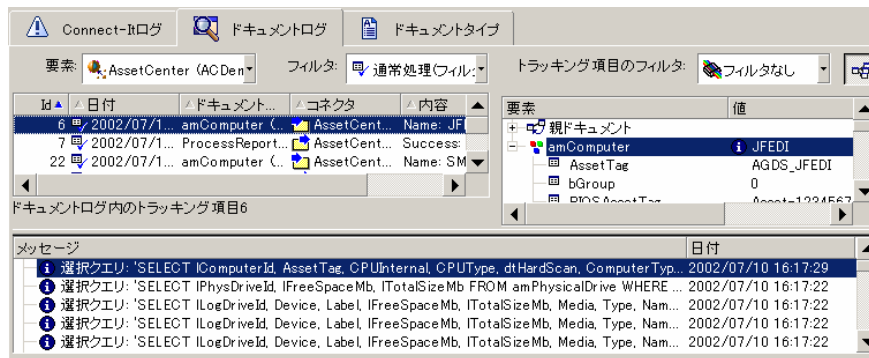
- ドキュメントがコンポーネントにより生成される、または取り込まれる。
- ドキュメントがリンクにより送信される。

シナリオビルダで[ドキュメントログ]タブを選択すると、ドキュメントログが表示されます。

このタブは3つの部分から成っています。

- 1ドキュメントにつき1つのトラッキング項目が表示される部分（左）
- 選択したドキュメントの詳細項目が表示される部分（右）
- ドキュメント内の要素が残したトラッキング項目の、詳細メッセージが表示される部分（下）

図 3.3. シナリオビルダ - ドキュメントログのタブ

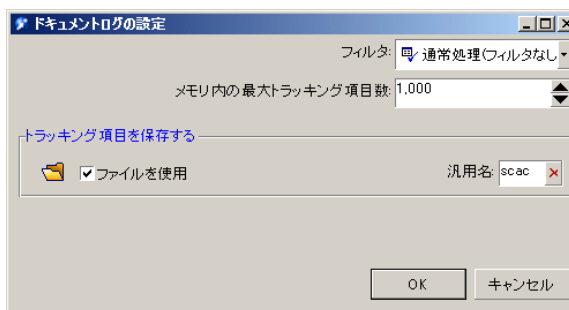


ドキュメントログの設定

ドキュメントログを設定するには以下の操作を実行します。

- ドキュメント処理のエラーの種類に基づいて、トラッキング項目にフィルタをかけます。
- ドキュメントログに表示されるトラッキング項目の最大数を指定します。
- ログに表示されるトラッキング項目を保存するかどうかが決めます。

図 3.4. ドキュメントログの設定ウィンドウ



ドキュメントログを設定するには、[ログ / ドキュメントログを設定する...] メニューを選択します。

トラッキング項目にフィルタをかける場合は、[フィルタ] フィールドでオプションを選択します。

ドキュメントログに表示されるトラッキング項目の最大数を指定するには、**[メモリ内の最大トラッキング項目数]**フィールドに希望の項目数を入力します。

 **注意:**

コンピュータのメモリ容量を考慮せずに表示されるトラッキング項目数を設定すると、処理時間が過大になる恐れがあります。

ドキュメント、ドキュメントのトラッキング項目と、シナリオの構成要素が残したトラッキング項目を保存するには、

- **[ファイルを使用]** オプションをオンにします。
- **[トラッキング項目を保存する]** 枠内の **[汎用名]** フィールド内に汎用名を入力します (例: acsc)。

汎用名は以下の内容を含む3つのファイルを作成します。

- シナリオに生成されるドキュメント: 「.dat」ファイル (例: acsc.dat)
- ドキュメントログ内のメッセージ (フィールドに関する警告、処理の失敗など): 「.msg」ファイル (例: acsc.msg)
- 「.dat」ファイルと「.msg」ファイル内のデータを迅速に検索するためのインデックスファイル: 「.idx」ファイル (例: acsc.idx)

 **注意:**

[汎用名] フィールドにはファイル用の完全パスまたは相対パスを入力できます。例えば、

-
-

ディスク内の記憶とメモリを削除する

[ログ] メニューには2つの削除コマンドがあります。

- **[表示されたトラッキング項目を削除する]**
このコマンドはシナリオビルダ内 (Connect-Itログとドキュメントログ) のログ項目を削除します。トラッキング項目を含むファイルは、このコマンドでは削除されません。
- **[格納されたトラッキング項目を削除する]**
トラッキング項目を保存している場合、シナリオの起動時に作成された3つのファイル (「.dat」、 「.msg」、 「.idx」) をこのコマンドで削除できます。

ドキュメントログを再び読み込む

[ログ / ドキュメントログを再び読み込む] コマンドにより、「.dat」、「.msg」と「.idx」ファイルにあるトラッキング項目全部を、ドキュメントログのタブ内に表示できます。このコマンドは、ドキュメントログ内のトラッキング項目の表示数を制限した場合に非常に有用です。この場合、このコマンドでドキュメント処理中に発生した問題点全てを表示できます。

トラッキング項目

ドキュメントログはトラッキング項目を表示します。各項目はシナリオの1構成要素に処理されたドキュメントに相当し、以下の内容を含んでいます。

- 識別番号
- ドキュメントが処理された方法を表すアイコン
- 処理の日時
- 処理されたドキュメントタイプ
- ドキュメントを生成した、または取り込んだ構成要素の名前
- ドキュメントの内容


識別番号はドキュメントの処理順に付けられています。識別番号が1のドキュメントは一番最初に処理されています。ドキュメントがメモリに保存されている場合、固有の識別番号によりドキュメントが検索しやすくなっています。

ドキュメントログの各列を昇順または降順で並べ替えるには、各列のタイトルをクリックします。

上向きの青の三角形は、並べ替えが昇順であることを示しています。下向きの三角形は並べ替えが降順であることを示しています。列が並べ替えられていない場合、三角形は灰色です。

項目の一部が切れて表示されている場合、マウスのカーソルを項目上に置くと、項目全体がヒントの形で表示されます。

ドキュメントログ内で使用されるアイコン

アイコンは最も重要な情報に対応しています。例えば、同じドキュメント内のあるフィールドは警告を受けただけなのに対し、他のフィールドが拒否されたとすると、 アイコンのみが表示されます。これは、ドキュメントの完全な拒否は警告よりも重要だと見なされるためです。

シナリオ図内では、これらのアイコンがコンポーネントの開閉を表す電球のアイコンの代わりに表示されます。アイコンは、コンポーネントが少なくとも1つのドキュメントを正常に処理しなかったこと（1フィールドに対する警告が発生した、フィールドが拒否された、またはドキュメントが完全に拒否されたこと）を示しています。

図 3.5. 処理中に起こった問題



問題発生アイコンがコンポーネント上に表示された場合、コンポーネント上を右クリックして【**トラッキング項目の表示**】をショートカットメニューから選択します。この時点で、ユーザは問題の発生を認識しているとみなされるため、アイコンは画面から消えます。

トラッキング項目のフィルタ

ドキュメントログ内に表示されるトラッキング項目には、次の条件に従ってフィルタをかけることができます。

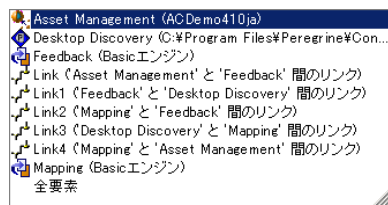
- ドキュメントを処理した構成要素
- ドキュメントの処理方法

シナリオ図内で選択されている構成要素がないと、メモリにあるトラッキング項目全体がドキュメントログに表示されます。

ある要素により処理されたドキュメントのトラッキング項目のみを表示するには、

- シナリオ図内でこの要素を選択します。または
- 【**要素**】フィールドで1項目を選択します。

図 3.6. ドキュメントログ内のトラッキング項目へのフィルタ

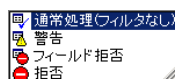


処理結果に応じてトラッキング項目を表示する場合、【フィルタ】フィールドで1項目を選択します。

選択した処理結果のトラッキング項目と、それよりも重要度の高い問題の項目が表示されます。

例：【フィールド拒否】を選択すると、フィールドを拒否されたドキュメントだけでなく、全体を拒否されたドキュメントも表示されます。

図 3.7. トラッキング項目のフィルタ



ドキュメントの詳細枠

ドキュメントの詳細枠では、ドキュメントのフィールド値についての情報と、ドキュメントがコンポーネントまたはリンクにより処理された方法を知ることができます。

ドキュメントログ内で1つのトラッキング項目が選択されると、そのドキュメントの詳細事項がログの右側にある枠内に表示されます。

ドキュメントはツリー構造で表示されます。

このツリー構造内の各ノードは以下の内容を表しています。

- ノードに対応する要素の名前
- ドキュメント内のノードの値
- ドキュメントを処理したコンポーネントまたはリンクが残すトラッキング項目

ドキュメントの詳細枠にある各トラッキング項目にはメッセージがあります。メッセージは、ドキュメント全体が拒否された理由などを説明します。

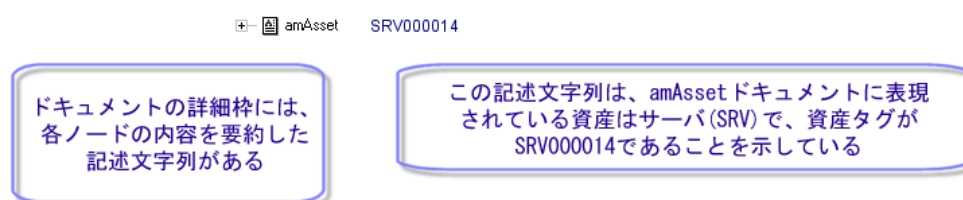
ドキュメントの詳細とドキュメントタイプの詳細

ドキュメントタイプの詳細は、生成用ドキュメントタイプ、または取り込み用ドキュメントタイプの編集時に表示されます。ドキュメントと同様に、ドキュメントタイプにはツリー構造があります。しかし、ドキュメントとドキュメントタイプの間には多くの相違点があります。

ドキュメントの詳細枠では、各ノードの値が表示されます。構造体またはコレクションの場合、この値は構造体またはコレクション内のフィールド値を要約した記述文字列です。例えば、AssetCenterの資産に対応するドキュメントのルートノードには、資産タグが記述（灰色で表示）されます。コレクションには、コレクション内の構成要素の数が表示されます。

ドキュメントタイプの詳細にはノードの名前だけが表示されます。

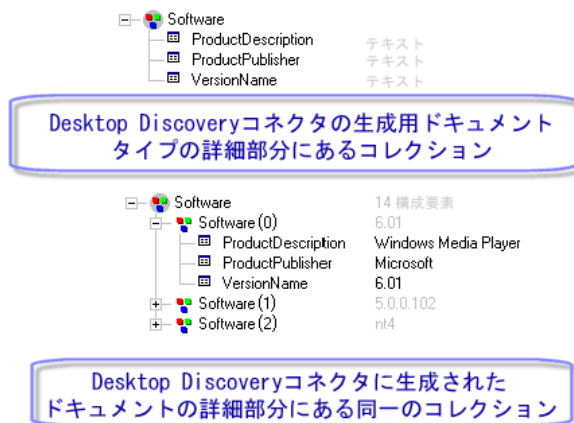
図 3.8. ドキュメント内の記述文字列



ドキュメントの詳細枠では、コレクション内の各構成要素はコレクションの子構造体として表示されます。これらの子ノードには、コレクションの名前と括弧で囲まれた数字が付きます。3構成要素のあるコレクションでは、第1構成要素には0、第2構成要素には1、第3構成要素には2が付きます。

ドキュメントタイプの詳細では、コレクションと、各フィールド値の性質（テキスト、整数、日付）のみが表示されます。

図 3.9. ドキュメントタイプ詳細内のコレクションと、ドキュメント詳細内のコレクション



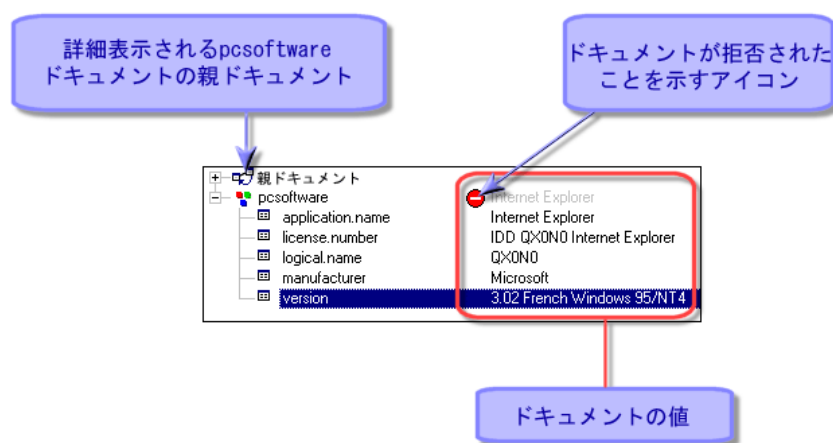
親ドキュメント

ドキュメントの詳細枠には「親ドキュメント」が含まれていることがあります。この親ドキュメントには、ドキュメント作成に使用されたデータが入っています。このため、マッピングボックスに生成されたドキュメントの詳細枠内の親ドキュメント下には、ドキュメント生成に使用されたソースドキュメントの詳細が含まれています。

例：「iddac.scn」シナリオでは、マッピングボックスに生成されたドキュメントの詳細枠内の親ドキュメントは、Desktop Discoveryコネクタに生成されたドキュメントの詳細を含んでいます。後者のドキュメントはマッピング内のソースドキュメントタイプにあたります。

マッピングボックスがドキュメントのフィールドを拒否した場合、親ドキュメントの詳細部分で、拒否されたフィールドの値と拒否の理由を確認することができます。

図 3.10. ServiceCenterコネクタに取り込まれた「pcsoftware」ドキュメントの詳細部分



XML文書（ドキュメント）の詳細とこのドキュメントのDTDをコピーする

Connect-It内で生成されたドキュメントの詳細情報はXML文書です。以下の操作が可能です。

- このXML文書をコピーする
- このXML文書のDTD（文書型定義）をコピーする
- このXML文書を開く

XML文書と、XML文書のDTDをコピーするには、

- 1 ドキュメントの詳細枠内にカーソルを置きます。
- 2 右クリックします。
- 3 表示されるショートカットメニューから、[このXML文書をコピーする]、[このXMLのDTDをコピーする]または[このXML文書を開く]を選択します。

 **注意:**

XML文書は、Connect-Itをインストールしたコンピュータで、「.xml」ファイルに関連付けられたアプリケーション上で開かれます。例えばInternet ExplorerやNetscapeなどです。


ドキュメントの詳細枠内で使用されるトラッキング項目

トラッキング項目は、シナリオのコンポーネントまたはリンクがドキュメントのフィールド上に実行した処理に関する情報です。

ドキュメントの詳細枠内では、トラッキング項目にアイコンが付きます。各トラッキング項目の内容は、別の枠に表示されるメッセージで説明されます。

トラッキング項目は2つに分類されます。

- 情報を提供するトラッキング項目（フィールドに関する重要な情報、または詳細情報）
- 処理中に発生した問題に関するトラッキング項目（フィールドに対する警告、処理の失敗）

処理エラーのトラッキング項目  のみが、ドキュメントの完全な拒否または部分的な拒否を意味します。

トラッキング項目、アイコンの意味と、項目を説明するメッセージのタイプは、以下の表の通りです。

トラッキング項目	意味	メッセージのタイプ	例
❶	詳細情報	データ処理に関する技術的な詳細情報	「INDコネクタはドキュメントに転換するデータをダウンロードします。」ファイルの大きさ (KB) やダウンロードの速度などの情報
❶	重要な情報	データ使用に重要な情報	ドキュメントの取り込み後にコネクタが作成したフォルダとファイルの名前
⚠	フィールドに対する警告	フィールドが適切に処理されなかった理由	「ServiceCenterコネクタはフィールドを取り込みましたが、ServiceCenterデータベースの構造に従ってフィールドの一部を切り詰めました。」
❶	処理の失敗	処理の失敗の理由	「予期されていないフィールド値があったためフィールドが拒否されました。ドキュメントの部分的拒否または完全な拒否が発生する可能性があります。」

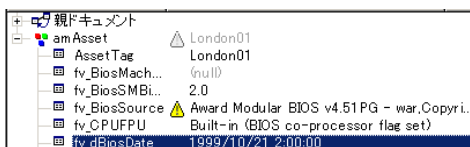
ノードの横に現れる灰色のトラッキング項目

ドキュメントの詳細枠では、トラッキング項目が親ノードの横に灰色で表示されることがあります。

灰色のトラッキング項目は、ノードの子ノードの内少なくとも1つに、トラッキング項目があることを意味します。

ノード内に複数のトラッキング項目がある場合、最も重要だと見なされるトラッキング項目が、親ノードの横に灰色で表示されます。

図3.11. ノード下にトラッキング項目があることを意味する灰色のトラッキング項目



トラッキング項目のタイプによるドキュメントのフィルタ

トラッキング項目のタイプに基づいてドキュメントの詳細にフィルタをかけると、選択したトラッキング項目タイプを含むノードのみが表示されます。

拒否されたノードのみを表示する場合は、ドキュメントの詳細枠の上にある【トラッキング項目のフィルタ】フィールドで ● を選択します。

このドロップダウンリストから1項目を選択すると、選択項目より下にあるトラッキング項目アイコンを含むノードも全て表示されます。

ドロップダウンリストでは、情報の重要度の低い項目から高い項目順に並べられています。つまり重要度の低い【詳細モード】が1番上にあり、重要度の高い【エラー】（処理の失敗）が1番下にあります。

図 3.12. ドキュメント詳細内のトラッキング項目用フィルタ




トラッキング項目の説明メッセージ

ドキュメントの詳細枠内のトラッキング項目は、メッセージで説明されます。


ドキュメントの詳細枠内のトラッキング項目を1つ選択すると、項目の内容を説明するメッセージが [メッセージ] 枠内に現れます。この枠はドキュメントログの下部にあります。

例：Asset Managementデータベースの健全性を破壊する値を含むフィールドが、Asset Managementコネクタにより拒否されたことを説明するメッセージ。


選択したノードの子ノードのメッセージを全て表示する




ドキュメント詳細の親ノード下の子ノードに含まれている全メッセージを表示するには、 をクリックします。

例

 オプションが選択されていないと、選択したノードのメッセージのみが表示されます。

 フィルタ条件 'evmap="auto devicepc" AND evttype="input"' に合うレコードはServiceCenterにはありません。

 が選択されていると、子ノードの全メッセージが表示されます。

 フィルタ条件 'evmap="auto devicepc" AND evttype="input"' に合うレコードはServiceCenterにはありません。
 イベントマップ 'auto devicepc' (イベント 'ICMdevicepc' 用)は無効です。
 'ICMdevicepc' タイプの入カイベントは作成できません。

オプション

シナリオビルダのオプションには以下のカテゴリがあります。

- アクセスビリティ
- 表示
- 確認
- コネクタ
- 文書
- ログ

シナリオビルダのオプション用ウィンドウを開くには、[編集 / オプション...] メニューを選択します。

このウィンドウでは、[値] 列からセルを選択してダブルクリックし、オプションを変更します。表示されるテキストゾーンに値を入力します。

 **注意:**

デフォルトのオプション値を変更すると、変更した値の名前は赤色になります。これにより、値を変更したオプションを素早く検索できます。

アクセシビリティ

この項目には、シナリオビルダの使用を快適にするためのオプションがまとめられています。

- 背景の画像を削除する
- フラッシュを避ける
- 色を避ける
- フォントを太字にする

注意: これらのオプションはシナリオビルダでは使用されていません。ペレグリンシステムズの他のアプリケーションで使用されています。

表示

この項目には、シナリオビルダの表示用オプションがまとめられています。

認証証明書に認証されたコネクタのみ表示する

このオプションにより、シナリオビルダのツールボックスには、認証証明書に使用を許可されたコネクタのみが表示されます。

ウィザード内の説明の色を変更する

このオプションではウィザード内の説明の色を変更できます。例えば設定ウィザード内の説明の文字の色などです。

フィールドで [Esc] キーを無効にする

Windows上では、[Esc] キーを使用すると編集用ウィンドウが閉じるようになっています。例：ソースコネクタの生成用ドキュメントタイプの編集用ウィンドウ。

シナリオビルダ内では、安全のために [Esc] キーは現在編集している要素のみに適用されます（このオプションは [いいえ] に設定されています）。例：マッピングの編集用ウィンドウでのコレクションからコレクションへのマッピングの編集。

[Esc] キーの動作をWindowsの標準的動作に戻すには、このオプションを選択してください。

Windowsグラフィックス

このオプションでは、シナリオビルダの表示をフラットまたは標準にできます。

詳細モード

このオプションを使用するとビルダは詳細モードで起動します。アプリケーションを、コマンドプロンプトを使って「-verbose」オプションで起動すると、このオプションの値は無視されます。

プレビュー時にブロック内に取得するドキュメントの数

このオプションでは、ドキュメントのプレビューウィンドウで表示するドキュメントの数を指定できます。デフォルトのドキュメント数は20です。

ドキュメントの生成テスト中に生成するドキュメントの数

このオプションでは、ドキュメントの生成テスト中に生成するドキュメントの数を指定できます。テストモードでは、シナリオのソースコネクタはドキュメントを生成しますが、ターゲットコネクタは外部アプリケーションとは相互作用しません。ドキュメント数を制限するには、チェックボックスをオンにする必要があります。

このオプションは、生成用ドキュメントタイプをテストする際に自動的に使用されます。しかし、シナリオがWindowsのサービスとして実行されると、このオプションは自動的に無効になります。

Connect-Itログ内に表示するトラッキング項目の最大数

このオプションにより、Connect-Itログに表示されるトラッキング項目の最大数を制限できます。

1回に表示する子ノードの最大数

このオプションにより、ドキュメントタイプとドキュメント内の親ノード下に表示される子ノードの数を指定できます。例：ルートノード下ではドキュメントタイプの最初の3コレクションのみを表示する。

1 ノードにつき自動的に展開する子ノードの最大数

このオプションにより、ドキュメントタイプとドキュメント内の親ノード下に表示される子ノードの数を指定できます。ドキュメントタイプのノードに多数の子ノードがある場合、このオプションは非常に有用です。

自動的にマップするレベルの最大数


このオプションでは、テーブルからテーブルへのマッピングまたはノードからノードへのマッピング時に、自動的にマップされるノード（構造体またはコレクション）のレベル数を指定できます。

起動時に [ご存知でしたか] ウィンドウを表示する

このオプションを選択すると、Connect-Itの起動時に [ご存知でしたか] ウィンドウが表示されます。このオプションは [ご存知でしたか] ウィンドウで直接オフにすることもできます（**スタートアップ時に表示する**）。

タブ

Connect-Itのタブを以下のオプションを用いて変更できます。

- **イメージの表示**
このオプションでは、タブ名の横にあるアイコンの表示 / 非表示を選択します。例： [ドキュメントログ] タブの  アイコン
- **ヒントの表示**
このオプションでは、タブの上にポインタを置くと現れるヒントの表示 / 非表示を選択します。
- **テキストの表示**
このオプションでは、タブのタイトルの表示 / 非表示を選択します。
- **タブの表示スタイル**
このオプションでは、生成用ドキュメントタイプ用ウィンドウのタブの表示スタイルを変更できます。

ヒントを表示する

このオプションを使用すると、Connect-It内でメニューやコンポーネントなどの上にカーソルを置くとヒントが表示されます。

確認

一定の操作の後に、プログラムがダイアログボックスを通じてユーザに確認を取るように設定するオプションが、このグループにまとめられています。

コレクションからコレクションへのマッピングが必要な場合に、警告を表示する

このオプションにより、コレクションからコレクションへのマッピングが必要な場合に、警告が表示されます。例：ソースコレクションのフィールドがターゲットコレクションのフィールドにマップされても、2つのコレクションがお互いにマップされていない場合、警告が表示され、2つのコレクションのマッピングが必要であることをユーザに知らせます。

一時停止モードの起動時に、メッセージを表示する

このオプションにより、一時停止モードの起動毎にメッセージが表示されます。

JVMパラメータが変更されるたびにメッセージを表示する

このオプションにより、JVMパラメータが変更されるたびにメッセージが表示されます。

コンパイルのエラーを含むスクリプトの認証を許可しない

このオプションを選択すると、シンタックスエラーを含むスクリプトの認証は許可されません。

1コネクタの始動時に現在のシナリオの全コネクタを開く

このオプションにより、1コネクタの起動時に現在のシナリオの全コネクタを開くことができます。このオプションにより、コネクタを1つずつ開くことなくシナリオのマッピングを素早く起動することが可能になり、非常に便利です。

編集するドキュメントタイプを選択するために、ダイアログボックスを使用する

このオプションにより、開かれたマッピングボックスやコネクタをダブルクリックすると、ダイアログボックスが表示され、ドキュメントタイプやマッピングを選択できるようになります。

コネクタ

以下のコネクタ用のオプションがあります。

- LDAPコネクタ
- InfraTools Network Discoveryコネクタ
- ServiceCenterコネクタ
- テキストコネクタとXMLコネクタ

その他に、全コネクタ用のオプションがあります。

- **トラッキング項目内にクエリを表示する**
- **データベース**
フィールドのデータ型がサポートされていない場合メッセージを表示する
- **処理するドキュメント数を計算する**
- **サーバとの時間差に応じて日時を調整する**
- **設定ウィザードのキャッシュ使用のオプションをデフォルトで有効にする**

LDAPコネクタ

[**ルートDSEで定義される属性を表示する**] オプションにより、サーバの現在の日付の取得を可能にする属性を、LDAPサーバが表示するかどうかを指定できます。

InfraTools Network Discoveryコネクタ

[**起動日以前のイベントのみを処理する**] オプションにより、シナリオを起動すると、起動前に起こったイベント全てを処理できます。イベントの数が多いと、シナリオが停止する恐れがあります。デフォルトではこのオプションは [**いいえ**] になっています。

ServiceCenterコネクタ

[**ServiceCenterコードページ番号を強制する**] オプションでは、自分で選択したコードページ番号を指定できます。デフォルトではこのオプション値は「0」で

す。この値は、ServiceCenterのコードページが、Connect-Itのインストールされたコンピュータのコードページであることを意味しています。

【フィルタのデータ型を確認する】オプションでは、ServiceCenterのフィールド値タイプが、要求の作成者に指定されたタイプに一致するかどうかを調べることができます。例えば、日付型のフィールドが、整数に比較されている「sysmodtime > 3」フィルタは無効です。この場合エラーメッセージが表示され、コネクタはドキュメントを生成しません。

【コネクタが使用可能なドキュメントタイプを発行する間、進行状況を表示する】オプションを使うと、開いたコネクタが発行する使用可能なドキュメントタイプの名前を、Connect-Itログ内に表示することができます。

データベース

【フィールドのデータ型がサポートされていない場合メッセージを表示する】オプションを使用すると、フィールドがコネクタによってサポートされていない場合、ドキュメントログ内にメッセージが表示されます。

XMLと区切られたテキスト

【Connect-Itログ内に処理中のURLを表示する】オプションを選択すると、XMLコネクタと区切られたテキストコネクタが処理しているURLを、Connect-Itログ内に表示することができます。

トラッキング項目内にクエリを表示する

このオプションにより、ドキュメントログ内で（メッセージ枠内）クエリのテキストを表示できます。これはシナリオのテスト段階やデバッグ中に便利です。

処理するドキュメント数を計算する

このオプションにより、シナリオビルダのステータスバーに、処理されるドキュメント全体数の内いくつかのドキュメントが処理されたかが表示されます。例えば、200の内4個のドキュメントが処理された場合、「4/200」がステータスバーに現れます。

警告:

処理するドキュメント数が多大な場合にこのオプションを使用すると、コンピュータの性能が低下する可能性があります。

サーバとの時間差に応じて日時を調整する

このオプションがオンになっていると、コネクタの設定時にサーバとの時間差を指定するたびに、読み取られたり書き込まれたりする日時は時間差の値に応じて調整されます。

注意：時間差の調整はConnect-It 2.7.1までは自動的に実行されていました。

サーバとの時間差のオプションについては、「コネクタの設定」章の「サーバとの時間差を指定する」の節を参照してください。

設定ウィザードのキャッシュ使用のオプションをデフォルトで有効にする

このオプションにより、コネクタでキャッシュファイルを使用するために、コネクタの設定ウィザードでキャッシュの使用を手動で選択する必要がなくなります。

キャッシュオプションについては、「コネクタの設定」章の「キャッシュを設定する」の節を参照してください。

文書

注意：この項目の文書は、コネクタに処理されるドキュメントではなく、シナリオを含むSCNファイルのことを指しています。

この項目は、SCNファイルの読み込みに関するオプションをまとめています。

- **[ファイル]メニューに格納する、最近開いた文書の最大数**
このオプションは、[ファイル]メニューのリストに表示されるSCNファイルに関するものです。
- **起動時に前回最後に使用した文書を自動的に読み込む**
- **保存前に確認を要求する**
このオプションにより、Connect-Itで保存を実行する前にダイアログボックスが表示されます（マッピング、シナリオ、スケジューラなどの保存）。

ログ

ログ用には[ログファイルのサイズ(MB)]オプションがあり、「.log」ファイルの大きさを指定できます。サイズの最大値に達するたびに、新規データは最も古いデータを削除して上書き保存されます。

4 | 統合シナリオのインプリメン テーション

統合シナリオのインプリメンテーションでは以下の操作を実行します。

- コンポーネントを選択する。
- コンポーネント同士をリンクする。


シナリオインプリメンテーションウィザードを使用する

シナリオインプリメンテーションウィザードにより以下の操作を実行できます。

- シナリオのソースコネクタとターゲットコネクタを選択する。
- 2コネクタの設定を実行する。

ターゲットコネクタの設定が終了すると、2つのコネクタはマッピングボックスにリンクされます。コネクタとマッピングボックスは、シナリオビルダのシナリオ図枠内に表示されます。

以下の操作を行うとウィザードが起動します。

- シナリオビルダのツールバーで  をクリックします。または、
- [ファイル / 新規作成] メニューを選択します。

シナリオの手動インプリメンテーション

シナリオを手動で作成するには、

- 1 シナリオ図内にコンポーネントを配置します。
コネクタをシナリオ図内に置くと、自動的に **[コネクタの設定]** ウィザードが起動します。
- 2 コンポーネントをリンクします。

シナリオ図内にコンポーネントを配置する

シナリオ図内にコンポーネントを配置するには、

- 1 ツールボックスからシナリオ図へコンポーネントをドラッグするか、またはコンポーネントをダブルクリックします。
- 2 表示される **[コネクタの設定]** ウィザードページに値を入力します。

コンポーネントを設定し直すには、

- 1 シナリオ図内でコンポーネントを選択します。
- 2 以下の操作の内1つを実行します。
 - **[ツール / 設定]** メニューを選択します。
 - 右クリックし、表示されるショートカットメニューから **[コネクタを設定する...]** を選択します。

図 4.1. シナリオビルダのツールボックス



コンポーネントをリンクする

コネクタをリンクする方法により、どのコネクタがソースコネクタまたはターゲットコネクタになるかを定義できます。

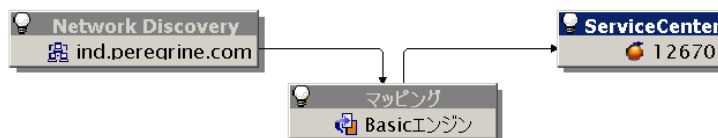
- コネクタが外にリンクを送信する場合、このコネクタはソースコネクタになります。
- コネクタがリンクを受信する場合、このコネクタはターゲットコネクタです。

注意:

ピボットドキュメントタイプを使用しない場合、ソースコネクタをターゲットコネクタに直接リンクすることは稀です。原則としてソースコネクタをマッピングボックスに連結し、マッピングボックスをターゲットコネクタに連結します。

以下の例では、Network Discoveryコネクタはソースコネクタで、ServiceCenterコネクタはターゲットコネクタです。Network Discoveryコネクタがドキュメントを生成すると、ドキュメントはまずマッピングボックスで変換され、その後ServiceCenterコネクタに取り込まれます。

図 4.2. シナリオのコンポーネントのリンク



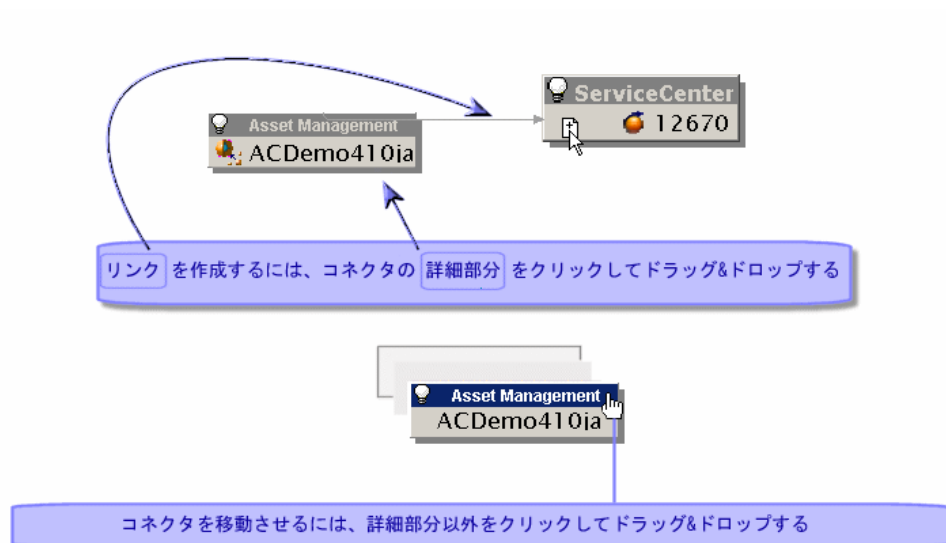
コンポーネントを別のコンポーネントにリンクするには、

- 1 コンポーネント下部の詳細部分（カーソルを置くと文字が白に変わります）をクリックします。
- 2 マウスの左ボタンを押したままにします。
- 3 表示されるリンクを別のコンポーネントへドラッグします。

注意:

コンポーネントの詳細部分以外をクリックしてドラッグすると、コンポーネントをシナリオ図内で移動させることになります。

図 4.3. リンクの作成またはコネクタの移動



マッピングボックスを通さずに2コネクタをリンクする

2コネクタをリンクすると、Connect-Itは自動的に2コネクタ間にマッピングボックスを作成します。マッピングボックスを作成しない場合は、【Shift】キーを押したまま2コネクタをリンクします。

2コネクタ間の直接リンクを使用するのは、テキストコネクタとXMLコネクタのみです。直接リンクにより、ソースコネクタに生成されるドキュメントの構造を利用して、ソースコネクタのデータを直接エクスポートすることができます。

コンポーネントを削除する

シナリオ図内のコネクタ、リンクまたはマッピングボックスを削除するには、

- 1 コンポーネントを選択します。
- 2 以下の操作の内1つを実行します。
 - 【編集 / 削除】メニューを選択します。
 - キーボードの【Del】キーを押します。
 - 右クリックし、ショートカットメニューから【削除】を選択します。

生成用または取り込み用ドキュメントタイプ

生成用または取り込み用ドキュメントタイプはテンプレートの役目を果たしており、コネクタはシナリオの実行時に、このテンプレートを使用してドキュメントを生成したり取り込んだりします。生成用または取り込み用ドキュメントタイプは、コネクタが発行する使用可能なドキュメントタイプから選択されたフィールドの集まりです。

例：

Desktop Discovery - Asset Managementシナリオ (iddac.scn) には、Asset Managementコネクタの取り込み用ドキュメントタイプ「amAsset」があります。この取り込み用ドキュメントタイプは、Asset Managementコネクタが発行する使用可能なドキュメントタイプ「amAsset」内の全フィールドから選択された一部のフィールド群です。

Connect-Itで処理されるドキュメントタイプとドキュメントの関係は、次の表の通りです。

使用可能なドキュメントタイプ	生成用または取り込み用ドキュメントタイプ	生成されるドキュメント
----------------	----------------------	-------------

amAsset	
AcctCode	テキスト
AssetTag	テキスト
BarCode	テキスト
bCreatedOnThe...	整数(16ビット)
bIpxGpxInstalled	整数(16ビット)
bIsCnxClient	整数(16ビット)
bNetBeuInstalled	整数(16ビット)
Brand	テキスト
bTcplnInstalled	整数(16ビット)
Comment	メモフィールド

amAsset	
AssetTag	
Brand	
mPrice	
Category	
Name	

amAsset		Hewlett Packard
AssetTag		PRNLND1006
Brand		Hewlett Packard
Category		C000007
BarCode		C000007
binvent		1

設定

コネクタは全部の使用可能なドキュメントタイプを発行します。ユーザはこの内1つの使用可能なドキュメントタイプを選択します。

ドキュメントタイプの作成

生成用、または取り込み用ドキュメントタイプを作成するには、使用可能なドキュメントタイプから必要な要素を選択します。

シナリオの起動

コネクタは、生成用または取り込み用ドキュメントタイプに沿って、ドキュメントの生成または取り込みを実行します。

コネクタの生成用ドキュメントタイプ、または取り込み用ドキュメントタイプを作成するには、

- コネクタが発行する使用可能なドキュメントタイプから、1つのドキュメントタイプを選択します。

- 選択したドキュメントタイプから必要なフィールドを選択します。

例：

Asset Managementコネクタが発行する全部の使用可能なドキュメントタイプから、「amProduct」ドキュメントタイプを選択します。次に、必要なフィールド（バーコード、メーカー、価格、カタログ番号など）を選択します。

コネクタの生成用または取り込み用ドキュメントタイプを編集するには、



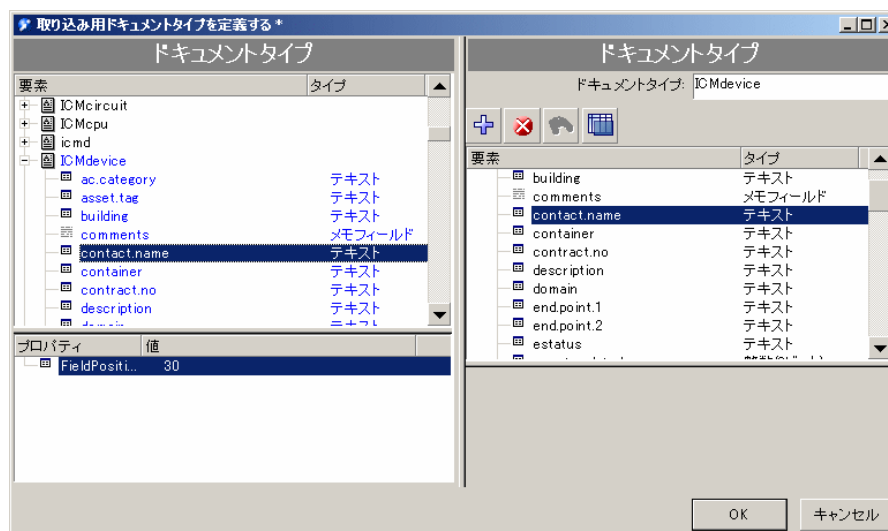
- 1 シナリオ図部分でコネクタを選択します。
- 2 **[ドキュメントタイプ]**タブを選択するか、またはコネクタをダブルクリックします。
- 3 生成用ドキュメントタイプ、または取り込み用ドキュメントタイプの枠を選択します。
- 4  をクリックすると、ドキュメントタイプを新規に作成できます。 をクリックすると、選択した既存のドキュメントタイプを編集できます。


図 4.4. ServiceCenterコネクタの取り込み用ドキュメントタイプの編集



編集用ウィンドウは2つのゾーンから成っています。

- 生成用または取り込み用ドキュメントタイプを作成するための、画面右側の作業枠

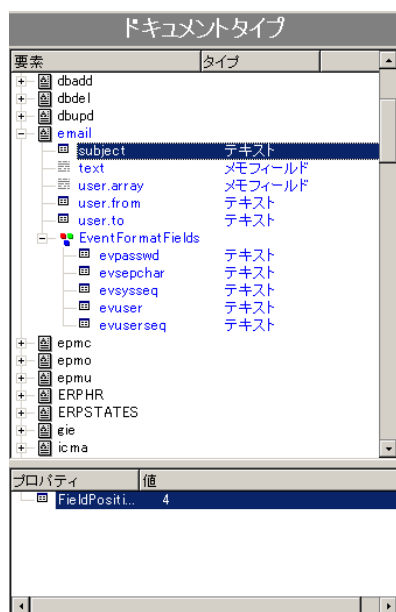
この作業枠の下にはルール（ディレティブ）用のタブが表示されます。ディレティブのないコネクタもあります。

- コネクタが発行する使用可能なドキュメントタイプが表示される画面左側のゾーン。このゾーンは2つの部分に分かれています。
 - 使用可能なドキュメントタイプのルートノード（）が表示される枠。この枠内でドキュメントタイプのノード（ルートノード、構造体、コレクション）内を全部表示すると、ノード内の全要素を参照できます。各ノードと各フィールド（端末ノード）には要素の名前とタイプ（テキスト、整数、日付...）が表示されます。ドキュメントタイプ内で1ノードを選択すると、このノードは現在の選択項目になります。
 - 現在の選択項目のプロパティと値を表示する枠。これらの技術的情報は、選択項目の意味やプロパティを理解するのに利用されます。

例：


以下の画面では、「email」ドキュメントタイプの「subject」フィールドのタイプは「テキスト」です。

図 4.5. 使用可能なドキュメントタイプのゾーン



生成用または取り込み用ドキュメントタイプを作成する


生成用または取り込み用ドキュメントタイプを作成するには、ルートノードまたはノード内の要素（コレクション、構造体、フィールド）を右側の作業枠内に移動させます。使用可能なドキュメントタイプから要素を移動させるには、以下の方法があります。

- 選択した要素をドラッグ&ドロップします。または、
- 必要な要素を選択し、 をクリックします。または、
- 選択した要素をダブルクリックします。

注意:

ノード（ルートノード、構造体、コレクション）下にある多数のフィールドを使用する場合は、ノードを直接右の作業枠に移動させる方法を取ると便利です。ノード下の全フィールドも自動的に作業枠へ移るので、ここから不必要なフィールドを削除してドキュメントタイプを作成します。

作業枠から要素を削除する方法

- 1 要素を選択し  をクリックするかまたは [Del] キーを押します。
- 2 要素を画面左側の使用可能なドキュメントタイプ枠へドラッグして戻します。

要素の選択が終了したら、ドキュメントタイプに名前を付けます。（デフォルトでは、画面左から右の作業枠に移した使用可能なドキュメントタイプのノードの名前が付いています。） [OK] をクリックします。

注意:

マッピングボックスでマッピングを作成すると、マッピング内で使用されたソースコネクタの使用可能なドキュメントタイプの要素は、生成用ドキュメントタイプに自動的に追加されます。ドキュメントタイプの編集は以下の場合に便利です。

- マッピングで使用されなくなった要素をドキュメントタイプから削除する場合
- マッピングで使用されていない要素の内、必要な要素を追加する場合。例えば、シナリオの実行時に InfraTools Network Discovery コネクタのスケジュールのポイントを更新するために、ソース内で取得されなければならない要素などです。

生成用または取り込み用ドキュメントタイプのDTDをコピーする

Connect-It内の各ドキュメントタイプは、それぞれXML形式のドキュメントに対応しています。このドキュメントのDTD（つまりConnect-Itでの「ドキュメントタイプ」）をコピーすることができます。

生成用または取り込み用ドキュメントタイプのDTDをコピーする方法


- 1 生成用ドキュメントタイプを作成または編集します。
- 2 カーソルを作業枠（画面右）に置きます。
- 3 右クリックします。
- 4 **[このXMLのDTDをコピーする]**をショートカットメニューから選択します。

ドキュメントタイプの編集用ウィンドウが開いている場合、右クリックすると表示されるショートカットメニューで、DTDをクリップボードへコピーできます。

生成用ドキュメントタイプのデータを参照する

Connect-Itでは、生成用ドキュメントタイプの要素に対応するデータを表示できます。表示されるのは、コネクタの接続先の外部アプリケーション内に記録されているデータです。

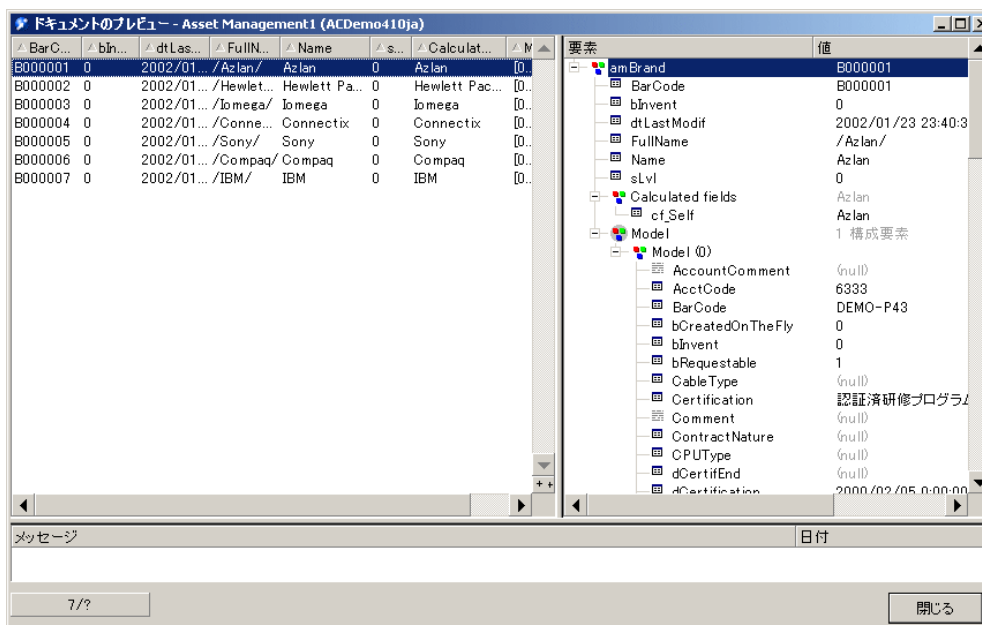
注意:

データの表示に一定以上の時間がかかると、進行中のアクションを中断するためのウィンドウが自動的に開きます。表示の中断を選択すると、既に検索されたデータにアクセスできるようになります。をクリックすると、中断したアクションを再開できます。

例:

データベースコネクタの生成用ドキュメントタイプのデータは、データベースの1つのテーブルにある複数のレコードに一致します。

図 4.6. 生成用ドキュメントタイプのデータプレビュー用ウィンドウ



プレビュー用ウィンドウ内に読み込まれるドキュメントの数は、[編集/オプション]メニューの表示項目内の[プレビュー時にブロック内に取得するドキュメントの数]オプションで指定されます。デフォルトでは、最初の20のドキュメントが取得されます。次のドキュメントを読み込むには、 ボタンを押します。全ドキュメントが表示されると、 ボタンは灰色になります。

生成用ドキュメントタイプを表示する方法

- 1 生成用ドキュメントタイプを作成または編集します。
- 2 以下の操作の内1つを実行します。
 - をクリックします。
 - カーソルを作業枠上に置いて右クリックし、ショートカットメニューから[データを表示する...]を選択します。

ドキュメントタイプのデータのプレビュー用ウィンドウは、2つの部分から成っています。

- 画面左側の枠には、ドキュメントタイプの要素に対応する外部アプリケーションの全レコードが表示されます。
画面左の各列は、生成用ドキュメントタイプの各要素に対応します。

- 画面右側の枠には、生成用ドキュメントタイプと、左側の枠で選択したレコードの値が表示されます。

ドキュメントタイプ編集用ウィンドウでのショートカットメニュー

ドキュメントタイプ編集用ウィンドウには、2種類のショートカットメニューがあります。

- 使用可能なドキュメントタイプのゾーン（画面左側）にカーソルが位置する時のメニュー
- 作業枠にカーソルが位置する時のメニュー

使用可能なドキュメントタイプのゾーン（画面左側）にカーソルが位置する時のメニュー

コマンド	機能
このXMLのDTDをコピーする	コンピュータのクリップボードに、トラッキング項目に対応するXMLのDTD（文書型定義）をコピーします。
パスをコピーする	コンピュータのクリップボードに、選択した要素のパスをコピーします。
この要素を追加する	作成中の生成用または取り込み用ドキュメントタイプに、選択した要素を追加します。
使用されていないドキュメントタイプにフィルタを適用する	作業枠内には、編集される生成用または取り込み用ドキュメントタイプのみが表示されます。

作業枠にカーソルが位置する時のメニュー

コマンド	機能
このXMLのDTDをコピーする	コンピュータのクリップボードに、トラッキング項目に対応するXMLのDTD（文書型定義）をコピーします。
パスをコピーする	コンピュータのクリップボードに、選択した要素のパスをコピーします。
この要素を削除する	作成中の生成用または取り込み用ドキュメントタイプから、選択した要素を削除します。
データを表示する	生成用ドキュメントタイプの要素に対応するデータを表示します。
使用されていないドキュメントタイプにフィルタを適用する	作業枠内には、編集される生成用または取り込み用ドキュメントタイプのみが表示されます。

5 | ドキュメントタイプのマッピング

ドキュメントタイプをマップすると、ソースコネクタが生成したドキュメントをターゲットコネクタが取り込めるようになります。Connect-Itでは、2つのコネクタ間に位置するマッピングボックスで、生成用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間のマッピングを編集できます。

マッピングを作成するには、ソースドキュメントタイプの要素と、ターゲットドキュメントタイプの要素を関連付ける必要があります。

本章の読解を容易にするために、**ソース要素**や**ターゲット要素**、更にソースまたはターゲットのフィールド、構造体、コレクションの概念を本章で用います。



ソースドキュメントタイプの要素を、ターゲットドキュメントタイプの要素に直接関連付けられない場合は、Basicスクリプトを作成できます。Basicスクリプトは以下の要素を使用します。

- ソースドキュメントタイプの複数の要素
- 固定値
- グローバル関数と変数
- マップテーブル
- 文字列ファイル

マッピングボックス

マッピングボックスでは、ソースコネクタの生成用ドキュメントタイプと、ターゲットコネクタの取り込み用ドキュメントタイプの間のマッピングを編集できます。

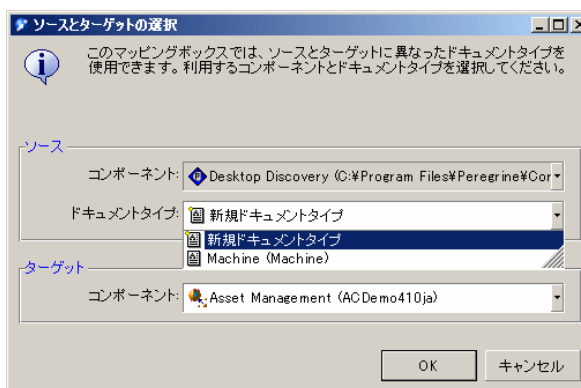
マッピングの編集用ウィンドウを開くには、

- 1 シナリオ図内でマッピングボックスを選択します。
- 2 以下の操作の内1つを実行します。
 - **[マッピング]** タブを選択します。
 - ダブルクリックします。
ダブルクリックすると、**[マッピング]** タブと同じ内容のダイアログボックスが表示されます。
- 3  をクリックすると、マッピングを新規に作成できます。 をクリックすると、選択した既存のマッピングを編集できます。

マッピングを新規に作成すると、**[ソースとターゲットの選択]** ウィンドウが表示されます。ここでは以下の要素を選択します。

- ソースコンポーネント
- ソースドキュメントタイプ
- ターゲットコンポーネント

図 5.1. マッピング - コンポーネントとソースドキュメントタイプの選択用ウィンドウ



 **注意:**

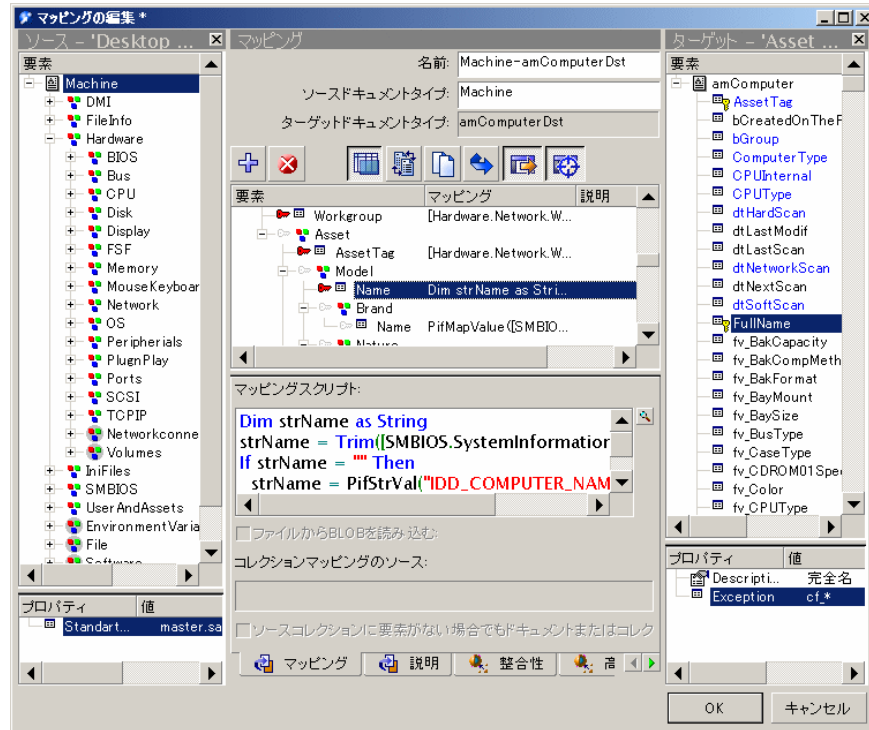
次の場合このダイアログボックスは表示されません。

- マッピングボックスが、1つのソースコンポーネントと1つのターゲットコンポーネントのみにリンクされている場合
- ソースコンポーネント用に、生成用ドキュメントタイプが作成されていない場合









マッピングウィンドウは3つのゾーンに分かれています。

- 選択されたソースコンポーネントが発行する全ての使用可能なドキュメントタイプが表示される部分（画面左）
[ソースとターゲットの選択] ボックスで既存の生成用ドキュメントタイプを選択すると、このドキュメントタイプのルートノードの全レベルが表示されます。
- 選択されたターゲットコンポーネントが発行する全ての使用可能なドキュメントタイプが表示される部分（画面右）
- マッピング用ゾーン（画面中央）。ここには以下のゾーンがあります。
 - ソース要素とターゲット要素をマップするための作業枠
 - ターゲットコンポーネントのルール（ディレクティブ）用タブと、マッピングの各ノードを説明するためのタブが現れる部分

図 5.2. マッピング編集用ウィンドウ

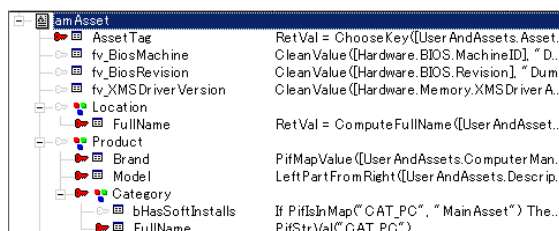


マッピングボックスのツールバーでは、頻繁に行う操作を実行できます。

アイコン	機能
	作業枠内に要素を置きます。
	作業枠内から選択した要素を取り除きます。
	ターゲット要素をソース要素に関連付けます。
	マッピングで使用されていないソースとターゲットドキュメントタイプに、フィルタを適用します。
	作業枠内にあるターゲット要素を複製します。
	マップされたターゲット要素を探します。
	ソースコネクタの使用可能なドキュメントタイプを表示 / 非表示にします。
	ターゲットコネクタの使用可能なドキュメントタイプを表示 / 非表示にします。

マッピングウィンドウの作業枠には、ソース要素とターゲット要素のマッピングの詳細が表示されます。スクリプトが使用されていると、スクリプトの関連付けられているターゲット要素の横に要約された形で表示されます。


図 5.3. 作業枠上のマッピングスクリプトの要約



マッピングボックスの設定

Connect-Itのマッピングボックスの設定はウィザードで実行できます。マッピングボックスの設定ウィザードを起動するには、以下の方法があります。

- マッピングボックスがシナリオ図内に位置しない場合
 - ツールボックス内のマッピングボックスをダブルクリックします。
 - ツールボックスからシナリオ図へマッピングボックスをドラッグします。
- マッピングボックスがシナリオ図内に位置する場合

- 1 マッピングボックスを選択してから **[ツール/設定]** を選択します。
- 2 マッピングボックスを選択し **[F2]** キーを押します。
- 3 マッピングボックスを選択し、右クリックしてショートカットメニューから **[コネクタを設定する]** を選択します。
- 4  をクリックします。

マッピングボックスの設定では、マッピングに名前を付け説明を加えることができます。この設定は必須ではありません。

マッピングボックスに名前を付け、説明を加える

設定ウィザードの1ページ目では、以下の内容を指定できます。

- マッピングボックスの名前
- マッピングボックスの役割の説明

[名前]

マッピングボックスに名前を付けます。フィールドのデフォルト値は「Mapping」です。1つの同じシナリオ内でこの名前は固有の名前でなければなりません。


[説明]

コネクタの役割を説明するテキストを作成します。

マッピングの編集

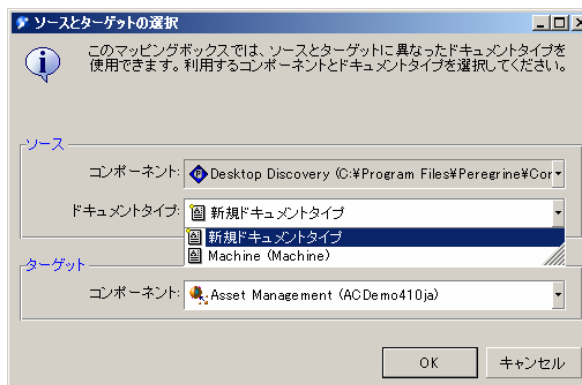
本節では、統合シナリオのマッピングを編集する様々な方法を説明します。

マッピングの作成方法

- 1 シナリオ図内でマッピングボックスを選択します。
- 2 以下の操作の内1つを実行します。
 - **[マッピング]** タブを選択します。
 - ダブルクリックします。
ダブルクリックすると、**[マッピング]** タブと同じ内容のダイアログボックスが表示されます。
- 3  をクリックします。
マッピングを新規に作成すると、**[ソースとターゲットの選択]** ダイアログボックスが表示されます。ここでは以下の要素を選択します。

- ソースコンポーネント
- ソースドキュメントタイプ
- ターゲットコンポーネント

図 5.4. コンポーネントとソースドキュメントタイプの選択用ボックス

**注意:**

次の場合このダイアログボックスは表示されません。

- マッピングボックスが、1つのソースコンポーネントと1つのターゲットコンポーネントのみにリンクされている場合
- ソースコンポーネント用に、生成用ドキュメントタイプが作成されていない場合

マッピングボックスには、ソースコネクタとターゲットコネクタが発行する全ての使用可能なドキュメントタイプが表示されています。このため、マッピング編集用ウィンドウでマッピングを作成すると（つまりソース要素とターゲット要素を、マッピングボックスで関連付けると）、ソースコネクタの生成用ドキュメントタイプとターゲットコネクタの取り込み用ドキュメントタイプを、マッピングと同時に自動的に作成することができます。

 **注意:**

1つのマッピング内で、1つのソースドキュメントタイプの要素を、複数のターゲットドキュメントタイプの要素に関連付けることはできません。またその逆（1つのターゲットドキュメントと複数のソースドキュメント）も不可能です。そのため、マッピングの編集中にドキュメントタイプの1つを変更すると、新規のマッピングを作成することになり、以前に作成したマッピング情報は失われます。1つのソース要素を複数のターゲットドキュメントタイプの要素に関連付けるには、複数のマッピングを作成し、各マッピングに同一のソースドキュメントを使用します。


異種のマッピング用に同じソースドキュメントを使用することにより、ソースコネクタ用に行うクエリの数を最小限に抑えることができます。

また、1つのターゲットドキュメントタイプを選択して、そのドキュメントタイプ内の1つのノードを作業枠内にドラッグした後、ソースドキュメントタイプに関連付けないでおくことも可能です。このノード下にあるフィールドは作業枠内に表示されます。


 **注意:**

ソース要素またはターゲット要素上に表示される **[データを表示する]** ショートカットメニューを使用すると、ソースデータやターゲットデータを表示できます。

既存マッピングの編集方法

- 1 シナリオ図内でマッピングボックスを選択します。
- 2 以下の操作の内1つを実行します。
 - **[マッピング]** タブを選択します。
 - ダブルクリックします。
ダブルクリックすると、**[マッピング]** タブと同じ内容のダイアログボックスが表示されます。
- 3 以下の操作の内1つを実行します。
 -  をクリックします。
 - 編集するマッピングをダブルクリックします。

マッピングの削除方法

- 1 シナリオ図内でマッピングボックスを選択します。
- 2 以下の操作の内1つを実行します。
 - **[マッピング]** タブを選択します。
 - ダブルクリックします。
ダブルクリックすると、**[マッピング]** タブと同じ内容のダイアログボックスが表示されます。
- 3  をクリックします。

マッピング説明の記述方法

シナリオビルダではマッピングの各ノードに説明を加えることができます。

例：構造体またはコレクション内に要素が存在するかどうかをテストするために使用されるマッピングスクリプトの説明

- 1 マッピングを編集します。
- 2 マッピングのノードを選択します。
- 3 作業枠の **[説明]** タブを選択します。
- 4 説明を加えます。
- 5 マッピングを保存します。
説明はSCNファイル内に保存されます。

複数のマッピングの並べ替え

マッピングは、ソースコネクタの生成用ドキュメントタイプを基準にして並べられています。複数のマッピングが同じドキュメントタイプに関連する場合、マッピングを並べ替えると、ドキュメントの生成時どの順番でマッピングが実行されるかを指定できます。場合によっては、最初のマッピングはターゲットアプリケーション内にレコードを作成し、2番目以降のマッピングはこのレコードを更新する、というようにも設定できます。

- 1 マッピングを選択します。
- 2 ▲または▼をクリックして、マッピングを上下に移動させます。

ドキュメント生成とマッピング実行の順番

以下の規則が適用されます。

- テストモード

適用されるのは、**【生成用ドキュメントタイプ】**タブ内に表示されている、ソースコネクタのドキュメント生成の順番です。

- **スケジュールモード**

適用されるのは、スケジュールの編集用ウィンドウ内で定義されているドキュメントの順番です。このウィンドウでは、1つのスケジューラが複数のドキュメントタイプに関連付けられることがあります。このため、複数のドキュメントタイプがスケジューラに関連付けられる順番が適用されます。この場合、コネクタの**【生成用ドキュメントタイプ】**タブ内で指定される順番は、考慮されません。


ソースとターゲットのデータをプレビューする

マッピングの編集用ウィンドウでは、ソースアプリケーションとターゲットアプリケーションのデータを、プレビュー表示することができます。注意：ターゲットデータのプレビューは一部のコネクタでのみ可能です。

- 1 マッピングを編集します。
- 2 ソースまたはターゲットドキュメントタイプのウィンドウ内に、カーソルを置きます。
- 3 右クリックします。
- 4 **【データを表示する】**ショートカットメニューを選択します。
データのプレビュー用ウィンドウが表示されます。

データのプレビューを編集する

編集用ウィンドウでマッピングのソースドキュメントタイプを編集し、選択した一部のデータのみをプレビューすることも可能です。

- 1 マッピングを編集します。
- 2 ソースドキュメントタイプのウィンドウにカーソルを置きます。
- 3 **【データベースのプレビュー用にドキュメントタイプを編集する】**を選択します。
- 4 マッピングのソースドキュメントタイプの編集用ウィンドウが表示されます。
- 5 作業枠で要素を追加または削除します。
- 6  をクリックします。
- 7 ソースデータをプレビュー表示します。
- 8 **【閉じる】**をクリックして、マッピングの編集用ウィンドウに戻ります。

重要：ソースドキュメントタイプを編集するためには、この機能を使用しないでください。

ソースドキュメントタイプに加えられた変更事項は、プレビューウィンドウを閉じると同時に失われます。

マッピングを非アクティブにする

マッピングを非アクティブにすると、コネクタがこのマッピング用にドキュメントを生成するのを妨ぐことができます。例えばあるマッピングは、ソースアプリケーション内で除去された資産に対応するレコードを、ターゲットアプリケーション内で削除できるとします。レコードを1度だけ削除する場合は、レコードの削除が終わった時点でこのマッピングを非アクティブにします。

マッピングを非アクティブにする場合は、**[マッピング]**タブ内で、非アクティブにするマッピングのチェックボックスをオフにします。

マッピングをコピー / 貼り付けする

この機能ではまず、ソースドキュメントタイプからターゲットドキュメントタイプへのマッピング全体、またはマッピングの一部を、クリップボードにコピーします。次に、クリップボードの内容を別のマッピングに追加または結合します。この追加（または結合）先のマッピングでは、コピーされたマッピングのターゲットドキュメントが使用可能である必要があります。（例：Asset Managementコネクタを含むシナリオ内で、ターゲットドキュメントタイプがamAssetになっているマッピングの一部をコピーします。そしてこのコピー内容を、ターゲットドキュメントタイプが同様にamAssetである別のマッピングに貼り付けます。）

この機能は、同一のマッピングまたは類似するマッピングを多数作成する場合に便利です。

- 1 既存シナリオのマッピングを開きます。
- 2 マッピングの作業枠内で、ターゲットドキュメントタイプの以下の要素上にカーソルを置きます。
 - ルートノード
 - 構造体
 - コレクション
 - フィールド
- 3 右クリックします。
- 4 **[マッピングをコピーする]**を、表示されるショートカットメニューから選択します。
- 5 コピーされたマッピングのターゲットドキュメントタイプを含む別のマッピングを開きます（ターゲットドキュメントタイプは、ルートノード、構造体またはコレクションとして含まれています）。

- 6 作業枠内で、マッピングを追加または結合する場所（ルートノード、構造体、コレクションなど）にカーソルを置きます。
- 7 右クリックして、ショートカットメニューから **[マッピングを貼り付ける]** を選択します。
- 8 以下の操作を実行します。
 - **[はい]** をクリックするとマッピングを結合できます。作業枠内で選択された要素は、クリップボード内のマッピングにより変更されます（2つのマッピングが結合し、1つになります）。
 - **[いいえ]** をクリックするとマッピングを追加できます。クリップボード内のマッピングは、既存のマッピングに新規の構造体またはコレクションの形で追加されます。

ターゲット要素をソース要素にマップする

ターゲット要素をソース要素にマップするには4つの方法があります。

- フィールドからフィールドへの直接マッピング
- コレクションからコレクションへのマッピング
- コレクションからドキュメントへのマッピング
- フィールドからコレクションへのマッピング

最後の3つのマッピングは、最終的にフィールドからフィールドへのマッピングを実行してから終了します（例えば、コレクションからコレクションへマップされた後、コレクション内のフィールド同士がマップされます）。


フィールドからフィールドへの直接マッピング

フィールドからフィールドへの直接マッピングを作成するには、ソースフィールドをターゲットフィールドへマップします。データが処理される時、ソースフィールドの値が、マッピングボックスに生成されるドキュメントのターゲットフィールドに入力されます。この操作はBasicエンジンの介入なしに実行されます。

例：

ソースドキュメントタイプの「Name」フィールドは、ターゲットドキュメントタイプの「Name」フィールドに関連付けられます。

フィールドからフィールドへの直接マッピング作成には、2つの方法があります。

- 1 ドラッグ&ドロップによる方法
 - ソースフィールド（またはターゲットフィールド）を選択します。
 - フィールドをターゲットフィールド（またはソースフィールド）にドラッグします。
- 2 キーボードによる方法
 - 関連付ける2つのフィールドを、画面のソースゾーンとターゲットゾーンで選択します。
 -  をクリックします。
- 3 2段階のドラッグ&ドロップ方法
 - ターゲットフィールドを選択します。
 - 作業枠にフィールドをドラッグします。
 - ソースフィールドを選択します。
 - 作業枠内のターゲットフィールドと同じ行へソースフィールドをドラッグします。
- 4 ドラッグ&ドロップと、マッピングスクリプトを使用する方法
 - ターゲットフィールドを選択します。
 - 作業枠にフィールドをドラッグします。
 - **[マッピングスクリプト]**フィールドに、直接ソースフィールド名を入力します。

フィールドからフィールドへの直接マッピングをすると、ソース要素名は自動的に**[マッピングスクリプト]**フィールドに表示されます。手動でソース要素名を入力する場合は、要素名を角括弧（`[]`）で囲む必要があります。

図 5.5. マッピングスクリプト内のソース要素



マッピングスクリプト
`[Software.VersionName]`

コレクションからコレクションへのマッピング

コレクションは要素（フィールド、構造体、コレクションまたは属性）から成り立っており、要素の数は不定です。

コレクション内の不特定の要素群は、コレクションの構成要素です。

Desktop Discoveryコネクタが発行する、使用可能なドキュメントタイプ「Machine」では、あるコンピュータにインストールされたソフトウェアはコレクションで示されています。コレクションの各構成要素はソフトウェアの内容を説明しています（名前、ソフトウェア会社、バージョンなど）。

図 5.6. コレクション



ソースコレクションをターゲットコレクションへマップすると、Connect-Itはデータの処理時に、ソースコレクションにある構成要素の数を計算し、ターゲットコレクション内に同数の構成要素を作成します。

例：

Desktop Discovery - AssetCenterシナリオでは、ドキュメントタイプ「Machine」の「Software」コレクション内の要素は、「amAsset」ドキュメントタイプの「Softinstall」（インストール済みソフト）コレクション内の要素へ関連付けられています。

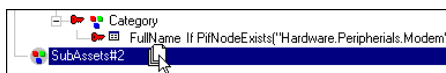
コレクションからコレクションへのマッピング作成には、2つの方法があります。

- ドラッグ&ドロップによる方法
- キーボードによる方法

ドラッグ&ドロップによる方法

- 1 ターゲットコレクションを作業枠内にドラッグします。

- 2 ソースコレクションを選択し、これを作業枠内のターゲットコレクションと同じ行にドラッグします。ドラッグする時には、マウスの左ボタンとキーボードの「Ctrl」キーを同時に押したままにします。
- 3 マウスポインタ下に3重の文書の形をしたアイコン（コレクション構成要素の重複を表しています）が表示されたら、マウスの左ボタンを放します。



【コレクションマッピングのソース】フィールドには、自動的にソースコレクションのパスが記入されます。


注意:

コレクションの1要素のマッピングを実行するたびに警告が表示され、多くの場合【コレクションマッピングのソース】フィールドに入力する必要があることをユーザに知らせます。【編集/オプション】メニューの【コレクションからコレクションへのマッピングが必要な場合に、警告を表示する】オプション（【確認】項目）で、警告を非表示にできます。

キーボードによる方法

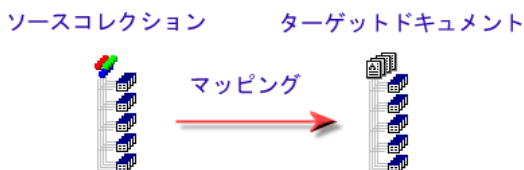
- 1 作業枠内にターゲットコレクションをドラッグします。
- 2 ソースコレクションのパス名を【コレクションマッピングのソース】フィールドに直接記入します。このコレクションの値を呼び出すわけではないので、名前は角括弧で囲みません。

コレクションからドキュメントへのマッピング

Connect-Itでは、ソースコレクションをターゲットドキュメントタイプのルートノード（)に関連付けることができます。データの処理時に、Connect-Itはソースコレクションにある構成要素と同じ数のターゲットドキュメントを作成します。

コレクションからドキュメントへのマッピングは、コレクションからコレクションへのマッピングと同じ方法で作成されます。ルートノードが、ターゲットコレクションの代わりになります。

図 5.7. コレクションからドキュメントへのマッピング



コレクションからドキュメントへのマッピングでは、コレクションのN個の構成要素はN個のドキュメントを作成する

例：

Desktop Discovery - ServiceCenterシナリオでは、「Software」コレクション（コンピュータにインストールされたソフトウェアを記録します）は、「pcsoftware」ドキュメントタイプにマップされています。データの処理時に「Software」コレクションのN個の構成要素は、N個の「pcsoftware」ドキュメントを作成し、これらのドキュメントは更にN個の入力イベントをServiceCenterに送信します。

フィールドからコレクションへのマッピング


フィールドからコレクションへのマッピングでは、コレクションに所属していない1つまたは複数のソースフィールドが、コレクションに所属する1つのターゲットフィールドにマップされます。このターゲットフィールドを含む構造体は、1つの構成要素のみから成るコレクションと見なされます。

例：

【InstalledCards】フィールドは、コンピュータにインストールされた全カード（マザーボード、グラフィックアダプタ、サウンドカード）をまとめています。ソースドキュメントタイプでは、これらの情報が別々のフィールド（【Motherboard】、【GraphicsCard】、【SoundCard】フィールド）になっています。

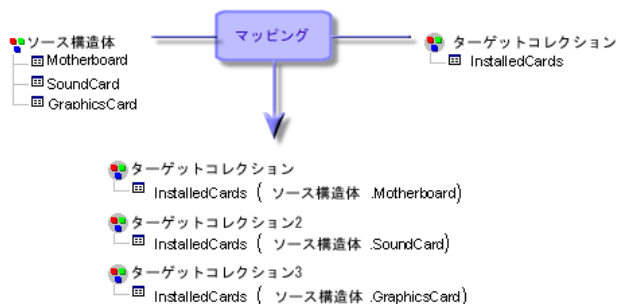
コレクションを作業枠内にドラッグした後、必要なだけコレクションを複製し、複製したコレクションの [InstalledCards] フィールドをそれぞれ、ソースフィールドにマップします。

作業枠内のコレクションを複製するには、

- 1 コレクションを選択します。
- 2  をクリックします。

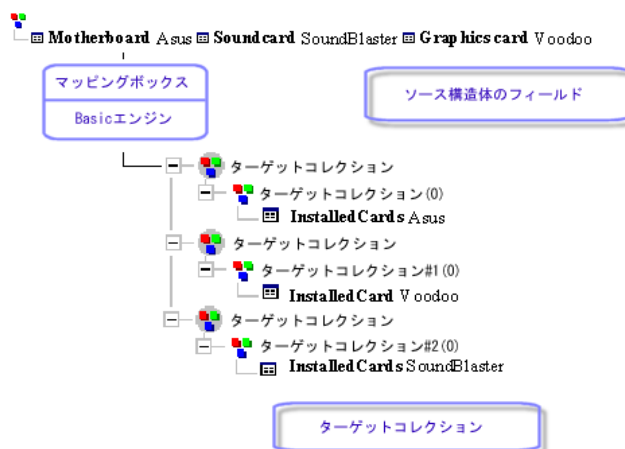
コレクションの1番目の複製には接頭辞#1が付き、2番目には#2が付きます。

図 5.8. フィールドからコレクションへのマッピング



フィールドからコレクションへのマッピングでは、ソース構造体のフィールドは複製されたターゲットコレクションの構成要素にマップされる

データの処理時に、マッピングボックスは1つの構成要素のみを含むターゲットコレクションを複製された数だけ生成します。各構成要素はそれぞれ1つのソースフィールド値を含みます。



ドキュメントタイプ要素のパス

ドキュメントタイプの要素の場所を [コレクションマッピングのソース] フィールドなどに入力するには、この要素のドキュメントタイプ内でのパスを指定する必要があります。ドキュメントタイプの要素の親子関係はピリオドで表現されます。以下の記述は関連し合う要素を表しています。

コレクションXのパスを指定するには以下の様に記述します。

構造体A.構造体B.コレクションX

このシンタックスは、コレクションXは構造体Bの子要素であり、構造体Bは構造体Aの子要素であることを示しています。

例：

Desktop Discovery - Asset Managementシナリオの「Asset information」マッピングでは、ターゲットドキュメントタイプ「amAsset」のコレクション「SubAssets」は、「Printer」コレクションにマップされています。このコレクションは「Peripherals」構造体の子に当たり、「Peripherals」はドキュメントタイプ「Machine」の「Hardware」構造体の子に当たるため、[コレクションマッピングのソース]フィールドには次のパスが表示されます。

Hardware.Peripherals.Printer

要素の名前自体にピリオドが含まれている場合は、シングルクォーテーションマーク (') で要素名を囲みます。

'Element.A'

例：

ServiceCenterコネクタの使用可能なドキュメントタイプの要素名の多くは、ピリオドを含んでいます。ServiceCenterコネクタの使用可能なドキュメントタイプ「ICMDevice」内にある、「sw.vendor」コレクションの[sw.vendor]フィールドを指すには、以下の用にパスを記述します。

'sw.vendor'.sw.vendor'

ピリオドが名前に含まれていない場合、シングルクォーテーションマークはつけてもつけなくても構いません。つまり、

ElementA または 'ElementA'の両方の書き方が可能です。

ドキュメント要素のパス

ドキュメント要素のパスの表記法は、ドキュメントタイプに使用されている表記法と同じです。コレクションの構成要素に関してのみ相違点があります。

ドキュメント内では、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。

パス内で特定の構成要素を指定するには、以下のシンタックスを用いて構成要素の順位を使用します。

コレクション名(構成要素の順位)

例：Softwareコレクションの第1番目の構成要素の[VersionName]フィールドのパスは、以下の通りです。

Software(0).VersionName コレクションの第2構成要素用のパスは、Software(1).VersionNameになります。

属性のコレクションのパス

属性のコレクション（1つのフィールドしか含まないコレクション）の特定の構成要素を指すには、パス名に必ず属性の名前を入れます。例えば、address属性を含むAddressコレクションの構成要素を指定するには、Address(1).Address, Address(2).Address, Address(3).Address と記述します。

注意:

コレクションからコレクションへのマッピングを実行する場合は（**[コレクションマッピングのソース]**フィールドに入力します）、コレクションの構成要素の番号を指定する必要はありません。

Blobタイプのフィールドのマッピング

Blob（binary large object）タイプのフィールドはバイナリデータ（サウンド、ビデオ、画像ファイル）のフィールドです。

Blobフィールドは別のBlobタイプのフィールドにのみマップされます。

2つの可能性があります。

- 1 ターゲットBlobフィールドをソースBlobフィールドにマップします。
この場合バイナリファイルは、シナリオのコネクタを経由してソースアプリケーションからターゲットアプリケーションへ転送されます。
- 2 Connect-Itとターゲットアプリケーションの共有ファイル内にあるバイナリファイルの名前を指定するフィールドに、ターゲットBlobフィールドをマップします。
この場合、マッピングスクリプトの入力ゾーン下にある**[ファイルからBLOBを読み込む]**オプションを使用する必要があります。

ファイルからBLOBを読み込む

このオプションでは以下の内容を指定できます。

- BlobファイルはConnect-Itとターゲットアプリケーションの共有フォルダ内にある
例：[ネットワーク上の共有フォルダ名]/blob/pictures
- ターゲットアプリケーションが共有フォルダからBlobファイルを読み込む
ターゲットアプリケーションが共有フォルダからファイルを読み込むようにするには、
 - 1 マッピング内で、ソースのBlobファイルの名前をターゲットアプリケーションのBlobタイプのファイルへ関連付けます。

ソースアプリケーションのファイル名は、ソースアプリケーションでBlobファイルに関連付けられている「name」フィールド（icon.name）、または共有フォルダ内のファイルの完全パス（[ネットワーク上の共有フォルダ名]/blob/pictures/monimage.png）になります。

2 **【ファイルからBLOBを読み込む】** オプションを選択します。

シナリオの起動中に、ターゲットアプリケーションは共有フォルダ内でBlobファイルを取得します。

ソースコレクションに要素がない場合でもドキュメントまたはコレクション要素を作成する

このオプションを選択すると、ソースコネクタに生成される構成要素がコレクションにない場合、以下の操作が実行されます。

- コレクションからドキュメントへのマッピングを実行すると、コレクションは、ターゲットコネクタに取り込まれるドキュメントを作成します。
本節の「[コレクションからドキュメントへのマッピング](#)」を参照してください。
- コレクションからコレクションへのマッピングを実行すると、コレクションは、ターゲットコネクタに取り込まれるドキュメント内に、コレクションの1構成要素を作成します。
本節の「[コレクションからコレクションへのマッピング](#)」を参照してください。

コレクションの構成要素やドキュメントの作成を強制すると、値がないことをターゲットアプリケーションに通知できるようになります。

スクリプトを使用したフィールドからフィールドへのマッピング

フィールドからフィールドへの直接マッピングが不可能な場合、マッピングスクリプト（[【マッピングスクリプト】](#)フィールド）が必要です。スクリプトは、ソース要素がターゲット要素に値を与えるようにするためどのようにソース要素を操作するか、を指定します。

例：

あるBasicスクリプトは2つのソース要素の連結を可能にします。この連結の戻り値は、データの処理時にターゲット要素に関連付けられます。

 **注意:**

マッピングスクリプトを作成する前に、フィールドからフィールドへの直接マッピングを試してください。

スクリプトの書き方

本節では以下の内容について説明します。

- 固定値をターゲット要素に関連付ける方法
- 複数のフィールドをドラッグ&ドロップで移動させる方法
- ソース要素とターゲット要素の位置を見つける方法
- ターゲット要素を複製する方法
- 要素のパスをコピーする方法
- マッピングスクリプトのショートカットメニューを使用する方法

固定値

固定値（日付、文字列、数値など）をターゲットフィールドへ関連付けるには、

- 1 作業枠にターゲットフィールドをドラッグします。
- 2 **[マッピングスクリプト]**フィールドに固定値を入力します。

例:

Desktop Discovery - Asset Managementシナリオでは、スキャンされた各コンピュータは、**[amAsset]**テーブルに新規の資産を作成します。デフォルトでは、このテーブルの各資産にはカテゴリ名がなければなりません。マッピングでは、固定値 `"/Materiel/Ordinateur de bureau/"` をamAssetドキュメントタイプの **[FullName.Category]**フィールドにマップします。

複数のフィールドをドラッグ&ドロップで移動させる方法

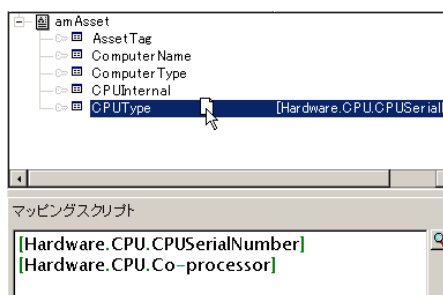
Basicスクリプトでは、Basic関数を用いて複数のソースフィールドを連結させることなどが可能です。

複数のフィールドをドラッグ&ドロップで移動させるには、

- 1 第1のソースフィールドをターゲットフィールドにマップします。


- 2 「Shift」キーを押したまま他のフィールドをドラッグします。フィールドはリストとして [マッピングスクリプト] 枠内に表示されます。これらのフィールドを使用してスクリプトを作成します。

図 5.9. 複数のフィールドをドラッグ&ドロップで移動させる方法



ソース要素とターゲット要素の位置を見つける方法

作業枠内にあるターゲット要素が、ターゲットドキュメントタイプ内のどこに位置していたかを調べる場合、以下の操作を実行します。

- 1 要素を作業枠内で選択します。
- 2  をクリックします。

この操作を行うと、ターゲットドキュメントタイプ枠内にある元のターゲット要素が選択されます。この機能は、ターゲットドキュメントタイプに多数の要素がある場合に便利です。


また、マップされたソース要素（青で表示されています）を作業枠内で見つけることもできます。

見つけるためには、ソースドキュメントタイプ枠内のソース要素をダブルクリックします。するとこのソース要素に関連するターゲット要素が、作業枠内で緑色になります。

ターゲット要素を複製する方法

複数のソース要素に1つのターゲットフィールドを関連付けるためには、ターゲットフィールドを複製する必要があります。例えば、ソースフィールドXとYをあるコレクション内のフィールドAに関連付けるとします。この場合、フィールドAが含まれているコレクションを複製してから、フィールドXを元のコレクションのフィールドAにマップし、フィールドYを複製されたコレクションのフィールドAへマップします。

要素を複製するには、

- 1 作業枠内にある要素を選択します。
- 2  をクリックします。

複製された要素には番号が付けられます。元の要素には番号0が付いており（これは表示されません）、最初に複製された要素には番号1、2つ目の複製には番号2がついています。この番号の付け方はConnect-Itで強制されており、変更は不可能です。

要素のパスをコピーする方法

ドキュメント内の要素	マッピングスクリプト内の要素のパス
amAstProjDesc	マッピングスクリプト
dIncluded	[Document.dDocDate]
dPlannedRemov	
dRemoved	
dtLastModif	
Field1	
Asset	
Project	
Document	
Comment	
dDocDate	

マッピングスクリプトでは、使用する要素の完全なパスを指定しなければなりません。例えば上の表では、フィールド [dDocDate] の完全パス名は [Document.dDocDate] です。

ドキュメントタイプの要素のパスを取得するには、以下の手順に従います。

- 1 要素を選択します。
- 2 右クリックして、ショートカットメニューから [パスをコピーする] を選択します（「Ctrl+C」キー）。
- 3 カーソルを [マッピングスクリプト] フィールド内に置きます。
- 4 右クリックして、ショートカットメニューから [貼り付け] を選択します。（「Ctrl+V」キー）

マッピングスクリプトのショートカットメニューを使用する方法

マッピングスクリプト内で右クリックすると、ショートカットメニューが表示されます。

このショートカットメニューでは以下の操作を実行できます（括弧内はショートカットキーです）。

- 元に戻す：キー
- やり直し：キー
- 切り取り（「Ctrl+X」キー）
- コピー（「Ctrl+C」キー）

- 貼り付け (「Ctrl+V」キー)
- すべて選択 (「Ctrl+A」キー)
- 使用されていないドキュメントタイプにフィルタを適用する
- ソースを表示する
- ターゲットを表示する

ショートカット (コンテキスト) メニュー

マッピングの編集用ウィンドウでは、4つのショートカットメニューのコマンドを、マッピングスクリプトを作成する際に使用できます。

ソースコネクタに発行された使用可能なドキュメントタイプの要素上に、カーソルが位置する時

コマンド	機能
このXMLのDTDをコピーする	ソースコネクタの生成用ドキュメントタイプのDTDをコピーします。
パスをコピーする	コンピュータのクリップボードに、選択した要素のパスをコピーします。
選択事項にマップされたノードを表示する	選択された要素に関連する (直接マッピングまたはスクリプト使用のマッピング) マッピングノードを緑色で表示します。
データを表示する	ソースドキュメントタイプのデータを確認するためのウィンドウを表示します。
データベースのプレビューのドキュメントタイプを編集する	コネクタの生成用ドキュメントタイプを作成するためのウィンドウが表示され、生成用ドキュメントタイプのデータを確認できるようになります。
使用されていないドキュメントタイプにフィルタを適用する	現在のマッピングで使用されていないドキュメントタイプにフィルタを適用します。
ソースを表示する	ソースドキュメントタイプ枠を表示 / 非表示にします。
ターゲットを表示する	ターゲットドキュメントタイプ枠を表示 / 非表示にします。

ターゲットコネクタに発行された使用可能なドキュメントタイプの要素上に、カーソルが位置する時

コマンド	機能
このXMLのDTDをコピーする	ターゲットコネクタの取り込み用ドキュメントタイプのDTDをコピーします。
パスをコピーする	コンピュータのクリップボードに、選択した要素のパスをコピーします。
この要素を追加する	作業枠内に要素を置きます。
データを表示する	ターゲットドキュメントタイプのデータを確認するためのウィンドウを表示します。
使用されていないドキュメントタイプにフィルタを適用する	現在のマッピングで使用されていないドキュメントタイプにフィルタを適用します。
ソースを表示する	ソースドキュメントタイプ枠を表示 / 非表示にします。
ターゲットを表示する	ターゲットドキュメントタイプ枠を表示 / 非表示にします。

作業枠にカーソルが位置する時

コマンド	機能
このXMLのDTDをコピーする	ターゲットコネクタの取り込み用ドキュメントタイプのDTDをコピーします。
パスをコピーする	選択した要素のパスをクリップボードにコピーします。
マッピングを編集する	テキストエディタでマッピングスクリプトを編集できます。
マッピングを説明する	選択されたマッピングノードを説明するためのウィンドウを表示します。
識別キー	ターゲットコネクタがデータベース型の場合、選択された要素が識別キーとして使用されていることを示します。
この要素を削除する	作業枠内から選択した要素を取り除きます。
マッピングをコピーする	マッピングの一部または全体をクリップボードにコピーします。
マッピングを貼り付ける	現在のマッピング内に、クリップボードのマッピングまたはマッピングの一部を貼り付けます。
使用されていないドキュメントタイプにフィルタを適用する	現在のマッピングで使用されていないドキュメントタイプにフィルタを適用します。
ソースを表示する	ソースドキュメントタイプ枠を表示 / 非表示にします。
ターゲットを表示する	ターゲットドキュメントタイプ枠を表示 / 非表示にします。

[マッピングスクリプト]の編集用ゾーンにカーソルが位置する時

コマンド	機能
元に戻す: キー	一番最後のキー入力を取り消します。
やり直し: キー	一番最後のキー入力を繰り返します。
切り取り	選択したテキストを切り取ります。
コピー	クリップボードに選択したテキストをコピーします。
貼り付け	編集用ゾーンにクリップボードの内容を貼り付けます。
全て選択	編集用ゾーンのテキストを全て選択します。
使用されていないドキュメントタイプにフィルタを適用する	現在のマッピングで使用されていないドキュメントタイプにフィルタを適用します。
ソースを表示する	ソースドキュメントタイプ枠を表示 / 非表示にします。
ターゲットを表示する	ターゲットドキュメントタイプ枠を表示 / 非表示にします。

スクリプトの作成

マッピングではBasicスクリプトを使用して以下の操作を実行できます。

- 計算された値をターゲットドキュメントタイプのフィールドに関連付ける
 - 構造体またはコレクションのフィールドの処理を、条件の検証に従属させる
- 以下の要素をスクリプトの作成に使用します。

- 通常のBasic関数
- Connect-It用に開発されたBasic関数
- 変数
- マップテーブル
- 文字列ファイル

Connect-Itのエディタを使用してこれらの要素を編集します。

エディタを表示するには [シナリオ] メニューを使用します。

文字列テーブル、マップテーブル、グローバル関数と変数は、「.scn」ファイルとは別に、それぞれの内容に対応する拡張子の付いたファイル名で保存されません。

ファイル名の拡張子

.str (文字列用)	文字列テーブル
.mpt (マップテーブル用)	マップテーブル
.bas (Basic用)	グローバル関数と変数

注意:

関連する「.str」、「.mpt」と「.bas」ファイルのないシナリオ (SCNファイル) は機能しません。シナリオを移動させる時は、これらの関連するファイルも同時に移すようにしてください。また、SCNファイルに対してこれらのファイルが適切な場所に位置するようにしてください。ファイルの位置が不適切であると、データの処理時にシナリオがファイルの内容を取得できない可能性があります。

本節では以下の内容について説明します。

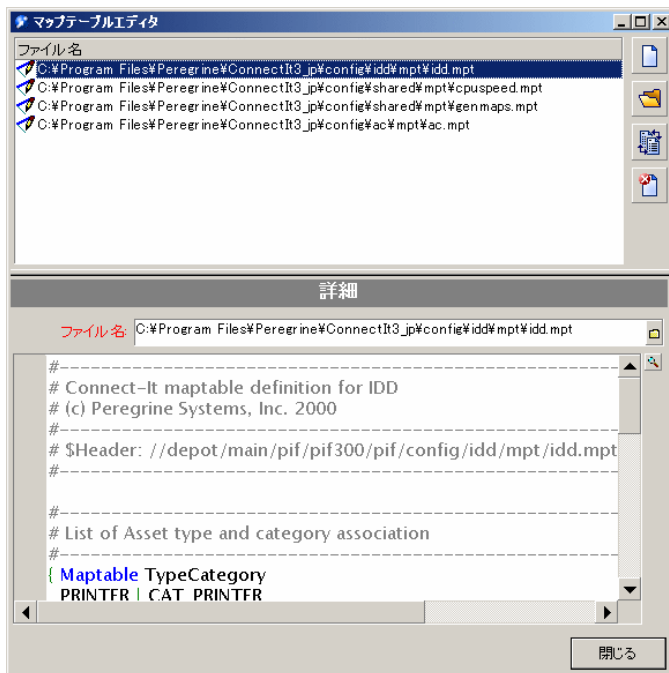
- Connect-Itエディタの使用方法
- 文字列テーブル
- マップテーブル
- ダイナミックマップテーブル
- グローバル関数と変数
- ユーザフォーマット
- Connect-It用に開発されたBasic関数 (PIF関数)
- テキストエディタ
- シナリオとシナリオに関連するファイルの移動
- オンラインのプログラム用参考ガイド

エディタの使用

[シナリオ]メニューでエディタを選択すると、ウィンドウが開きます。ウィンドウは2つの部分から成っています。

- 現在のシナリオに関連したファイル (「.str」、「.mpt」または「.bas」) を含む枠
- ファイルの編集用の枠 (文字列テーブル、マップテーブル、グローバル関数と変数)



図 5.10. マップテーブルのエディタ



上記3つのエディタは、マッピングスクリプトで使用するエディタと同一のテキストエディタを使用します。

新規ファイルを作成する

新規ファイルを作成するには、

- 1 ツールバーの  をクリックします。
- 2 **【ファイル名】**フィールドにファイル名を入力します。
ハードディスク内を検索し、ファイルを作成するフォルダを直接指定するには、 をクリックします。
- 3 **【作成】**をクリックします。

警告:

ファイルが既に存在する場合、既存ファイルは削除されます。

既存のファイルを開く


既存のファイルを開くには、以下の2通りの方法があります。

関連するファイルの枠（画面上）にあるファイルを開く

現在のシナリオに関連するファイルの枠内にあるファイルを開くには、ファイルをクリックします。



関連するファイルの枠に表示されていないファイルを開く

現在のシナリオに関連するファイルの枠内に、表示されていないファイルを開くには、

- 1 ツールバーの  をクリックします。
- 2 **【ファイル名】** フィールドにファイルの完全パスを入力します。
ハードディスク内を検索してファイルを見つけることも可能です。
- 3 **【開く】** をクリックします。


関連するファイルを複製する

関連するファイルを複製するには、

- 1 関連するファイルの枠内でファイルを選択します。
- 2 ツールバーの  をクリックします。
- 3 **【ファイル名】** フィールド内に複製されたファイルの名前を入力します。
ハードディスク内を検索し、ファイルを作成するフォルダを直接指定するには、 をクリックします。

関連するファイルを削除する

現在のファイルに関連するファイルを削除するには、

- 1 関連するファイルの枠内でファイルを選択します。
- 2 ツールバーの  をクリックします。

文字列テーブル

【シナリオ】メニューで文字列テーブルにアクセスします。

文字列テーブルでは、固有の識別子が各文字列に付いています。データの処理時に、Basicスクリプトは識別子を識別子に対応する文字列に置き換えます。

マッピング内で文字列を使用するには、PifStrVal関数の後に、ダブルクォーテーションマークと括弧で囲まれた文字列の識別子を入力します。

次の例は、Desktop Discovery - Asset Managementシナリオで使用されている、「category.str」文字列テーブルの一部です。

```
CAT_UNIX, "/Hardware/Unix Workstation"
CAT_SERVER, "/Hardware/Server"
CAT_WORKSTATION, "/Hardware/PC"
CAT_MAC, "/Hardware/Mac"
CAT_TERMINAL, "/Hardware/Terminal"
CAT_PORTABLE, "/Hardware/Portable"
```

例：

"/IT/UNIXワークステーション" の値を取得するには、マッピングスクリプトで次のコード行を使用します：PifStrVal("CAT_UNIX")

このテーブルでは、識別子「CAT_UNIX」は"/IT/UNIXワークステーション"の値に対応しています。このため、文字列内で名前を変更すると、シナリオはこの変更事項を考慮に入れるため、この識別子を使用するBasicスクリプトを再びコンパイルする必要はありません。また、各言語ごとの文字列テーブルを複数使用することもできます。例えば「frcategory.str」をフランス語用に、「grcategory.str」をドイツ語用に使用することなどが可能です。

マップテーブル

[シナリオ] メニューでマップテーブルにアクセスします。

マップテーブルはテーブルの形を取っており、第1列目にはキーが、それ以降の列には値が含まれています。1つの列の中で、各キーは1つの値に対応します。

マッピング内で列の値を取得するには、PifMapValue()関数と、キーのパラメータ、キーがマップテーブルで定義されていない場合はデフォルト値、マップテーブルの名前、と列番号を使用します。

データの処理時に、スクリプト内でキーが見つかると、キーはスクリプト内で指定された列の値に置き換えられます。

以下のBrandマップテーブルでは、AppleキーとMacIntoshキーは、スクリプト内で列1番が指定されていると、"Apple" という値を戻します。

```
{ MapTable Brand
Compaq | Compaq
IBM -Lexmark| IBM - Lexmark
Hewlett Packard| Hewlett Packard
HP-UX| Hewlett Packard
Toshiba | Toshiba
```

```
Apple | Apple
Macintosh | Apple}
```

例：

"HP-UX" という値を取得するためには、以下のコード行をマッピングスクリプト内で使用します：PifMapValue([マッピングでのフィールド名], "Brand", 0, "DefaultValue")

この関数に関する詳しい説明は、オンラインの『プログラム用参考ガイド』のPifMapValue()関数を参照してください。

複数の言語用にマップテーブルを作成する

様々な言語の文字列を戻すマップテーブルを作成するには、

- 1 文字列ファイルを作成します。ファイル内の各行では、1つの識別子が1つの外国語の文字列（英語、フランス語、など）に、[識別子], ["外国語の文字列"] の形で関連付けられます。例：「category.str」ファイルには、英語用のCAT_UNIX, "UNIX Workstation") という行が含まれており、「fcategory.str」ファイルには、フランス語用のCAT_UNIX, "Station de travail UNIX") という行が含まれています。
- 2 このファイルをマップテーブルのファイルに含むには、次のシンタックスを使用します：#include_str "[ファイル名]" 例：#include_str "category.str"
- 3 [dollar]([識別子]) のシンタックスで識別子に各文字列に関連付けると、文字列をマップテーブル内で参照できます。

```
#include_str "category.str"
{ MapTable Category
[dollar](IDS_CAT_UNIX) | workstation}
{ MapTable Sc2AcCat
[dollar](IDS_CAT_UNIX) | workstation}
```

ダイナミックマップテーブル

ダイナミックマップテーブルを使用すると、ODBCデータベースでSQLクエリを実行し、マップテーブルと呼ばれるデータテーブルの形でクエリに戻された値を取得することができます。シナリオをテストまたは起動するたびに、このデータテーブルの内容は動的に更新されます。

ダイナミックマップテーブルを作成するには、PifCreateDynaMappable関数を使用する必要があります。この関数に関する詳しい説明は、オンラインの『プログラム用参考ガイド』を参照してください。

グローバル関数と変数

【シナリオ】メニューで、グローバル関数と変数のエディタにアクセスします。マッピングスクリプトは、スクリプト内の数箇所に関数や変数を使用します。これら複数のグローバル関数と変数は、同一の「.bas」ファイル内で保存することができます。保存された後、関数と変数はマッピングスクリプト内で呼び出されます。マッピングボックスは、データの処理中にシナリオに付属した「.bas」ファイルにある関数と変数を使用します。

グローバル関数CPUSpeed()は、「gen.bas」ファイルに含まれています。この関数はマップテーブルCPUSpeed()を参照しており、CPU速度の汎用値を指定します。


```
-----
' Returns the generic CPU speed frequency
' Use the generic maptable
' CPUSpeed'
Function CPUSpeed(ByVal strValue As String) As Integer
  Dim iTmp As Integer
  iTmp = CInt(strValue) / 10
  CPUSpeed = CInt(PifMapValue(CStr(iTmp * 10), "CPUSpeed", 1, strValue))
End Function
```

ユーザフォーマット

ユーザフォーマットとは、ユーザがマッピングスクリプト内で使用するために定義する日付型または数値のフォーマットを指します。フォーマットは、PifUserFmtVarToStr関数とPifUserFmtStrToVar関数と共にのみ使用されます。使用に関する詳細は、オンラインの『プログラム用参考ガイド』（Connect-Itでのスクリプト作成中に「F1」キーを押すと表示されます）を参照してください。

日付型のユーザフォーマットを作成する

日付型のユーザフォーマットを作成するには、

- 1 【シナリオ / ユーザフォーマット】メニューを選択します。
- 2 表示されるウィザードのページで、【次へ】をクリックします。
このページに表示される日付型のフォーマットを、ユーザが定義することはできません。
- 3 表示されるウィザードのページで、 をクリックします。
- 4 既存フォーマットのリスト内で、リスト最後のフォーマット名の下（【名前】の列）をクリックします。
- 5 表示される編集用ゾーンにフォーマットの名前を入力します。

- 名前にスペースを使用することはできません。
- 6 入力した名前の隣の **[フォーマット]** 列をクリックします。
 - 7 **[日付型フォーマットの記号]** 枠内の属性を用いて、公式を入力します。
 - 8 **✓** をクリックします。
- 作成したフォーマットの例が、**[日付型フォーマットの結果]** フィールドに表示されます。

数値のユーザフォーマットを作成する

数値のユーザフォーマットを作成するには、

- 1 **[シナリオ / ユーザフォーマット]** メニューを選択します。
 - 2 表示されるウィザードのページで、**[次へ]** をクリックします。
このページに表示される数値のフォーマットを、ユーザが定義することはできません。
 - 3 **✎** をクリックします。
 - 4 既存フォーマットのリスト内で、リスト最後のフォーマット名の下 (**[名前]** の列) をクリックします。
 - 5 表示される編集用ゾーンにフォーマットの名前を入力します。
名前にスペースを使用することはできません。
 - 6 入力した名前の隣の **[フォーマット]** 列をクリックします。
 - 7 **[数値フォーマットの記号]** 枠内の属性を用いて、公式を入力します。
 - 8 **✓** をクリックします。
- 作成したフォーマットの例が、**[数値フォーマットの結果]** フィールドに表示されます。

フォーマット作成に使用するシンタックス

フォーマット公式の作成時には、以下の規則を守る必要があります。

- 日付型または数値のフォーマットに使用する記号は、Windowsオペレーティングシステムに使用されている記号と同一にします。
- フォーマット内に現れる文字列は、シングルクォーテーションマークで囲みます。
- 2つの値の間のスペースも、文字列と同様にシングルクォーテーションマークで囲みます。

表 5.1. 日付型フォーマットの例

フォーマットの公式	例
yyyy'-mm'-'dd	2002-02-07
hh':nn':ss	11:55:29
h':nn':ss	11:55:41
hh':nn	14:18
hh" 'h' "nn	15:54:53
h t 'nn' 'tt	2 29 pm
dd/'MM'/yy	07/02/02
dd.'MM'.yy	07.02.02
dd-'MM'-'yy	07-02-02
dd/'MM'/yyyy	07/02/2002
yyyy'年'MM'月'dd'日'dddd	2002年02月07日木曜日


表 5.2. 数値のユーザフォーマットの例

フォーマットの公式	例
n' 'n','dd'-'USD'	1 0 2 0 3,41-USD
-'USD'n'.nnn','ddd	-USD10.203,408
-n'.nnn','dd'USD'	-10.203,41USD
-n'.nnn','ddd'USD'	-10.203,408USD

PIF関数

PIF関数は、Connect-Itのマッピングスクリプト用に特別に開発されたものです。全関数は、Connect-Itのインストール先フォルダの「doc」サブフォルダにある、オンライン『プログラム用参考ガイド』内で説明されています。

テキストエディタ

 アイコンをクリックすると、テキストエディタにアクセスできます。このアイコンは、各エディタ内と、マッピングの編集用ウィンドウ内の【マッピングスクリプト】フィールドの横にあります。

このテキストエディタでは、Basicスクリプトと「.str」、「.mpt」と「.bas」ファイルを容易に編集できます。エディタは通常のテキストエディタの機能（コ

ピー、貼り付け、現在のアクションをキャンセルする、など)を備えています。これらの機能には、**[編集]**メニューまたはツールバーからアクセスできます。

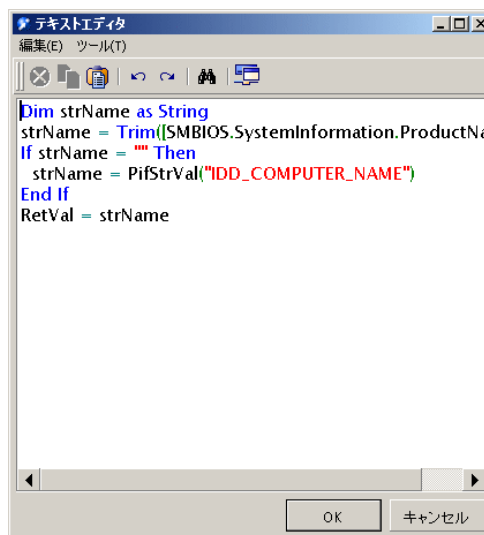


表 5.3. テキストエディタ - ツールバー

アイコン	機能
	選択したテキストを切り取ります。
	テキストをコピーします。
	クリップボードにテキストを貼り付けます。
	現在のアクションをキャンセルします。
	キャンセルしたアクションを繰り返します。
	テキストエディタの設定用ウィンドウを表示します。

テキストエディタの設定

テキストエディタを設定するには、

- をクリックします。
または
- **[ツール / オプション]** メニューを選択します。

テキストエディタを設定すると、ファイル作成用のテンプレートを変更できません。

警告:

各テキストエディタで使用可能なテンプレートを編集する場合は、まず、Connect-Itのインストール先フォルダのサブフォルダ「bin32」にある「codeedit.cfg」ファイルが、読み取り専用でないことを確認してください。

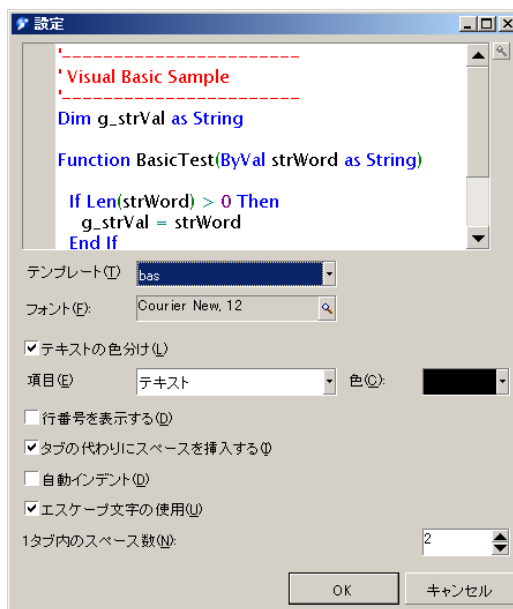
各エディタ用のテンプレートを使用する必要があります。例えばマップテーブルを作成している場合、「mpt」テンプレートがテキストエディタに使用されません。

テンプレート名	編集する要素
bas	マッピングスクリプト グローバル関数
str	文字列テーブル
mpt	マップテーブル
sql	SQLクエリ
scn	Connect-Itシナリオ
Default	上記以外の全ファイル

テンプレートには以下の設定項目があります。

- 文字のフォントを変更する（タイプ、スタイル）
- 要素の色を選択する
- 行番号を表示 / 非表示にする
- 1から8個のスペースをタブの代わりに挿入する

図 5.11. テキストエディタの設定



テンプレートの変更

テンプレート内で使用されているフォントを変更するには、【フォント】フィールドの横にある拡大鏡アイコンをクリックします。ダイアログボックスが現れ、コンピュータのコンピュータにインストールされているフォントから選択できるようになります。

テキスト内の要素の色も変更できます。色を変更できる要素は以下の通りです。

- テキスト
- 背景
- 選択されたテキスト
- 選択されたテキストの背景
- 行番号
- 数値
- 区切り文字
- コメント
- 文字列
- キーワード
- オペレータ

要素に色をつけるには、【要素】フィールドで1項目を選択し【色】フィールド内で色を選択します。

テキストに色をつけない場合は、【テキストの色分け】チェックボックスをオフにします。

行番号を表示し、タブの代わりにスペースを挿入するには、それぞれのチェックボックスをオンにします。

シナリオと、シナリオに関連するファイルの移動

シナリオを別の場所移すたびに、シナリオに関連するファイルも同時に移動させる必要があります。これらのファイルがないとシナリオは正常に実行されません。また、シナリオの元のツリー構造を、新規の場所で正確に再現することも重要です。

移されたシナリオが、全シナリオ共通のマッピングテーブルを使用している場合、これらの共通ファイルの格納されているフォルダを含むツリー構造を、再現する必要があります。例：全用例シナリオに共通している「.str」、「.mpt」と「.bas」ファイルは、Connect-Itインストール先フォルダのサブフォルダ「config」の、「shared」フォルダにあります。

オンラインのプログラム用参考ガイド

スクリプトの作成には、Connect-Itにあるオンラインの『プログラム用参考ガイド』が便利です。

以下の場所にカーソルが位置する時に「F1」キーを押すと、『プログラム用参考ガイド』が表示されます。

- マッピングの編集用ウィンドウ内の【マッピングスクリプト】フィールド
- エディタの入力ゾーン

スクリプトの認証

スクリプトの一貫性は、Connect-Itによりデフォルトで検査されています。スクリプトの認証では、Basicシンタックスが検査され、また、マッピングスクリプト内で参照されている要素がソースドキュメントに存在するかどうかテストされます（検証では、ソースドキュメントの構造内で表示されている要素のみが「存在する」と見なされます）。

 注意:

[コンパイルのエラーを含むスクリプトの認証を許可しない] オプション (確認項目) では、上記のスクリプトの認証を有効または無効にできません。

6 | ピボットドキュメントタイプ

ピボットドキュメントタイプは、構成要素（資産、ソフトウェア、従業員など）の一般的な表現に当たります。ソースコネクタは、自分のドキュメントタイプとピボットドキュメントタイプ（生成用または取り込み用）を一致させます。ピボットドキュメントタイプを使用するコネクタは、マッピングなしでリンクされます。

ピボットドキュメントタイプは以下のコネクタで使用できます。

- Asset Managementコネクタ（ソフトウェアのバージョン4.xxと4.1）
- ServiceCenterコネクタ（ソフトウェアのバージョン3.xxと4.xx）
- InfraTools Desktop Discoveryコネクタ
- InfraTools Managementコネクタ
- Unicenter AMOコネクタ
- PDA inventoryコネクタ
- Tivoli Inventoryコネクタ
- Winpark Actimaコネクタ
- ゲートウェイ3.xコネクタ

ピボットドキュメントタイプの機能

ピボットドキュメントタイプは、一般的なドキュメントタイプです。

ソースコネクタは、使用可能なドキュメントタイプからピボットドキュメントタイプへのマッピングを定義します。ターゲットコネクタはピボットドキュメントタイプから、取り込み用ドキュメントタイプへのマッピングを作成します。これにより、特定のマッピングを作成せずに、ソースコネクタのデータをターゲットコネクタへ送信することが可能になります。ピボットドキュメントタイプからまたはピボットドキュメントタイプへのマッピングでは、ソースコネクタとターゲットコネクタ間でデータを交換するために必要なデータ変換を定義します。

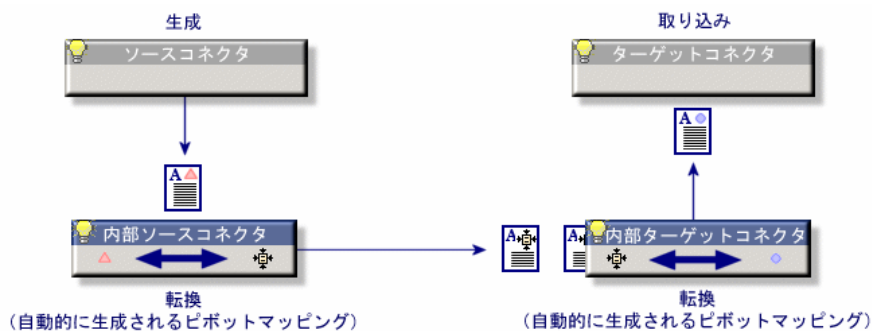
自動的に作成されるマッピング

ソースコネクタが、ピボットドキュメントタイプに基づいて初めてドキュメントを生成すると、自動的に以下のマッピングが作成されます。

- ソースコネクタの生成用ドキュメントタイプと、ピボットドキュメントタイプ間のマッピング
- ピボットドキュメントタイプと、ターゲットコネクタの取り込み用ドキュメントタイプ間のマッピング

以上2つのマッピングはピボットマッピングに基づいています。ピボットマッピングは、それぞれの使用可能なドキュメントタイプをピボットドキュメントタイプにマップします。

図 6.1. ピボットドキュメントタイプの使用



ピボットドキュメントタイプを使用して統合シナリオを作成する

本節では、ピボットドキュメントタイプを使用して統合シナリオを作成する方法を説明します。このシナリオは、1つのソースコネクタと1つのターゲットコネクタを使用します。

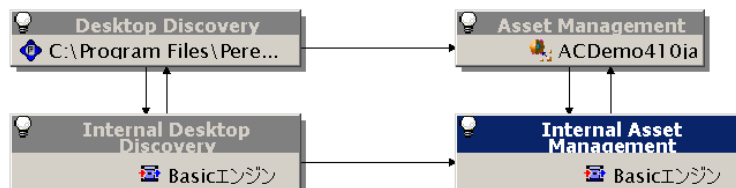
シナリオを作成するには、

- 1 シナリオビルダを起動します。
- 2 ソースコネクタとターゲットコネクタを、シナリオ図内にドラッグします。
- 3 **[コネクタの設定]** ウィザードで2つのコネクタを設定します。
[コネクタの設定] ウィザードの **[ピボットドキュメントタイプの使用]** ページで、**[ピボットドキュメントタイプを使用する]** オプションを選択します。
- 4 「Shift」キーを押したまま、ソースコネクタをターゲットコネクタにリンクします。
「Shift」キーを押したままにしないと、2つのコネクタ間に自動的にマッピングボックスが作成されます。
- 5 ソースコネクタを選択します。
- 6 ソースコネクタの使用可能なドキュメントタイプのリストにある、ピボットドキュメントタイプ (PrgnAssetなど) に基づいて、1つまたは複数の生成用ドキュメントタイプを作成します。
- 7 ソースコネクタを起動してシナリオをテストします。
ソースコネクタを初めて起動すると、使用可能なドキュメントタイプとピボットドキュメントタイプ間のマッピングが、自動的に作成されます。
- 8 スケジュールとサービスをシナリオに関連付け、シナリオをプロダクションモードにします。

内部コネクタを表示する

あるコネクタがピボットドキュメントタイプを使用する場合、シナリオの各コネクタ用に内部コネクタが作成されます。内部コネクタを表示するには、**[シナリオ / 内部コネクタを表示する]** メニューを選択します。

図 6.2. 内部コネクタの表示



内部コネクタを表示すると、ピボットドキュメントタイプが使用される時に自動的に作成されるマッピングを、編集できるようになります。

[マッピング] タブと [ピボットマッピング] タブ

ピボットコネクタをシナリオ図内で選択すると、2つのタブが表示されます。

- [マッピング] タブ
- [ピボットマッピング] タブ

[マッピング] タブ

このタブは、コネクタの生成用または取り込み用ドキュメントタイプと、ピボットドキュメントタイプ間に自動的に作成されたマッピング用です。

これらのマッピングを参照することは可能ですが、編集はできません。マッピングはシナリオが起動するたびに再生成されます。マッピングを変更するには、その土台にあたるピボットマッピングを変更する必要があります。

[ピボットマッピング] タブ

このタブは、コネクタの使用可能なドキュメントタイプと、ピボットドキュメントタイプ間のマッピング用です。これらのマッピングは、自動的に作成される [マッピング] タブ内のマッピングの基礎となる、モデルの役割を果たしています。

ピボットドキュメントタイプのリスト

ピボットドキュメントタイプには以下のものがあります。

- prgnAsset

- prgnCompany
- prgnContract
- prgnCostCenter
- prgnDepartment
- prgnFeatures
- prgnLicense
- prgnLocation
- prgnModel
- prgnNetworkDevice
- prgnPC
- prgnPerson
- prgnPrinter
- prgnSoftInst
- prgnTelephony

コネクタの使用可能なドキュメントタイプのリスト内では、この種のドキュメントタイプは専用のアイコン(☼)で示されています。

「使用可能なドキュメントタイプ」と「ピボットドキュメントタイプ」間のマッピングを変更する

コネクタの使用可能なドキュメントタイプと、ピボットドキュメントタイプ間のマッピングを変更するには、

- 1 ピボットドキュメントタイプを使用するシナリオを作成します。
- 2 **[シナリオ / 内部コネクタを表示する]**メニューで、ピボットコネクタを表示します。
- 3 ピボットコネクタを選択します。
- 4 **[ピボットマッピング]**タブでマッピングを選択します。
- 5 マッピングを変更します。

コネクタの使用可能なドキュメントタイプと、ピボットドキュメントタイプ間のマッピングの詳細情報については、ピボットドキュメントタイプ用各コネクタの「**ピボットマッピング**」節内を参照してください。

注意:

マッピングを変更するには、コネクタの「.piv」ファイルが読み取りと書き込み可能でなければなりません。

ターゲットコネクタを変更する

ピボットドキュメントタイプを使用すると、シナリオ内でターゲットコネクタを変更しても、他に変更を加えずに、ソースコネクタのドキュメント生成をそのまま再起動することができます。

ターゲットコネクタを変更するには、

- 1 ピボットドキュメントタイプを使用したシナリオを開きます。
- 2 シナリオ図内に新規のターゲットコネクタを置きます。
- 3 **[コネクタの設定]** ウィザードで、コネクタを設定します。
- 4 「Shift」キーを押したまま、ソースコネクタをターゲットコネクタにリンクします。
- 5 スケジュールまたはソースコネクタを手動で再起動します。

7 | 統合シナリオのテストとデバッグ

シナリオをテストすると以下の内容を確認できるようになります。

- 生成用ドキュメントタイプを作成した場合、ソースコネクタが適切にドキュメントを生成するか
- マッピングボックスがこれらのドキュメントを適切に変換するか
- マッピングボックスがドキュメントを変換した後、ターゲットコネクタがドキュメントを適切に取り込むか

⚠ 警告:

テストには、テスト用データ（データベースまたはデモ用ファイル）の使用をお勧めします。テストをせずにシナリオをプロダクションモードで使用すると、実際のデータを破損する可能性があります。

生成用ドキュメントタイプのテスト

1つのシナリオ内でコネクタが複数の生成用ドキュメントタイプを使用する場合、各生成用ドキュメントタイプをテストすることをお勧めします。

ソースコネクタの生成用ドキュメントタイプをテストするには、

- 1 ソースコネクタを開きます。

- 2 [ドキュメントタイプ] タブを選択します。
- 3 テストするドキュメントタイプ以外の、生成用ドキュメントタイプのチェックボックスを全てオフにします。
複数のドキュメントタイプをテストする場合は、ドキュメントタイプを複数選択したまま次の手順に進みます。
- 4 ▶をクリックして、または[ツール / 生成する]メニューを選択してドキュメントタイプの生成を起動します。
- 5 ドキュメントログの内容を読み、ソースコネクタがドキュメントを生成する際にどのような問題が発生したかを確認します。

 **注意:**

シナリオビルダのオプションの、([編集 / オプション]) [ドキュメントの生成テスト中に生成するドキュメントの数] オプションで、テスト時に生成するドキュメントの数を指定できます。

このオプションは、生成用ドキュメントタイプをテストする際に自動的に使用されます。しかし、シナリオがWindowsのサービスとして実行されると、このオプションは自動的に無効になります。

ドキュメントログの使用

ドキュメントログを使用すると、ソースコネクタがドキュメントを生成している際にどのような問題が発生したかを確認できます。

問題を解決するには、処理中に問題の起こったドキュメントのみを表示するようドキュメントログを設定することをお勧めします。

ドキュメントログに関する詳細は、「シナリオビルダ」章の「ログ」の節、「ドキュメントログ」を参照してください。

キャッシュファイルを使用する

コネクタ用のキャッシュファイルには、コネクタの使用可能なドキュメントタイプの情報が含まれています。

例:

データベース型のコネクタを開くと、コネクタは、コネクタの設定で指定したデータベースの全テーブルの詳細を取得します。キャッシュファイルを使用すると、最後にコネクタを開いた際に作成されたキャッシュファイル内のデータ記述を使用するため、データ記述を新たに取得する必要がなくなります。

シナリオのテストやデバック段階でシナリオのコネクタを頻繁に開閉する場合、この機能は非常に便利です。

キャッシュファイルは以下のコネクタで使用できます。

- Action Request Systemコネクタ
- Asset Managementコネクタ
- ServiceCenterコネクタ
- InfraTools Network Discoveryコネクタ
- InfraTools Managementコネクタ
- FacilityCenterコネクタ
- TeleCenterコネクタ
- InfraTools Managementコネクタ
- LDAPコネクタ
- データベースコネクタ
- Lotus Notesコネクタ
- 全てのインベントリコネクタ

コネクタが外部アプリケーションに接続せずに、キャッシュファイルを使用するように設定するには、

- 1 [コネクタの設定] ウィザードの [キャッシュを設定する] ページで、[キャッシュファイルを使用する] を選択します。
- 2 コネクタの使用可能なドキュメントタイプの詳細を、キャッシュファイルに含めるために、コネクタを開きます。

注意: コネクタを最初に開く際に、キャッシュの使用を示すアイコンは、コネクタのボックス上には表示されません。

- 3 コネクタを一旦閉じてから、再び開きます。

この時点で、コネクタは外部アプリケーションには接続せず、キャッシュファイルから使用可能なドキュメントタイプを発行します。

注意:

キャッシュファイルを使用するコネクタを開くと、データベースを表すアイコンが、黄色の電球のアイコンの上に表示されます。

図 7.1. コネクタ - キャッシュ使用を示すアイコン



キャッシュファイルの内容を消去する

キャッシュファイルの内容を消去するには、

- 1 コネクタを開きます。
- 2 **[シナリオ / キャッシュを削除する]** メニューを選択します。
コネクタ1つのみのキャッシュファイルの内容を削除するには、**[ツール / キャッシュ / キャッシュを削除する]** を選択します。
コネクタを次回開くと、コネクタは外部アプリケーションに接続し、キャッシュファイルの内容と外部アプリケーション内の情報の同期をとります。

警告:

キャッシュファイルの同期をとるには、シナリオビルダのオプションで**[オフラインで作業する]** を**[いいえ]** にする必要があります。

キャッシュファイルの内容と外部アプリケーション内の内容の同期をとる

キャッシュファイルの内容と、外部アプリケーション内の内容の同期をとるには、


- 1 シナリオのコネクタを開きます。
- 2 **[シナリオ / キャッシュの同期をとる]** メニューを選択します。
コネクタ1つのみのキャッシュファイルの同期をとるには、**[ツール / キャッシュ / キャッシュの同期をとる]** を選択します。
- 3 Connect-Itがコネクタの使用可能なドキュメントタイプを取得するのを待ちます。
同期化が終了すると、コネクタの発行する使用可能なドキュメントタイプは、外部アプリケーションの内容に一致するようになります。

警告:

キャッシュファイルの同期をとるには、シナリオビルダのオプションで**[オフラインで作業する]** を**[いいえ]** にする必要があります。

オフラインで作業する

[オフラインで作業する] を使用すると、シナリオのコネクタを外部アプリケーションに接続させずに、作業することができます。

オフラインで作業するには、コネクタを最初に開いた後、 をクリックします。オフラインモードは、シナリオのコネクタ用にキャッシュファイルを自動的に作成します。

コネクタを閉じてから再度開くと、オフラインモードはオンのままになっており、コネクタはキャッシュファイルの内容に基づいて、使用可能なドキュメントタイプを発行します。

注意:

オフラインでの作業中にコネクタを開くと、データベースを表すアイコンが、消えた電球のアイコンの上に表示されます。

図 7.2. コネクタ - オフラインセッションを示すアイコン



重要項目:

オフラインモードでは、ソースコネクタからデータを取得することはできません。生成用ドキュメントタイプの編集用ウィンドウで、ソースデータのプレビュー機能は無効になっています。このモードはマッピングを定義するためのみに使用されます。

1コネクタの始動時に現在のシナリオの全コネクタを開く

このオプションにより、1コネクタの始動時に全コネクタを自動的に開くことができます。このオプションは、多数のコネクタを含むシナリオを使う場合に便利です。

テストモード

- ターゲットコネクタがトランザクションをサポートする場合、データが挿入されても、トランザクションのロールバック（データ挿入前の状態への復帰）が実行されます。これにより、ターゲットアプリケーションのデータを変更することなく、ドキュメントの取り込みをテストできるようになります。
- ターゲットコネクタがトランザクションをサポートしない場合、データはターゲットアプリケーションへは送信されません。このためドキュメントの生成とマッピングのテストは実行できますが、ドキュメントの取り込みをテストすることはできません。

一時停止を実行する

シナリオ内で多数のドキュメントを処理する場合、全ソースデータの処理の終了を待たずにConnect-Itログを読み、エラーがあるかどうかを確認するには、一時停止を実行する必要があります。

マッピング定義に関するアドバイス

Basicスクリプトのシンタックスエラーの場所（マッピング、関連する要素）を見つけるのが困難な場合もあります。このため、全スクリプトを作成してまとめてテストするよりも、1スクリプトの作成ごとにテストを実行することをお勧めします。

8 | シナリオ文書

シナリオビルダでは、現在のシナリオに関する詳細を含むシナリオ文書を表示または作成することができます。

シナリオ文書の情報はシナリオのSCNファイルから動的に抽出されるため、常に更新された内容になっています。

シナリオ文書の内容

文書には以下のセクションが含まれています。

- 一般情報
このセクションには以下の情報が含まれています。
 - 名前
シナリオ名の前にはコンピュータ上の完全パスがあります。
 - 最終変更日
 - シナリオの最終変更時に使用したConnect-Itのバージョンとビルド番号
 - シナリオの最終履歴にユーザが入力した全情報（**[ファイル/シナリオの履歴]**メニュー）。
- コネクタの設定

このセクションには、シナリオのコネクタの設定パラメータが含まれています。コネクタの設定に関する詳細については、「コネクタの設定」の章を参照してください。

- ピボットドキュメントタイプ

このセクションにはシナリオの各ピボットドキュメントタイプの要素が含まれます。

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

各ピボットドキュメントタイプごとに以下の情報が表示されます。

- ピボットドキュメントタイプ名
- 各要素の名前
- 要素の属性
 - フィールド (ATTRIBUTE)
 - 構造体 (STRUCT)
 - コレクション (ARRAY)
- フィールドのデータ型 : String、Short、Double、Timestamp、など

 **注意:**

シナリオにピボットドキュメントタイプが含まれていない場合、このセクションは表示されません。

- ユーザ定義のドキュメントタイプ

このセクションはユーザが定義する各ドキュメントタイプの要素をまとめています。

各ドキュメントタイプごとに以下の情報が表示されます。

- ドキュメントタイプ名
 - 各要素の名前
 - 要素の属性
 - フィールド (ATTRIBUTE)
 - 構造体 (STRUCT)
 - コレクション (ARRAY)
 - フィールドのデータ型 : String、Short、Double、Timestamp、など
- ドキュメントタイプの作成に関する詳細は、「統合シナリオのインプリメンテーション」章の「生成用または取り込み用ドキュメントタイプ」の節を参照してください。
- ピボットマッピング

このセクションには、シナリオのピボットマッピングの詳細が含まれていません。

 **注意:**

シナリオにピボットマッピングが定義されていない場合、このセクションは表示されません。

- ユーザ定義のマッピング

このセクションは、ユーザが定義するドキュメントタイプをリンクするマッピングの詳細をまとめています。

各マッピングごとに以下の情報が表示されます。

- マッピングの名前

マッピングの名前の後に、コネクタの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が括弧内に表示されます。

- ソースドキュメントタイプの名前
- ターゲットドキュメントタイプの名前
- 各マッピングノードごとに以下の要素が表示されます。
 - ターゲット要素
 - マッピングスクリプト
 - ソース要素
 - マッピングの説明 (必要に応じて)

マッピング作成の詳細については、「[ドキュメントタイプのマッピング](#)」を参照してください。

- 関連するスケジューラとファイル

このセクションの内容は以下の通りです。

- シナリオのスケジューラ
 - スケジューラの以下の情報が記載されます。
 - 名前
 - 周期
 - 関連するコネクタとドキュメントタイプ
- マップテーブルファイル
- Basicスクリプトファイル
- 文字列ファイル

- シナリオ変更の履歴

このセクションには、各シナリオごとに、全変更事項の情報が表示されません。これらの情報は、[\[シナリオの履歴\]](#) ウィンドウ ([\[ファイル/シナリオの履歴\]](#) メニュー) でシナリオ作成者が入力する情報に相当します。

シナリオ文書をHTMLフォーマットで表示する

シナリオビルダには、現シナリオのHTMLフォーマットの文書をインターネットブラウザ内に表示する機能があります。

シナリオ文書をHTMLフォーマットで表示する

- 1 シナリオビルダでシナリオを開くか、または作成します。
- 2 [ファイル/HTMLシナリオ文書を表示する]を選択します。
- 3 インターネットブラウザが起動し、文書が表示されます。

 注意:

[HTMLシナリオ文書を表示する]機能を使用するには、オペレーティングシステムで、HTMLファイル名の拡張子がインターネットブラウザ（例えばMicrosoft Internet ExplorerやNetscape Navigator）に関連付けられている必要があります。

シナリオ文書のプロパティ

シナリオ文書の作成には以下の要素を使用します。

- XSLプロセッサ
Connect-It付属のプロセッサはXalanプロセッサです。
- HTMLフォーマットでの表示に使用するCSSスタイルシート
- SCNファイルをDBKフォーマットに変換するXSLスタイルシート
 - SCNファイルからDBKフォーマットへ
 - DBKファイルからHTMLフォーマットへ

SCNファイルのDBKフォーマットへの変換は、SCNファイルのXMLフォーマットへの変換を経由してから実行されます。

SCNファイルをHTMLフォーマットへ変換する場合、SCNファイルのXMLフォーマットへの変換を経由してから、HTMLフォーマットへ変換されます。

シナリオ文書のフォーマット 変換順序

DBK	1	SCN
	2	XML
	3	DBK
HTM	1	SCN
	2	XML
	3	DBK
	4	HTM

シナリオ文書のプロパティを編集する

シナリオ文書を編集するには、[ファイル/シナリオ文書のプロパティ]を選択してウィザードを起動します。

XSLプロセッサ

ウィザードのこのページでは、SCNファイルをDBKフォーマットとHTMフォーマットに変換するために使用するXSLプロセッサを設定できます。

Connect-It付属のXalanプロセッサは、デフォルトのプロセッサです。

このファイルのパスは「[Connect-Itインストール先フォルダ]
/datakit/doctrans/xalan/xalan.bat」です。

他のXSLプロセッサを使用することも可能です。例：MicrosoftのWebサイトで無料ダウンロード可能なMSXSLプロセッサなど。

XSLプロセッサの実行可能ファイル

このフィールドにXSLプロセッサの完全パスを指定します。デフォルトのパスは、[Connect-Itインストール先フォルダ]/datakit/doctrans/xalan/xalan.batです。

XSLプロセッサのパラメータ

このフィールドのデフォルトのパラメータは以下の通りです。

- %IN
この変数は変換するSCNファイルの完全パスに相当します。
- %XSL
この変数は、DBKまたはHTMフォーマットへの変換に使用するXSLファイルの完全パスに相当します。
- %OUT

この変数は、DBKまたはHTMの出力ファイルの完全パスに相当します。

HTMLシナリオ文書にCSSスタイルシートの使用を許可する

HTMフォーマットでシナリオ文書を表示する際にCSSスタイルシートを使用するためには、このオプションを選択します。

デフォルトで使用されるCSSスタイルシートのパスは、[Connect-Itインストール先フォルダ]/datakit/doctrans/css/default.cssです。

既製XSLスタイルシートのリスト

このページでは、以下の変換に使用するスタイルシートが表示されます。

- XMLファイルからDBKフォーマットへの変換
- DBKファイルからHTMフォーマットへの変換

【説明】と【エイリアスまたはXSLスタイルシート】の列の値を変更するには、値を直接クリックします。


スタイルシートのエイリアスは【拡張子】の列で入力する値です。スタイルシートの完全パスの前にセミコロンで区切ってエイリアスを入力すると（【エイリアス】；【完全パス】）、XSLプロセッサが最初に使用するスタイルシートはこのエイリアスのスタイルシートになります。

拡張子HTM用の値が、DBK;C:\Program Files\Peregrine\ConnectIt\datakit\doctrans\xsl\dbk2htm.xslである場合、XSLプロセッサがXMLフォーマットをまずDBKフォーマットへ変換してから、次にHTMフォーマットへ変換することを意味します。

ユーザ定義のXSLスタイルシートのリスト

このページでは、ファイルのフォーマットを変換するための新規XSLスタイルシートを指定できます。

ユーザ定義のXSLスタイルシートを指定する方法

- 1  をクリックします。
- 2 【拡張子】の列にファイルの拡張子を入力します。
- 3 同じ行の【説明】の列をクリックして、説明を入力します。
- 4 同じ行の【エイリアスまたはXSLスタイルシート】の列をクリックして、XSLスタイルシートのパスを入力します。

他のXSLスタイルシートのエイリアスを指定すると、最初のフォーマットと最終フォーマットの間の中間の変換を必要なだけ指定できるようになります。

シナリオ文書の作成

シナリオビルダでは、文書を以下のフォーマットで作成できます。

- HTMフォーマット
- DBKフォーマット

シナリオ文書の作成

- 1 シナリオビルダでシナリオを開くか、または作成します。
- 2 **[ファイル/シナリオ文書を作成する]**を選択します。
- 3 文書ファイルに名前を付けます。
- 4 **[タイプ]**フィールドで文書のフォーマットを選択します。
- 5 **[保存]**をクリックします。

 **注意:**

シナリオ文書は、デフォルトでシナリオのSCNファイルと同じフォルダ内に保存されます。

DBKフォーマットの文書

DBKフォーマットの文書を作成すると、DocBook DTDでXMLファイルが作成されます。

DocBook DTDの情報については、<http://www.docbook.org> のWebサイトを参照してください。

9 | 統合シナリオをプロダクションモードにする

シナリオをプロダクションモードにする場合、以下の操作により外部アプリケーションとシナリオの統合を自動化することができます。

- シナリオのソースコネクタがいつデータを処理するかを決めるスケジュールを作成する。
- シナリオに関連付けられたWindows 32ビットのサービスを作成し、このサービスがConnect-Itサーバ上でバックグラウンドジョブとして実行されるようにする。
- Connect-Itコンソール経由で、処理中に発生する問題を管理する。

スケジュールの作成

スケジュールにより、ドキュメントを生成するコネクタがいつ起動するのかが指定できるようになります。スケジュールのないシナリオは不完全です。

スケジュールを作成するには、コネクタの生成用ドキュメントタイプをスケジューラに関連付けます。

スケジューラは、コネクタを起動するタイマの役割を果たしています。

- タイマはある時間帯内または時間帯外に、定期的にコネクタを起動します。または、

- ある日時に1回のみ（例えば2003年3月6日）起動します。
- 規則を作成すると、スケジューラを変更できます。例えばあるスケジューラがコネクタを毎日起動するとしします。別の規則を作成すると、ある1日のスケジューラの機能を変更できます。

例：

Desktop Discovery - Asset ManagementシナリオのDesktop Discoveryコネクタは、「Machine」というドキュメントを、毎日朝の9時から夜の10時まで5分おきに生成します。この時間帯以外は、Desktop Discoveryコネクタは1時間おきにドキュメントを生成します。これに別の規則を追加すると、以上のパラメータをある一定の期間用に変更することもできます。

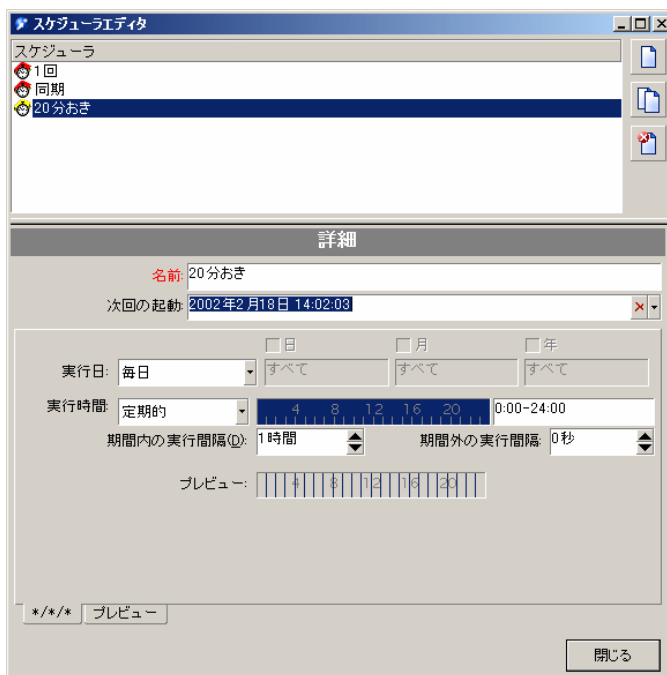
スケジューラの編集

[シナリオ/スケジューラ]メニューを選択すると、スケジューラ編集用ウィンドウが表示されます。

変更不可能なスケジューラが2つ提供されています。

- 「一回」スケジューラ
このスケジューラはドキュメントの生成を一回起動します。生成用ドキュメントタイプを作成すると、デフォルトでこのスケジューラに関連付けられます。
- 「同期」スケジューラ
このスケジューラは、0時から24時まで毎秒ドキュメントの生成を起動します。

図 9.1. スケジューラエディタ



アイコン



機能

- 新規スケジューラの作成を起動します。
- 選択したスケジューラを複製します。
- 選択したスケジューラを削除します。

[実行日] フィールド

このフィールドでは、ドキュメントタイプが生成される日を指定します。
このフィールドには以下のオプションがあります。

- **[毎日]**
年間を通じて毎日、例外なく実行します。
- **[日付指定]**
[日] [月] [年] チェックボックスをオンにすると、1日または複数の日を選択できます。(例：2002年1月6日)
第1、第2、第3、第4、最後から2番目、最終

[日]チェックボックスをオンにすると、曜日を、[月][年]チェックボックスをオンにすると、月と年を指定できます。

例：毎月第1金曜日

[実行時間] フィールド

このフィールドには2つのオプションがあります。

- [定期的]
[定期的] オプションを選択すると、1日のある期間を設定できます。[期間内の実行間隔] フィールドで間隔を指定します。
例：朝4時から夜10時までの期間で、5分おきの実行。
この期間外の実行の間隔は [期間外の実行間隔] で設定できます。
テキストフィールドに値を入力します。手動入力には以下のシンタックスに従います。

```
<開始時> -  
<終了時>, <開始時> - <終了時>...
```

時間入力用のフォーマットは、使用するコンピュータのWindowsコントロールパネルなどで設定する、地域のオプションにより変化します。[AM|PM] オプションパラメータが指定されていないと、時間は24時間制で入力されているものと見なされます。例：「18」と入力すると、6:00 PMという値が確認後に自動的に表示されます。

注意:

グラフィカルエディタを使う場合は、30分単位で指定できます。直接に時間を数値で入力する場合は、1分単位で指定できます。


- [項目リスト]
時間のリストをセミコロンで区切って入力します。これにより、コネクタを選択した時間に起動することが可能になります。

```
5:00AM;8:00PM;...
```

スケジューラの作成 (例)

本節ではコネクタ起動用のスケジューラの作成方法を説明します。

- 毎日
- 朝8時から夜10時の間は10分おき、この期間外では1時間おきに起動
これにドキュメントの生成用の以下の規則を追加します。
- 毎月の第1日曜日

- 朝6時から夜10時の間は10分おき、この期間外では30分おきに起動
スケジューラを作成するには、をクリックします。
スケジューラの詳細下に表示されるタブのフィールドに入力します。

毎日

[実行日] フィールドで [毎日] オプションを選択します。

朝8時から夜10時の間は10分おき、この期間外では1時間おきに起動

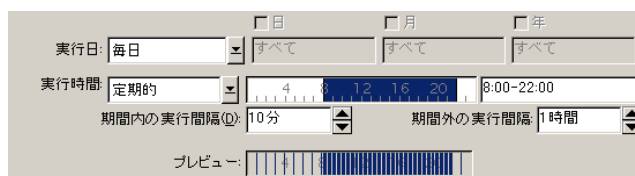
[実行時間] フィールドで [定期的] オプションを選択します。

朝8時から夜10時の期間を選択するには、以下の方法があります。

- テキストフィールドに直接入力します。(8:00 AM-10:00 PMまたは8-22)
- 24時間を表すグラフィカルエディタを使って入力します。

次に、[期間内の実行間隔] と [期間外の実行間隔] フィールドに「10分」と「1時間」を入力します。このフィールドは、選択した期間内と期間外での起動の頻度を設定します。

以上の手順に従うと、タブの内容は以下のようになります。



実行日: 毎日 [すべて] [すべて] [すべて]

実行時間: 定期的 [4] [8] [12] [16] [20] 8:00-22:00

期間内の実行間隔: 10分 期間外の実行間隔: 1時間

プレビュー: 

このタブは、[実行日] フィールドに入力されたデータにより自動的に名前が付けられます。

[作成] をクリックし、スケジューラを確定します。

規則の作成

規則を作成するには、スケジューラエディタのタブ上で右クリックします。ショートカットメニューで [規則の追加] オプションを選択します。

新規のタブが表示されます。スケジュールに追加する規則を入力します。

この例（毎月の第1日曜日、朝6時から夜10時の間は10分おき、この期間外では30分おきに起動）では、タブの内容は以下のようになります。

[作成] をクリックして規則を確定します。

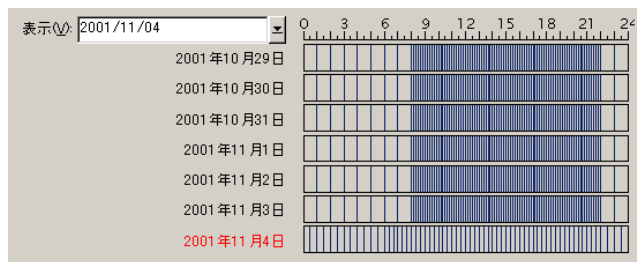
規則の削除

規則を削除するには、規則のタブ内で右クリックします。ショートカットメニューで [規則の削除] オプションを選択します。

プレビュー

[プレビュー] タブでは、現在の週のスケジューラの予定が表示されます。

[表示] フィールドの日付を選択すると、カレンダーは選択した日付の週のスケジューラを表示します。




スケジューラの変更

スケジューラを変更するには、

- 1 リストからスケジューラを選択します。
- 2 タブ内でパラメータを変更します。
- 3 [変更] をクリックして変更事項を確定します。

スケジューラの削除

スケジューラを削除するには、

- リストからスケジューラを選択します。
- [Delete] キーを押すか、または  をクリックします。

スケジュールの編集

スケジュールを編集するには、[シナリオ/スケジュール] メニューを使用します。

シナリオのデフォルトのスケジュールは、生成用ドキュメントタイプを「一回」スケジューラに関連付けています。

生成用ドキュメントタイプをスケジューラに関連付けるには、

- 生成用ドキュメントタイプをクリックし、選択したスケジューラにドラッグします。
- または
- 生成用ドキュメントタイプを選択し、編集用ウィンドウの右側にある矢印を使って移動させます。

スケジューラは、編集用ウィンドウにアルファベット順に表示されます。スケジューラを、シナリオの生成用ドキュメントタイプに関連付けます。

図 9.2. スケジュールの編集ウィンドウ



スケジュール内の生成用ドキュメントタイプの順番

生成用ドキュメントタイプを別のスケジュールへ移動させる

▲	前のスケジュールへ移動する
▼	後のスケジュールへ移動する

同じスケジュール内でドキュメントタイプを移動させる

▲	上に1行移動する
▼	下に1行移動する

データの処理方法は以下の2つの要因に左右されます。

- スケジュール内の生成用ドキュメントタイプの順番
- 同じ生成用ドキュメントタイプ内のマッピングの順番

ドキュメントタイプが並べられている順番により、生成の順番が決定されます。Connect-Itメインウィンドウの【マッピング】タブ内のマッピングの順番も重要です。同じソースドキュメントタイプに複数のマッピングがある場合、【マッピング】タブ内のマッピングの順番により、Connect-Itがどのマッピングを最初に行うかが決まります。






マッピングを並べ替えるには、

- シナリオ図内でマッピングボックスを選択します。
- 【マッピング】タブを選択します。
- リスト内で生成用ドキュメントタイプを選択します。
- 矢印をクリックして、リスト内でドキュメントタイプを移動させます。

例：Desktop Discovery - ServiceCenterシナリオでは、Machine (Machine) ドキュメントタイプに6つのマッピングがあります。

図 9.3. マッピングの順番



アイコン	機能
	マッピングの作成を起動します。
	選択したマッピングを編集できるようになります。
	選択したマッピングを削除します。
	マッピングを上へ1行移動させます。
	マッピングを下へ1行移動させます。

シナリオを起動する際、Connect-Itは同一のスケジューラ内では以下の順番で優先度を決定します。

- 生成用ドキュメントタイプの順番
- 同じ生成用ドキュメントタイプ用のマッピングの順番

ポインタのステータス

多くの場合、ポインタのステータスは特定の時間を指しています。ポインタに提供されるこの情報は非常に大切です。これにより、外部アプリケーションのデータ処理の進み具合を確認できます。

ポインタのタイプは関連する外部アプリケーションにより変わります。

- Asset Managementコネクタ
ポインタは、シナリオ内の各マッピングの最終起動日時に当たります。このポインタの値よりも、最終変更日（ [dtLastModif] フィールド ）が後のレコード全てが処理されます。

注意:

Connect-Itがインストールされているコンピュータと、AssetCenterサーバ間の時間差は自動的に補正されます。

- NTセキュリティコネクタ
スケジューラが起動するたびに、コネクタは補足情報だけでなく、検索するドメインの全情報を取得します。

注意:

新規スケジューラを作成し、そのスケジューラをNTドメインに割り当てる場合、スケジュールのポインタは表示されません。

- Desktop Discoveryコネクタ
使用されるポインタは、既に処理された「.fsf」ファイルの最も最新の作成日です。
- Network Discoveryコネクタ

使用されるポインタはイベントの番号です（EVENTドキュメントタイプの場合）。

- ServiceCenterコネクタ

このコネクタのポインタは、ソースアプリケーションから来るデータの性質により変化します。

- イベントの場合、ポインタは連続番号（evsysseq）に当たります。
ポインタは、シナリオの各マッピングの最終の起動時に処理されたイベントの連続番号に当たります。次回の起動時には、このポインタ値よりも高い連続番号のイベント全てが処理されます。
- テーブルの場合、ポインタの値はマッピングの最終起動日です。次回の起動時には、このポインタよりも、最終変更日（【sysmodtime】フィールド）が後のレコード全てが処理されます。

 注意:

ServiceCenterコネクタの設定により、Connect-Itがインストールされているコンピュータと、ServiceCenterサーバ間の時間差は以下の方法で処理されます。

- 時間差は自動的に補正されます。
- ユーザの入力した時間差に応じて補正されます。

各ポインタのステータスをダブルクリックすると、「.fsf」ファイルの別の日付、またはNetwork Discoveryの別のイベント番号を強制できます。

例

Network Discoveryは数万個のイベントを記憶します。ポインタのステータス（スケジュールの最初の起動前の値は「0」です）を20000にすると、番号が20000より大きなイベントのみがシナリオで処理されます。

「.fsf」ファイルのポインタ

例：

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

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

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


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

スケジュールの編集用ウィンドウ内でのポインタを更新する

シナリオを起動する時にスケジュールの編集用ウィンドウが開いていると、ポインタのステータスは更新されません。スケジュールの編集用ウィンドウのポインタのステータスを更新するには、一度このウィンドウを閉じてから新たに開く必要があります。


スケジュールの起動

スケジュールを起動するには、シナリオを起動する必要があります。シナリオを起動するには、

- [シナリオ / 全スケジューラを起動する] オプションを選択します。
または
-  をクリックします。

スケジュールの停止

スケジュールを停止するには、シナリオを停止する必要があります。シナリオを停止するには、

- [シナリオ / 停止] オプションを選択します。または、
-  をクリックします。

Connect-Itサービスの作成

Connect-Itでは、Windows上のサービスをシナリオに関連付けることができます。このサービスにより、Connect-Itサーバはバックグラウンドジョブとしてデータ処理を実行できます。データ処理は、シナリオに関連付けられたスケジュールに応じて起動されます。

警告:

シナリオ内の1つのコネクタがODBC接続を使用している場合、この接続はシステムデータソース（システムDNS）を使用しなければなりません。この接続がユーザデータソース（ユーザDNS）を使用していると、接続はシナリオに関連付けられたサービスの管理下に置かれなくなります。

Windowsのサービスでは、バックグラウンドジョブとして実行されるアプリケーションを使用できません。Connect-Itでは、シナリオと同じ数だけのサービスを作成、起動できます。

注意:

Windows 32ビットでは、環境変数（例えば検索パスなど）を変更しても、コンピュータを再起動しない限りこの変更事項はサービスに適用されません。

重要項目:

コネクタが他のコンピュータのフォルダやファイルを指定する場合は、マップされたディスク名は使わずに、遠隔のコンピュータ名を直接指定します。例えばディスク「z」が「/FSFStore」にマップされている場合、

```
/FSFStore/Scan
```

が正しい指定方法で、

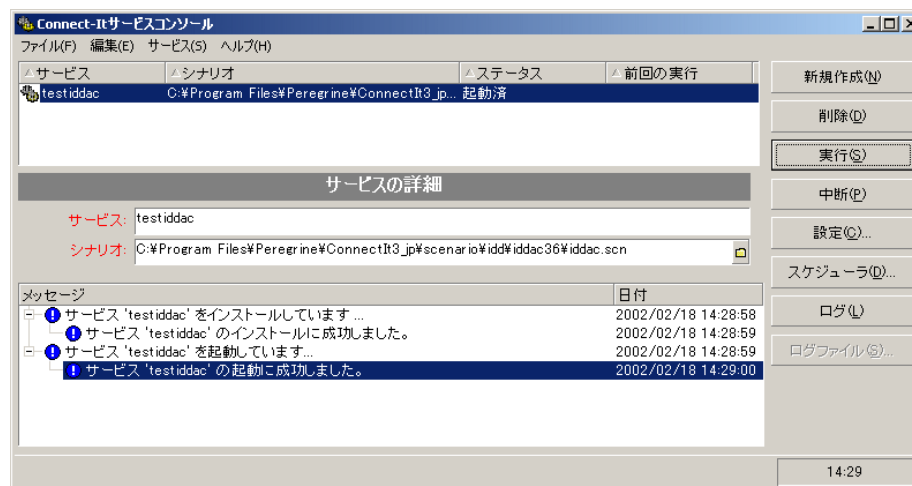
```
Z:/Scan
```

は間違いです。

Connect-Itコンソール

グラフィカルインターフェイスであるConnect-Itコンソールでは、シナリオを管理し、シナリオに関連付けるサービスを作成することができます。

図 9.4. Connect-Itコンソール



コンソールを起動するには、【Connect-Itサービスコンソール】を、Windowsの【スタート】メニュー内のConnect-Itプログラムグループから選択します。または、Connect-Itインストール先フォルダの「bin32」フォルダ内の、「console.exe」実行可能ファイルを起動することもできます。

メニュー

Connect-Itコンソールには4つのメニューがあります。

【ファイル】メニュー

コマンド

終了

機能

Connect-Itコンソールを終了します。

【編集】メニュー

コマンド

切り取り、コピー、貼り付け

機能

コンソール用の標準の編集機能

[サービス] メニュー


コマンド	機能
実行	Connect-Itサービスを起動します。
中断	Connect-Itサービスを停止します。
設定	[コネクタの設定] ウィザードを起動し、シナリオのコネクタの設定を可能にします。
スケジューラ	サービスに関連付けられたシナリオ内で使用されているスケジューラを変更できるようにします。
ログ	シナリオビルダを起動し、 [Connect-Itログ] と [ドキュメントログ] タブを表示します。
ログファイル	作成されたConnect-Itの各サービスに付属するログファイルを表示します。

[ヘルプ] メニュー

コマンド	機能
バージョン情報...	[バージョン情報] ボックスが表示されます。

WindowsでConnect-Itサービスを作成する

Connect-Itサービスを作成するには、

- 1 Connect-Itコンソールを起動します。
- 2 [**新規作成**] をクリックします。
- 3 [**サービス**] フィールドにサービスの名前を入力します。
- 4  をクリックして、シナリオの名前とパスを指定します。
- 5 [**作成**] をクリックします。
コンソールの上部枠に作成されたサービスが表示されます。

シナリオのパス

[**シナリオ**] フィールドにシナリオのパスを入力する場合は、以下の規則に従います。

- 1 シナリオはConnect-Itサーバに保存されている。

シナリオが、ネットワークドライブのフォルダ内にある場合は、ネットワークドライブの文字を【シナリオ】フィールド内に入力しないでください。

例：シナリオ「scenario/myscenario.scn」は、ネットワークドライブR:に関連付けられたフォルダ「C:/Program Files/ConnectIt」内に位置します。この場合、【シナリオ】フィールドには、「C:/Program Files/ConnectIt/scenario/myscenario.scn」を指定し、R:/scenario/myscenario.scnとは入力しないでください。これは、サービスがローカルシステムアカウントまたは特定のアカウントを使用しているためです。これらのアカウントでは、C:/Program Files/ConnectItフォルダは、ネットワークドライブR:に関連付けられていません。

- シナリオはネットワーク上にインストールされている。
サービスが、【シナリオ】フィールド内に指定されているネットワークパスを使用できるかどうか、確認してください。
サービスがネットワークパスを使用できるようにするには、このネットワークパスを使用できるユーザ名とパスワードに、サービスに関連付けます。

UNIXでConnect-Itサービスを作成する

UNIXでConnect-Itサービス（デーモン）を作成するには、

- クライアントのルートディレクトリにある「conit.ini」ファイルを編集します。
- [Service] セクションを作成し、以下の行を追加します。

```
[サービス名]=[シナリオの完全パス]
```

```
[Service]
indsc=/export/home/jean/test/ConnectIt/scenario/ind/indsc.scn
```

例

このUNIXデーモンを起動するには、以下のコマンドラインを実行します。

```
conitsvc -svc indsc &
```

Connect-Itサービスの起動または停止

Connect-Itサービスを起動するには、

- Connect-Itコンソールを起動します。
- Connect-Itコンソールのメイン枠内で、サービスを選択します。
- 【実行】をクリックします。

 **注意:**

サービスを起動するには、シナリオを事前に設定する必要があります。

Connect-Itサービスを停止するには、

- Connect-Itコンソールを起動します。
- Connect-Itコンソールのメイン枠内で、サービスを選択します。
- **[中断]** をクリックします。

Connect-Itサービスの削除

Connect-Itサービスを削除するには、

- Connect-Itコンソールを起動します。
- Connect-Itコンソールのメイン枠内で、サービスを選択します。
- **[削除]** をクリックします。

コマンドライン

コマンドラインからサービスを起動するには、

- 1 Connect-Itのインストール先フォルダの、「bin32」サブフォルダに移動します。

例 : C:\Peregrine\ConnectIt\bin32/

- 2 コマンドラインで以下の行を実行します。

NetStart <サービス名>

例 :

Asset Management-ServiceCenterサービスを起動する場合のコマンドラインは以下の通りです。

NetStart Asset Management-ServiceCenter

シナリオのトラッキングをConnect-Itのコンソールで管理する

Connect-Itコンソールを使用すると、シナリオのトラッキング管理用の以下の操作を実行できます。

- コネクタの設定を変更する
- シナリオのスケジューラを編集する
- Connect-Itログを参照する
- サービスに関連するLOG（ログ）ファイルを参照する

コネクタの設定の変更

作成したサービスに関連するシナリオのコネクタの設定を変更するには、

- 1 シナリオに対応するサービスを選択します。
- 2 **[設定]**をクリックします。
- 3 **[コネクタの本設定]**ウィザードの各ページに変更事項を入力します。
最後のコネクタの設定を終了すると、シナリオビルダは自動的に終了します。

スケジューラの編集

作成したサービスに関連するシナリオのスケジューラを編集するには、

- 1 シナリオに対応するサービスを選択します。
- 2 **[スケジューラ]**をクリックします。
- 3 Connect-Itのスケジューラ編集用ウィンドウが表示されます。
コネクタの編集の詳細については、本章の「**スケジュールの作成**」の節を参照してください。

Connect-Itログの参照

シナリオビルダでは2つのログを参照できます。

- Connect-Itログ
このログは、シナリオの起動時にConnect-Itで実行された全アクション（シナリオの非シリアル化、外部アプリケーションへのコネクタの接続、など）を記述します。

- **ドキュメントログ**
このログでは、シナリオのコンポーネントにより生成された、または取り込まれたドキュメントの詳細を確認できます。このログではまた、ドキュメントの処理中に発生する問題の原因を突き止めることができます。

シナリオビルダのログ用のタブを表示するには、

- 1 シナリオに対応するサービスを選択します。
- 2 **[ログ]**をクリックします。
シナリオビルダが、シナリオ図とConnect-Itログに対応するタブと共に起動します。

サービスのログファイルの参照

Connect-Itサービスが起動するたびに、ログファイルが使用可能になります。ファイルにはサービス名がつけられています（例：「test.log」）。ログファイルを読むには、

- 1 シナリオに対応するサービスを選択します。
- 2 **[ログファイル]**をクリックします。
シナリオのログファイルは、オペレーティングシステムで「.log」ファイルに関連付けられたアプリケーション内で開きます。

シナリオの性能を最適化する

本節では、シナリオのコンポーネントがドキュメント処理に費やす時間を短縮することによって、シナリオの性能を最適化する方法を説明します。

実行結果を使ってドキュメント処理の時間を評価する

シナリオビルダのグラフィカルインタフェースの [Connect-Itログ]、またはログファイルでは、以下の内容に関する実行結果を取得できます。

- ソースコネクタがドキュメントの生成に費やした時間
- マッピングボックスがドキュメントの変換に費やした時間
- ターゲットコネクタがドキュメントの取り込みに費やした時間
- 拒否されたドキュメントの数

例：Asset Managementコネクタの実行結果は、製品のテーブルのレコードに対応するドキュメントが処理されなかったことを示します。

コンポーネントのデータ処理の実行結果を取得する

- 1 シナリオを開きます。
- 2 シナリオの全コネクタを開きます ([Ctrl+F4] キー)。
- 3 シナリオのソースコネクタを選択します。
例 : `idd/iddac36/iddac.scn`シナリオのInfraTools Desktop Discoveryコネクタ
- 4 ▶をクリックします ([F5] キー)。
- 5 ソースコネクタ、ターゲットコネクタ、またはマッピングボックス上にマウスのカーソルを置きます。
[実行結果] 欄を含む状況依存ウィンドウが表示されます。

注意:

これらのデータは、シナリオ図の下にある [Connect-Itログ] タブ内にも表示されます。

実行結果の分析

Connect-Itログ内に表示される以下のデータは、Action Request Systemソースコネクタ、マッピングボックス、LDAPターゲットコネクタを含むシナリオの実行結果です。

```
コネクタ 'Action Request System (fdcitsrv01)' の実行結果 (セッション: 26分 57.310秒 / API: 26分 04.550秒)
取り込まれたドキュメント: 7143
拒否されたドキュメント: 1
挿入されたレコード: 1199
更新されたレコード: 5943
コネクタ 'LDAP (mail-sd.peregrine.com)' の実行結果 (セッション: 47.628秒 / API: 35.765秒)
生成されたドキュメント: 7143
コネクタ 'Mapping (Basic engine)' の実行結果 (セッション: 01.404秒)
評価されたBasic script: 14286
取り込まれたドキュメント: 7143
生成されたドキュメント: 7143
```

作成された実行結果は以下のように分析されます。

- Action Request Systemコネクタによるドキュメントの処理 = 27分、または1秒につき4.4ドキュメント
処理時間のうち、Action Request Systemに関連するAPIにかかった時間が26分で、Connect-Itでの処理にかかった時間は1分です。

APIは、ネットワークの応答時間、「commit」の実行、データベースの健全性の管理などを含みます。

- LDAPコネクタによるドキュメントの処理=48秒、または1秒につき150ドキュメント

処理時間のうち、LDAP APIでかかった時間が36秒で、Connect-Itでの処理にかかった時間は12秒です。

- マッピングボックスによるドキュメントの処理：2秒、または1秒につき5100ドキュメント

この時間は、Connect-Itでのドキュメント処理の時間に相当します。

ドキュメント処理の速度の例

ドキュメント処理の速度の例は以下の通りです。

- データベースコネクタ
1500から2000ドキュメント / 秒
- ServiceCenterコネクタ
450ドキュメント / 秒
- AssetCenterコネクタ
400ドキュメント / 秒

ドキュメントの生成を改善する

コネクタのドキュメント生成を改善するには、ソースアプリケーションから来るデータをより迅速に取得することが必要です。

本節では、ドキュメント生成にかかる時間を短縮する方法を説明します。

ソースデータサーバの最適化

ドキュメント生成の性能は、ソースデータサーバの使用方法とサーバの接続方法に左右されます。

Connect-Itが実行するデータの変換やデータの生成処理はごくわずかです。

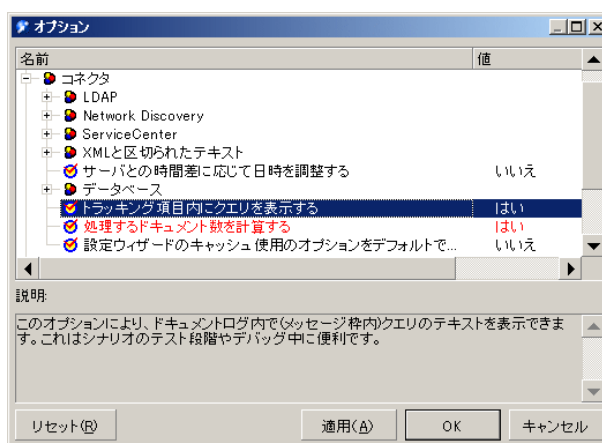
結果を改善するためには、以下の内容を確認する必要があります。

- サーバのデータソースの読み込み
読み込みの機能はサーバの技術的な性能（プロセッサ、使用可能なメモリなど）に左右されます。
- ネットワーク接続（WAN、LAN）

コネクタが実行するクエリの検証

ソースデータベースへ伝達されるクエリをトラッキングするには、

- 1 シナリオビルダを起動します。
- 2 **[編集/オプション]** を選択します。
- 3 **[コネクタ]** 項目を選択して全レベル表示します。
- 4 **[トラッキング項目内にクエリを表示する]** オプションで、**[はい]** を選択します。
- 5 **[OK]** をクリックします。



このオプションが選択されると、シナリオのソースコネクタが実行するクエリに対応するメッセージが、ドキュメントログ内に表示されます。

例

```
SELECT AcctCode,AssetTag,BarCode,dDispos,DisposProfit,dtListPriceCv,Field2,FullName FROM amAsset
```

この情報により、以下の操作が可能になります。

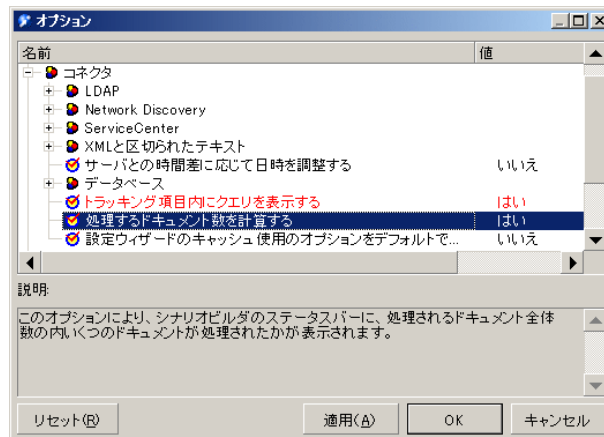
- 指定したドキュメントタイプと生成用ルールに従って、クエリが生成されているかどうかを確認する
- 別のツールを使ってクエリを再起動し、クエリ実行に必要な時間を監視する

生成するドキュメント数の計算

ソースコネクタが生成するドキュメントの数を計算するには、

- 1 シナリオビルダを起動します。

- 2 [編集/オプション]を選択します。
- 3 [コネクタ]項目を選択して全レベル表示します。
- 4 [処理するドキュメント数を計算する]オプションで、[はい]を選択します。
- 5 [OK]をクリックします。



このオプションを使用すると、ソースコネクタが処理するドキュメントの数を表示する進行状況バーが、シナリオビルダ内に表示されます。

図 9.5. ドキュメントの処理 - 進行状況バー



Asset Managementコネクタで、このオプションはSELECT COUNT型のクエリを起動します。

```
SELECT COUNT(AcctCode) FROM amAsset
```

例

警告

このクエリの実行には時間がかかるため、シナリオのプロダクションモードでは、このオプションをオフにしてください。

コネクタのキャッシュオプションの使用

データベース型のコネクタでは、生成用ドキュメントタイプ（メタデータ）の構造にキャッシュを使用することが推奨されています。

キャッシュを使用すると、生成用ドキュメントタイプ（メタデータ）の記述の読み込みが、ローカルコンピュータで実行されるため、コネクタがより速く開くようになります。

コネクタのキャッシュ使用の詳細については「コネクタの設定」章の「キャッシュを設定する」の節を参照してください。

キャッシュを使用しない場合、コネクタは、コネクタを開く度にソースデータベースの記述に対応するデータ全体を取得します。

データベース型コネクタでは、データベース記述は次のデータに当たります。

- テーブルのリスト
- フィールドのリスト
- 書式のリスト
- インデックスのリスト
- リンクのリスト
- 結合のリスト
- その他

上記のデータが多数であるほど、コネクタを開く際に時間がかかります。

注意:

コネクタを開く時に使用可能な帯域幅が低い場合、かかる時間は長くなります。これはネットワークの接続（WAN、LAN）の性能に左右されます。

警告

シナリオのプロダクションモードでは、キャッシュの使用が推奨されています。しかし、ソースデータベースの記述内容が変更した場合は、コネクタのキャッシュの同期をとらなければなりません。

コネクタのキャッシュの同期をとるには、

- 1 シナリオビルダを起動します。
- 2 シナリオを開きます。
- 3 キャッシュの同期をとる必要があるコネクタを選択します。
- 4 コネクタを開きます（[F4]キー）。
- 5 **[ツール/キャッシュ/キャッシュの同期をとる]**を選択します。

キャッシュの同期をとった後、ソースデータベース記述の新規データに存在しない要素が、マッピングに含まれていないかどうか確認します。

自動再接続

データソースとして遠隔のサーバを使用するコネクタでは、サーバへの自動再接続のオプションを選択する必要があります。

自動再接続のオプションは、ネットワークでの待ち時間（データパケットの伝達）に多大な影響を与えます。

自動再接続のオプションについては、「コネクタの設定」章の「再接続のパラメータを設定する」の節を参照してください。

再接続に関する情報は、Connect-Itログ内に表示されます。

Action Request Systemコネクタの再接続試行の例

```
サーバへ接続中...
ダイナミックライブラリ 'arapi50.dll' の使用。
(ARS ERROR 90) ARシステムサーバへネットワーク接続できません。 fcitsrv0
1 : RPC: Name to address translation failed - No such hostname
サーバへの次回の再接続は4秒後に試行されます。
再接続を試行しています...
ダイナミックライブラリ 'arapi50.dll' の使用。
(ARS ERROR 90) ARシステムサーバへネットワーク接続できません。 fcitsrv0
1 : RPC: Name to address translation failed - No such hostname
サーバへの次回の再接続は8秒後に試行されます。
再接続を試行しています...
ダイナミックライブラリ 'arapi50.dll' の使用。
(ARS ERROR 90) ARシステムサーバへネットワーク接続できません。 fcitsrv0
1 : RPC: Name to address translation failed - No such hostname
サーバへの次回の再接続は16秒後に試行されます。
```

ネットワークへの接続が不安定な場合、待ち時間は長くなります。

一定期間以上経った接続を切断するルータ/ファイアウォール/サーバ型の接続設定には、注意が必要です。この種のパラメータについては、ローカルネットワークの担当部署で確認してください。

各セッション毎の再接続（Asset Managementコネクタ用）

Asset Managementコネクタ専用の設定オプションを使用すると、シナリオの各セッションの終了時に接続を切断し、新規セッションの開始時のみに接続を再開するように設定できます。

Connect-Itサーバのリソースが保存されるほど、サーバへより速く再接続されるようになります。

このオプションの詳細については、「Peregrine Systemsコネクタ」の章の「Asset Managementコネクタ」の節、「Asset Managementコネクタの設定」を参照してください。

警告

シナリオの実行頻度が高い場合（5分おきなど）は、このオプションを無効にする必要があります。

スケジュールを使ったシナリオ実行の頻度設定については、「統合シナリオをプロダクションモードにする」章の「the section called “スケジュールの作成”」の節を参照してください。

WHERE句の使用の最適化

データベース型コネクタの生成用ルールでは、WHERE句を使用できません。

WHERE句作成に関する詳細は、「コネクタのルール（ディレクティブ）」章の「生成用ルール」の節の「WHERE句とORDER BY句」を参照してください。

WHERE句の使用を最適化するには、句の適用先のフィールドがインデックス化されている必要があります。WHERE句がインデックス化されていないフィールドに適用されると、句を実行する際にテーブルの全レコードが検索されることとなります。

ポインタの管理

シナリオが処理するデータ量を少なくするために、ポインタを使用することも可能です。

ポインタがレコードの最終変更日である場合、コネクタは最後の起動以降に作成または更新されたレコードのみを処理します。

例：ソースコネクタは、[従業員]テーブルのレコードのうち、企業の新入社員に対応するレコードのみを処理します。

ポインタの使用方法については、「コネクタの設定」章の「スケジュールのポインタを設定する」の節を参照してください。

一般的に、ポインタのステータスは特定の日時を指します。日時は、シナリオのスケジュール用ウィンドウ（[シナリオ/スケジュール]メニュー）で変更できます。

シナリオのポインタがインデックス化されていないフィールドを使用すると、データの処理にかかる時間は長くなります。処理時間を短縮するには、ポインタとして使用するフィールドを、インデックス化することが推奨されています。

ドキュメントの取り込みを改善する

コネクタのドキュメント取り込みを改善するには、ターゲットアプリケーションでデータをより迅速に処理することが必要です。

本節では、ドキュメントの取り込みにかかる時間を短縮する方法を説明します。

識別キーの選択

整合性チェックは旧データと新規データの比較に当たります。データベース型のコネクタ (Asset Management) では、コネクタが作成または更新するテーブルレコードを一意的な方法で識別するためのフィールドを定義することが、整合性チェックに相当します。

識別キーとしてどのフィールドを選択するかは、ターゲットコネクタのドキュメントの取り込みに大きく影響します。

ドキュメントの取り込みは、以下の2つのアクションが実行されることを意味します。

- 1 識別キーとして選択されたフィールドを使ってクエリを送信し、ターゲットアプリケーションにレコードが存在するかどうかを確認します。
- 2 レコードの挿入または更新のアクションが実行されます。

識別キーとして選択されるフィールドがインデックス化されていない場合は、ドキュメントの取り込み速度は遅くなります。例：memoやFeatParamフィールドが、Asset Managementコネクタ用の識別キーとして選択されていると、ドキュメントの取り込みの効率は悪化します。


トランザクションの管理

データベース型コネクタは、1ドキュメントを取り込むたびに、デフォルトで「コミット」を実行します。

使用する帯域幅が低い場合、ドキュメントのグループごとにコミットを実行することも可能です。

何個のドキュメントごとにコミットするかを指定するには、

- 1 シナリオビルダを起動します。
- 2 シナリオを開きます。
- 3 トランザクションのパラメータを変更する必要があるコネクタを選択します。
- 4 コネクタの設定ウィザードを起動します ([F2] キー)。

注意：コネクタの高度な設定が有効になっている (ツールバーの  アイコンがオンになっている) かどうか確認します。

- 5 [次へ] を数回押して、[トランザクションを管理する] ページを開きます。

- 6 [ドキュメントグループごとにコミットする] オプションを選択します。
- 7 ドキュメント数を記入して、いくつかのドキュメントごとにコミットするかを指定します。

莫大な量のドキュメントを取り込む場合は、多数のドキュメントグループごとのコミットが推奨されています。

シナリオ `ldap/ac/complete.scn` の性能向上の例

- 1 各ドキュメントごとのコミット (デフォルトのオプション)
性能: 6000ドキュメント用に4分 (25ドキュメント/秒)
- 2 500ドキュメントのグループごとのコミット
性能: 6000ドキュメント用に3分 (33ドキュメント/秒)

同期処理と非同期処理 - 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へ送信されるクエリに相当します。

ドキュメントの取り込み速度を上げるには、以下のどちらかを選択する必要があります。

- データの同期処理
取り込まれる各ドキュメントは、ターゲットアプリケーションが前のドキュメントを処理した後に送信されます。
- データの非同期処理
取り込まれる各ドキュメントは、ターゲットアプリケーションが前のドキュメントを処理していなくても送信されます。

Connect-Itエンジンをを使ってドキュメント処理を改善する

本節では、Connect-Itエンジンをを使ってドキュメント処理の時間を短縮する方法を説明します。

ドキュメントログの設定

Connect-Itのドキュメントログでは、以下の要素を指定できます。

- ドキュメントログ内に表示するエラーの種類 ([フィルタ] フィールド)
- Connect-Itサーバのメモリに保存されるトラッキング項目の最大数
- ドキュメントログのメッセージを保存するテキストファイル

データ処理にかかる時間を短縮するには、

- 1 シナリオビルダを起動します。
- 2 **[ログ/ドキュメントログを設定する]** を選択します。
- 3 **[フィルタ]** フィールドで1項目を選択し、ドキュメントログ内に保存するエラーの種類を限定します。
例：拒否
- 4 保存するトラッキング項目の数を制限します。
トラッキング項目の最大数は5000です。
[ファイルを使用] オプションを使用しない場合、Connect-Itエンジンのリソースを節約できます。

非グラフィカルモードの使用

シナリオの進行中にシナリオビルダが表示されていると、コマンドが選択されるたび（更新、ドキュメントログの閲覧、新規オプションの入力など）にConnect-Itエンジンは減速します。

スケジュールの規則

Connect-Itのスケジュールは、シナリオの生成用ドキュメントタイプに1つまたは複数のスケジューラを関連付けることによって作成されます。例：あるスケジューラはAsset Managementコネクタを1時間ごとに起動します。起動するごとに、コネクタはAsset Managementアプリケーションのレコードに対応するドキュメントを生成します。

コネクタが起動するたびに、Connect-Itは処理するデータの量を計算し、データ全体が処理されるまでConnect-Itは停止しません。処理するデータ量に応じて、シナリオのスケジューラを調整する必要があります。大規模なデータベースのマイグレーションの場合は、稼働率が低い時期（夜間のスケジュールなど）にソースコネクタの起動をスケジュールすることが推奨されています。少量のデータを処理する統合シナリオの場合は、既製の「同期」スケジューラの使用が推奨されています（ソースコネクタが毎秒起動）。

スケジュールの詳細については、本章の「[スケジュールの作成](#)」の節を参照してください。

シナリオの並行起動

1つのシナリオが多数のデータを移行する場合、シナリオを複数のシナリオに分割し、各シナリオを別々のサービス（Windows）またはデーモン（UNIX）で起動させる方法が推奨されています。

サービスとデーモンの詳細については、本章の「[Connect-Itサービスの作成](#)」の節を参照してください。

例

データベースに記録された従業員のリストを、別のデータベースにシナリオがインポートするとします。

移行にかかる時間を縮小するには、以下の2つのシナリオを作成します。

- 1番目のシナリオは、姓のイニシャルがAからJである従業員のリストを移行します。
- 2番目のシナリオは、姓のイニシャルがKからZである従業員のリストを移行します。

従業員を選択するには、2つのシナリオのソースコネクタで生成用ルール（WHERE句）を作成します。

WHERE句の作成に関する詳細は、「コネクタのルール（ディレクティブ）」章の「生成用ルール」の節の「WHERE句とORDER BY句」を参照してください。

Asset Managementコネクタを使用するシナリオの性能を向上させる

本節では、Asset Managementコネクタを使うシナリオで、ドキュメントの処理にかかる時間を短縮する方法をいくつか紹介します。

dtLastModifフィールドのインデックス化

Asset Managementコネクタを使用するシナリオでは、シナリオに関連するAssetCenterの全テーブルの【dtLastModif】フィールドにインデックスを追加しなければなりません。Connect-Itはこのフィールドを一貫して使用して、最終セッション後に作成または変更されたレコードを確認します。

資産のテーブル（amAsset）、製品のテーブル（amProduct）、従業員と部署のテーブル（amEmpDept）でインデックスを作成するには、以下のコマンドを実行します。

```
CREATE INDEX Ast_dtLastModif ON amAsset (dtLastModif)
GO
CREATE INDEX Prod_dtLastModif ON amProduct (dtLastModif)
GO
CREATE INDEX EmpDept_dtLastModif ON amEmpDept (dtLastModif)
GO
```

【dtLastModif】フィールドがインデックス化されているかどうかを確認するには、実行可能ファイル「adblog」を使用します。このファイルを使用すると、WHERE句で【dtLastModif】フィールドにフィルタを適用するSQLクエリの実行を検証できます。

Asset Managementデータベース用の調整 - Sybase ASE エンジン

クエリのFROM部分に多数のテーブルがあるため、クエリの実行に時間がかかるという内容がLOGファイルに表示される場合は、Sybase Query Optimizerアプリケーションを使用して、処理時間を短縮することが推奨されています。

「amdb.ini」ファイルの変更

SQLクエリの処理方法を改善するには、Asset Managementアプリケーションの「amdb.ini」ファイルに以下の行を追加することも可能です。

```
PostConnectSql= set forceplan on
```

以下の行は、DB ASE COPPERデータベース用の「amdb.ini」ファイルの設定です。

```
[DB ASE COPPER]
PostConnectSql=set forceplan on
stmtcache=500
LongDesc=
Engine=Sybase
Location=COPPER
EngineLogin=itam
EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC641ED76CDAA17050EA1A124254200200
ReadOnly=0
CacheDir=
CacheSize=5120000
Base=EDS
Owner=
TableSpace=
TableSpaceIndex=
AmApiDll=aamapi35.dll
UseNTSecurity=0
```

これらのパラメータは、Asset Managementデータベースサーバの「amdb.ini」ファイル内で設定されなければなりません。

Asset Managementアプリケーションのクライアント部分が、Connect-Itサーバにインストールされている場合は、同じSybaseデータベースに関連付けられた2つの別の接続を確立できます。

- 1番目の接続はオプション「PostConnectSql=set forceplan on」と「stmtcache=500」を使用します。

```
[DB ASE ConnectIt]
PostConnectSql=set forceplan on
stmtcache=500
LongDesc=
Engine=Sybase
Location=COPPER
EngineLogin=itam
EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC64
1ED76CDAA17050EA1A124254200200
ReadOnly=0
CacheDir=
CacheSize=5120000
Base=EDS
Owner=
TableSpace=
TableSpaceIndex=
AmApiDll=aamapi35.dll
UseNTSecurity=0
```

- 2番目のオプションはこれらのパラメータを使用しません。

```
[DB ASE COPPER ACGUI]
LongDesc=
Engine=Sybase
Location=COPPER
EngineLogin=itam
EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC64
1ED76CDAA17050EA1A124254200200
ReadOnly=0
CacheDir=
CacheSize=5120000
Base=EDS
Owner=
TableSpace=
TableSpaceIndex=
AmApiDll=aamapi35.dll
UseNTSecurity=0
```

従来のAssetCenterクライアント（Sybase ASE以外）で、オプション「stmtcache=500」を使用してはなりません。

従来のAssetCenterクライアントでの処理性能に問題がある場合は、以下のオプションのいずれかを使用できます。

- PostConnectSql=set forceplan on
- PostConnectSql=set table count 3

- PostConnectSql=set table count 2

データベース型コネクタを使用するシナリオの性能を向上させる

Sybaseネイティブ接続を使用するデータベース型コネクタの場合、
「PostConnectSql=set forceplan on」オプションを詳細オプションで入力すると、
SQLクエリ実行の性能が向上する可能性があります。

詳細オプションについては、「コネクタの設定」章の「詳細オプション」の節を参照してください。

10 | 処理レポート

処理レポートは、コネクタまたはマッピングボックスが、ドキュメントを取り込む度に生成することができるドキュメントです。

各処理レポートでは、ドキュメントが適切に処理されたかどうかを確認できません。正常に処理されたドキュメントとは、拒否された要素が全くないドキュメントのことを指します。

ソースコネクタが処理レポートを使用するためには、ソースコネクタは、SuccessReportドキュメントタイプを使用可能なドキュメントタイプとして発行しなければなりません（InfraTools Desktop Discoveryコネクタでは、DirectoryPoolerActionと呼ばれています）。

ProcessReport処理レポートと、SuccessReportドキュメントタイプ間のマッピングにより、ソースコネクタは、自分がドキュメントに転換したデータの抽出元のファイルに、特定のアクションを実行することができます。

注意:

更に処理レポートのフィールドを、Eメール（送信）コネクタの取り込み用ドキュメントタイプにマップすることにより、EメールメッセージがConnect-Itの管理者に送信されるようになります。

処理レポートの内容

処理レポートは、各コネクタとマッピングボックスの使用可能なドキュメントタイプとして存在します。処理レポートに含まれる情報の詳細は以下の表の通りです。

表 10.1. 処理レポートの内容

要素	情報	フィールドタイプ
ProcessReport	処理レポートのルートノード	
DocumentType	取り込み用ドキュメントタイプ	テキスト
DocumentTypeID	ドキュメントの固有識別子	テキスト
ErrorNumber	ドキュメントの取り込み中に発生したエラーの番号	整数 (32ビット)
Success	ドキュメントの正常な処理 (値: 1) または正常でない処理 (値: 0) 正常でない処理は、ドキュメントの一部または全体の拒否を意味します。	ブール
WarningNumber	ドキュメントの取り込み中に発生する警告の数	整数 (32ビット)
Logs	処理の失敗を説明するメッセージに対応するコレクション	
Date	メッセージの日付	日付と時刻
LogType	メッセージのタイプ 可能な値は以下の通りです。	整数 (32ビット)
	<ul style="list-style-type: none"> エラー 警告 情報 追加情報 	
Msg	メッセージの内容	テキスト
Path	メッセージに関連するドキュメントのパス	テキスト

既製シナリオ内で処理レポートを使用する

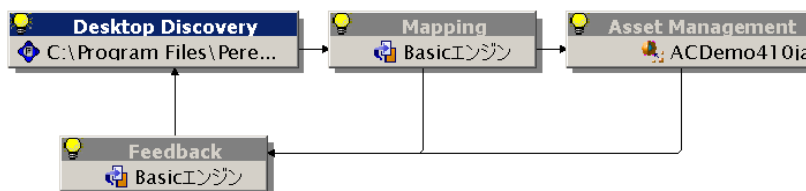
Connect-It付属の既製シナリオでは、ソースコネクタに使用されるファイル上に実行するアクション（削除、移動）を決定するために、処理レポートが使用される場合もあります。

処理レポートをシナリオに統合するためには、まず処理レポートを使用している以下のシナリオを試行することをお勧めします。

- InfraTools Desktop Discovery - Asset Managementシナリオ
- XML (MQSeries) - Asset Managementシナリオ

InfraTools Desktop Discovery - Asset Managementシナリオでの処理レポートの使用

図 10.1. InfraTools Desktop Discovery - Asset Managementシナリオ図



このシナリオでは、feedbackというマッピングボックスは、MappingマッピングボックスとAsset Managementコネクタにより生成された処理レポートを、InfraTools Desktop Discoveryコネクタに送ります。

このマッピングボックスの中には2つのマッピングがあります。

- Mappingマッピングボックスの処理レポートと、InfraTools Desktop Discovery ソースコネクタのDirectoryPoolerActionドキュメントタイプ間のマッピング
- Asset Managementコネクタの処理レポートと、InfraTools Desktop Discovery ソースコネクタのDirectoryPoolerActionドキュメントタイプ間のマッピング

表 10.2. Mappingマッピングボックスに生成された処理レポートと、InfraTools Desktop Discoveryコネクタに取り込まれるDirectoryPoolerActionドキュメントタイプ間のマッピングの詳細

DirectoryPoolerAction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

DirectoryPoolerAction (ルートノード)	if [Success] = 1 then PifIgnoreDocumentMapping end if
-----------------------------------	---

コメント: ドキュメントが正常に取り込まれた場合、マッピングスクリプトは、マッピングボックスが処理レポートを生成しないようにします。

このシナリオでは、ドキュメントの処理の成功は、Asset Managementターゲットコネクタによるドキュメントの正常な処理に、従属しています。

FileName	[\$ParentDoc\$.FileInfo.FileName]
----------	-----------------------------------

InfraTools Desktop Discoveryコネクタにより生成されるMachineドキュメントタイプの「FileInfo.FileName」フィールドを取得できるようにします。このフィールドは、コネクタに処理されるFSFファイルの名前に一致します。

Success	Success
---------	---------

コメント: プールフィールド

- 値「0」は、ドキュメントの一部または全部がマッピングボックスにより拒否されたことを意味します。
- 値「1」は、ドキュメントがマッピングボックスにより正常に処理されたことを意味します。

表 10.3. Asset Managementコネクタに生成される処理レポートと、InfraTools Desktop Discoveryコネクタに取り込まれるDirectoryPoolerActionドキュメントタイプ間のマッピングの詳細

DirectoryPoolerAction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

DirectoryPoolerAction (ルートノード)	
FileName	[\$ParentDoc\$.ParentDoc\$.FileInfo.FileName]

コメント: InfraTools Desktop Discoveryコネクタにより生成されるMachineドキュメントタイプの「FileInfo.FileName」フィールドを取得できるようにします。このフィールドは、コネクタに処理されるFSFファイルの名前に一致します。

Success	Success
---------	---------

DirectoryPoolerActionドキュメントタイプの要素

コメント：プールフィールド

- 値「0」は、ドキュメントの一部または全部がAsset Managementコネクタにより拒否されたことを意味します。
- 値「1」は、ドキュメントがAsset Managementコネクタにより正常に処理されたことを意味します。



注意:

\$ParentDoc\$変数の使用

- 表記法 [\$ParentDoc\$. フィールド] により、処理されるドキュメントの親ドキュメント内のフィールド値を、取得できるようになります。
- 表記法 [\$ParentDoc\$. \$ParentDoc\$. フィールド] により、処理されるドキュメントの親ドキュメントの親ドキュメント内のフィールド値を、取得できるようになります。
- その他

InfraTools Desktop Discoveryコネクタに処理されたFSFファイルへの処理レポートの影響

処理レポートにより、InfraTools Desktop Discoveryコネクタは、処理レポートに関連するドキュメントに対応するFSFファイルに特定のアクションを実行できるようになります。処理される各ドキュメントのFileInfo.FileNameフィールドで、ドキュメントに対応するFSFファイルが識別されます。

特定のアクションは、InfraTools Desktop Discoveryコネクタの設定時に選択できません。

例：

FSFファイルの管理オプションは以下の通りです。

- ドキュメントの処理に成功した場合、ドキュメントに対応するFSFファイルは元のフォルダから削除されます。
- ドキュメントの処理に失敗した場合、ドキュメントに対応するFSFファイルは、例えば「エラー」と名前を付けたフォルダに移されます。

シナリオでドキュメントの処理を実行した後、「エラー」フォルダを開ければ、Asset Managementアプリケーションに適切に転送されなかったデータを含む全てのFSFファイルを、見つけることができます。

ドキュメントの一部または全部が拒否される時に起こった問題を確認するには、シナリオビルダのドキュメントログを参照してください。

Asset Managementコネクタにドキュメントが適切に処理されると、処理レポート（Successフィールドの値は「1」）は、InfraTools Desktop Discoveryコネクタが、ドキュメントに対応するFSFファイルを元のフォルダから削除するようにします。

11 | Connect-ItのJava開発キット (JDK)

Connect-ItのJava開発キットを使用すると、ユーザは独自のコネクタを開発できます。

この開発キットは、XMLテクノロジーとJava Connector Architecture (JCA) 1.0規格を使用しています。

この規格の詳細については、Webサイト <http://java.sun.com/j2ee/connector/> を参照してください。

Java開発キットの内容

Connect-ItのJava開発キットのパッケージ内容は以下の表の通りです。

表 11.1. Connect-ItのJava開発キットの内容

パッケージ	機能
com.peregrine.common.assert	アサーション機構 (デバッグモード)
com.peregrine.common.collection	コレクションのクラス

パッケージ	機能
com.peregrine.common.ids	STRファイル処理用のツールを提供します。アプリケーションのストリングの場所を見つけるのに便利です。
com.peregrine.common.io	ファイル管理用の相対クラス
com.peregrine.common.log	ログファイル (LOG) のメッセージを作成し配信する機構を適用します。
com.peregrine.common.text	文字列のフォーマット用ツール
com.peregrine.common.thread	スレッドの管理
com.peregrine.common.util	多種のツール
com.peregrine.common.xml	XML操作
com.peregrine.conit.core	Connect-Itフレームワークの中核
com.peregrine.conit.document	Connect-Itが使用するUnified Document Content規格
com.peregrine.conit.document.description	Connect-It XMLメタデータ
com.peregrine.conit.document.schema	XMLSchemaサポートのAPI
com.peregrine.conit.ra	JCAコネクタ
com.peregrine.resourceimpl.cci	CCIクラスの実装
com.peregrine.resourceimpl.event	イベントクラスの実装 (EVENT)
com.peregrine.resourceimpl.spi	SPIクラスの実装
com.peregrine.resourceimpl.tools	導入ツール
com.peregrine.resourceimpl.util	JCA実装用のヘルパー

JCA規格の実装

本節は以下の3点について説明します。

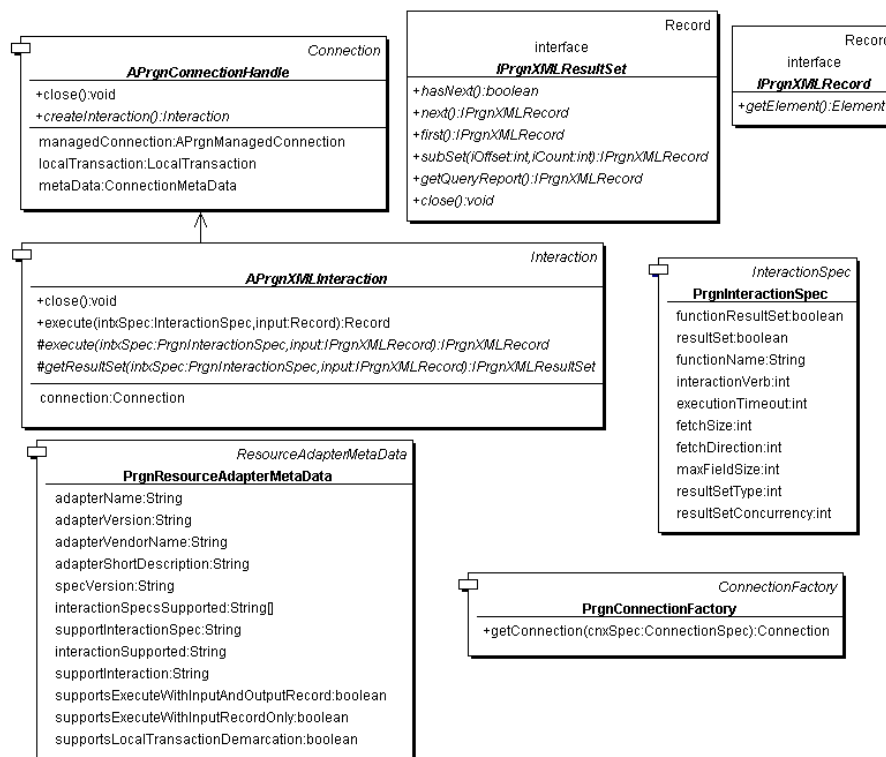
- CCI (Common Client Interface) クラスの実装
- SPI (Service Provider Interface) クラスの実装
- EVENTクラスの作成によるJCA規格の拡張

CCIクラスの実装

インタフェース	javax.resource.cci.Connection
実装されるクラス	com.peregrine.resourceimpl.cci.APrgnConnectionHandle
インタフェース	javax.resource.cci.ConnectionFactory
実装されるクラス	com.peregrine.resourceimpl.cci.PrgnConnectionFactory

インタフェース	javax.resource.cci.ConnectionMetaData
実装されるクラス	com.peregrine.resourceimpl.spi.APrgnManagedConnectionMetaData
インタフェース	javax.resource.cci.Interaction
実装されるクラス	com.peregrine.resourceimpl.cci.APrgnXMLInteraction
インタフェース	javax.resource.cci.InteractionSpec
実装されるクラス	com.peregrine.resourceimpl.cci.PrgnInteractionSpec
インタフェース	javax.resource.cci.ResourceAdapterMetaData
実装されるクラス	com.peregrine.resourceimpl.cci.PrgnResourceAdapterMetaData
インタフェース	javax.resource.cci.Record
実装されるクラス	com.peregrine.resourceimpl.cci.IPrgnXMLRecord
	com.peregrine.resourceimpl.cci.IPrgnXMLResultSet
インタフェース	javax.resource.cci.LocalTransaction
実装されるクラス	com.peregrine.resourceimpl.spi.APrgnLocalTransaction

図 11.1. Java開発キット - CCIクラス



本節ではJCA規格の主要なクラスを説明します。

javax.resource.cci.Connection

このクラスは、物理的な接続を確立するために必要なクライアントのアプリケーションレベルのハンドルを表しています。

実装

APrgnConnectionHandleクラスは抽象的なクラスです。このクラスにより、ManagedConnectionオブジェクト経由で、物理的な接続へアクセスできるようになります。ManagedConnectionインスタンスが作成するブリッジにより、異種のクライアント間での物理的なリソースの共有が可能になります。これら2レベルの一貫性を保つために、接続の確認機構が付属しています。

javax.resource.cci.ConnectionFactory

これはクライアントコンポーネントのエントリポイントで、`getConnection()`の2メソッドのうち1つを経由して接続を確立します。

実装

`PrgnConnectionFactory`は、即時に使用可能な具体クラスです。非管理環境では、このクラスは`PrgnNonManagedConnectionManager`オブジェクトに代表される`ConnectionManager`インターフェイスのインスタンスを、デフォルトで作成します。

javax.resource.cci.ConnectionMetaData

このクラスは、`Connection`タイプの接続経路でシステムに関する情報を提供します。これは`ManagedConnectionMetaData` SPIインターフェイスクライアントと対をなします。

実装

選択された基本的実装は「インタフェースプログラミング」契約に基づきます。抽象クラス`APrgnManagedConnectionMetaData`は、`cci`と`spi`インタフェースで記述された役割をまとめます。

javax.resource.cci.LocalTransaction

このクラスはアプリケーションレベルのトランザクションを表します。このインタフェースは、SPIインタフェース`LocalTransaction`に関連しています。

実装

`cci`と`spi`の役割は、`Service Provider Interface`パッケージのベースクラス`APrgnLocalTransaction`内にまとめられています。

javax.resource.cci.ResourceAdapterMetaData

このクラスはコネクタに対応する情報を説明します。

実装

`PrgnResourceAdapterMetaData`クラスは具体クラスで、インタフェースに記述されるプロパティのアクセッサ (`getXXX`と`setXXX`メソッド) を提供します。

javax.resource.cci.Record

このクラスは、システムとの相互作用時の入力データと出力データを表します。

実装

実装フレームワークにサポートされている`Record`タイプの全オブジェクトは、XML規格に基づいています。以下の2つのタイプがあります。

- IPrgnXMLRecord
単一XMLドキュメントに相当します。
- IPrgnXMLResultSet
XMLドキュメント全体のイテレータに相当します。

SPIクラスの実装

インタフェース	javax.resource.spi.ConnectionEventListener
実装されるクラス	com.peregrine.resourceimpl.spi.APrgnConnectionEventListener

インタフェース	javax.resource.spi.ConnectionManager
実装されるクラス	com.peregrine.resourceimpl.spi.PrgnNonManagedConnectionManager

インタフェース	javax.resource.spi.ConnectionRequestInfo
実装されるクラス	com.peregrine.resourceimpl.spi.PrgnConnectionRequestInfo

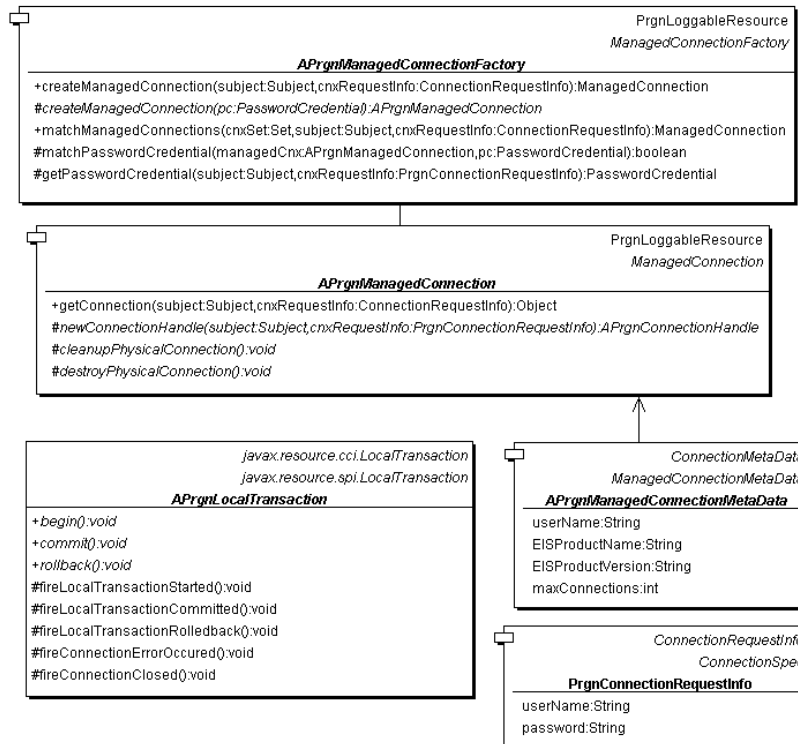
インタフェース	javax.resource.spi.LocalTransaction
実装されるクラス	com.peregrine.resourceimpl.spi.APrgnLocalTransaction

インタフェース	javax.resource.spi.ManagedConnection
実装されるクラス	com.peregrine.resourceimpl.spi.APrgnManagedConnection

インタフェース	javax.resource.spi.ManagedConnectionFactory
実装されるクラス	com.peregrine.resourceimpl.spi.APrgnManagedConnectionFactory

インタフェース	javax.resource.spi.ManagedConnectionMetaData
実装されるクラス	com.peregrine.resourceimpl.spi.APrgnManagedConnectionMetaData

図 11.2. Java開発キット - SPIクラス



javax.resource.spi.ConnectionEventListener

このクラスは、クライアント接続に関するManagedConnectionインスタンスの通知を受信できるようにします。

実装

抽象クラスAPrgnConnectionEventListenerは、インタフェースで記述されるメソッドを実装します。これらのメソッドは空のため、複数の適切な機能をオーバーロードできません (design pattern 'adapter')。

javax.resource.spi.ConnectionManager

プールやトランザクションを管理するために、コネクタとアプリケーションサーバをリンクできるようにします。

実装

デフォルトの具体的な実装はコネクタ（非管理環境では必須）に提供されています。プール機能は追加されます。

javax.resource.spi.ConnectionRequestInfo

接続要求のフローの一部である特定のクライアントの、プロパティの集合です。

実装

PrgnConnectionRequestInfoは、**ユーザ名とパスワード**のプロパティをサポートします。特定のコネクタを実装すると他のプロパティも追加できます。

javax.resource.cci.ConnectionSpecクラスの実装も可能にします。

javax.resource.spi.LocalTransaction

ローカルトランザクションの管理をサポートできるようにします。

実装

APrgnLocalTransactionは、APrgnManagedConnectionインスタンスにリンクしており、トランザクション通知用のメソッドも提供します。

javax.resource.spi.ManagedConnection

EISインスタンスの物理的な接続を表します。

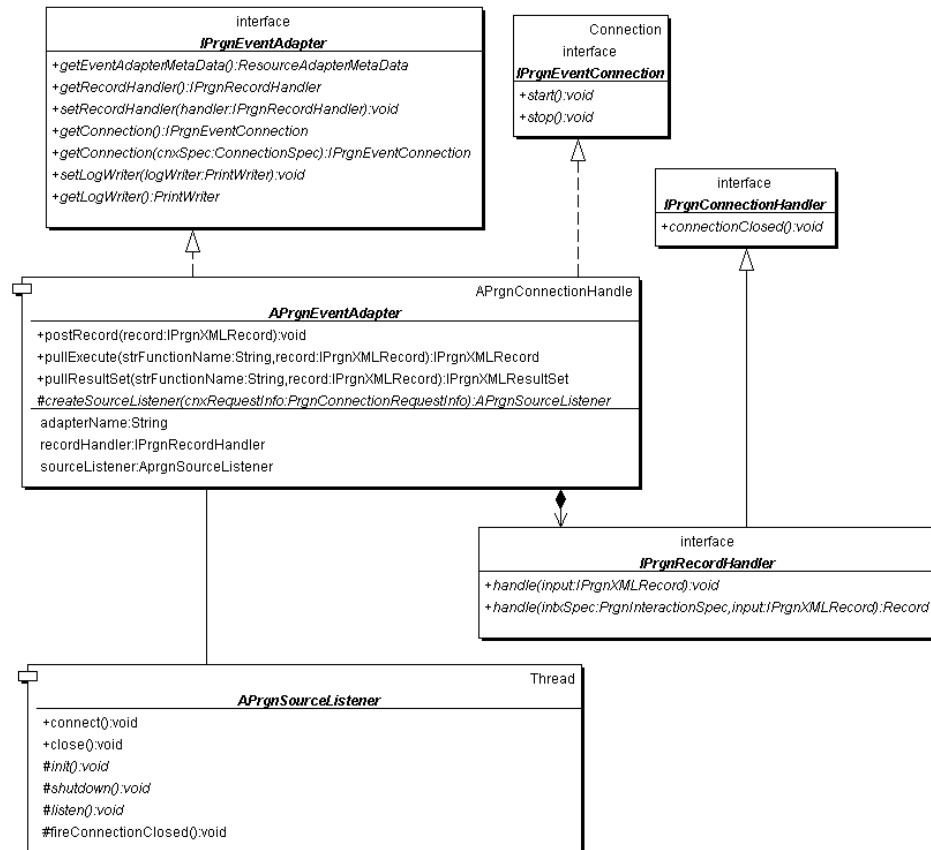
実装

APrgnManagedConnectionは、接続ハンドルの集合を維持しつつ接続の共有をサポートします。

JCAの拡張：イベントクラス

JCA規格は非同期の作動をサポートしません。このイベントモードを使用するためには、EISがコールバックの相互作用と通知を管理できなければなりません。この種の相互作用は、以下の表にあるクラスを使って生成されます。

図 11.3. Java開発キット - イベントクラス



com.peregrine.resourceimpl.event.IPrgnEventAdapter

```

//Event adapter metadata
javax.resource.cci.ResourceAdapterMetadata getEventAdapterMetadata()

//EIS event listener registration
IPrgnRecordHandler getRecordHandler()
void setRecordHandler(IPrgnRecordHandler handler)

//connection methods
IPrgnEventConnection getConnection() throws javax.resource.ResourceException
IPrgnEventConnection getConnection(javax.resource.cci.ConnectionSpec cnxSpec) thr
    
```

```

ows javax.resource.ResourceException

//logging methods
void setLogWriter(PrintWriter logWriter) throws ResourceException
PrintWriter getLogWriter() throws ResourceException

```

com.peregrine.resourceimpl.event.IPrgnEventConnection

```

//event treatment methods
void start() throws javax.resource.ResourceException
void stop() throws javax.resource.ResourceException

```

com.peregrine.resourceimpl.event.IPrgnRecordHandler

```

//Connection notification
void connectionClosed();

```

UDC (Unified Document Content) 規格

Connect-Itでは、コネクタの生成用または取り込み用ドキュメントタイプに合致したXMLドキュメントの形で、データが伝達されます。

ドキュメントタイプの構造は、Connect-It専用のUDC (Unified Document Content) 規格に準拠します。

UDC規格はドキュメントタイプ内で3つの要素 (またはノード) を定義します。

- <STRUCTURE>
この要素は、コネクタの使用可能なドキュメントタイプ内の構造体に相当します。
- <COLLECTION>
この要素は、コネクタの使用可能なドキュメントタイプ内のコレクションに相当します。
- <ATTRIBUTE>
この要素は、コネクタの使用可能なドキュメントタイプ内の属性に相当します。

要素	サポートされる 従属要素	コンテンツの混 合をサポート	データコンテン ツをサポート	コンテンツの基 数
<COLLECTION>	<ul style="list-style-type: none"> • <COLLECTION> • <STRUCTURE> • <ATTRIBUTE> 	<ul style="list-style-type: none"> • いいえ 	<ul style="list-style-type: none"> • いいえ 	Unbounded
<STRUCTURE>	<ul style="list-style-type: none"> • <COLLECTION> • <STRUCTURE> • <ATTRIBUTE> 	<ul style="list-style-type: none"> • はい 	<ul style="list-style-type: none"> • いいえ 	Bounded
<ATTRIBUTE>	<ul style="list-style-type: none"> • なし 	<ul style="list-style-type: none"> • いいえ 	<ul style="list-style-type: none"> • はい 	N/A

<ATTRIBUTE>要素にサポートされているデータ型は以下の通りです。

- Boolean
- Char
- Short
- Integer
- Long
- Float
- Double
- String
- Date
- Time
- Timestamp
- Blob
- Memo

Javaコネクタの使用可能なドキュメントタイプは、この規格に従っています。使用可能な各ドキュメントタイプのルートノードは、<STRUCTURE>要素に相当し、<STRUCTURE>要素の名前は、コネクタの接続先アプリケーションの1データ集合に当たります。例：このデータ集合は、データベース型コネクタではテーブルの名前になります。

JDBC Javaコネクタの使用可能なドキュメントタイプの例

```
<STRUCTURE name="myTable">
  <ATTRIBUTE name="field1" type="String"/>
  <ATTRIBUTE name="field2" type="Long"/>
  <COLLECTION name="link">
    <STRUCTURE name="myLinkedTable">
      <ATTRIBUTE name="field3" type="String"/>
      <ATTRIBUTE name="field4" type="Memo"/>
    </STRUCTURE>
  </COLLECTION>
</STRUCTURE>
```

```
</COLLECTION>
</STRUCTURE>
```

1ドキュメントの全ノードの名前は固有でなければなりません。例外は以下の場合です。

- ノードの複数フィールド (<ATTRIBUTE>要素) が同じデータを含む場合
- ノードが統括する<COLLECTION>と<STRUCTURE>要素が同じである場合

UDC規格の実装

Connect-ItのJava開発キットには、XMLとUDCドキュメントの編集用に多数のクラスが付属しています。

クラスは、パッケージcom.peregrine.conit.documentにまとめられています。

- PrgnDocument
XML文書の一般的な定義を統括します。
- APrgnElement
UDCラップの親クラス
- PrgnAttribute
<ATTRIBUTE>ノードのラップ
- APrgnElementContainer
<STRUCTURE>と<COLLECTION>ノードの親クラス
- PrgnStructure
<STRUCTURE>ノードのラップ
- PrgnCollection
<COLLECTION>ノードのラップ

ドキュメントタイプはPrgnStructureラップと共に実装されなければなりません。

これらのクラスの詳細については、「[Connect-Itインストール先フォルダ]doc/javadoc/conit/index.html」ファイルをクリックし、HTML文書Javadocを参照してください。

Javaコネクタの作成

本節では、Connect-ItでのJavaコネクタの作成方法を説明します。

前提条件

Javaコネクタの作成方法は、外部アプリケーション（EIS）の作動モードに応じて変化します。

外部アプリケーション（EIS）には、大きく分けて以下の2つの作動モードがあります。

- 要求 / 応答モード（例：リレーショナルデータベース）
- イベント / リッスン（例：メッセージシステム）

本節では、要求 / 応答モードを使用するJavaコネクタの作成について説明します。イベントコネクタの説明については、本章の「[イベントコネクタを作成する](#)」の節を参照してください。

接続の契約を定義する

接続の契約を定義する場合、以下の内容を定義します。

- 外部アプリケーション（EIS）の接続タイプ
- この接続を確立するためクライアントコンピュータに必要なパラメータ
これらのパラメータはXML導入ファイル（例：ra.xml）の<config-property>ノードに対応します。

接続タイプ

物理的な接続は、`com.peregrine.resource.impl.spi.APrgnManagedConnection`派生クラスに含まれています。

以下のメソッドを書く必要があります。

- `cleanupPhysicalConnection(): void`
このメソッドは以下の内容を実装しなければなりません。
 - プールされた接続が送信された時点で実行するアクション
 - 次回使用時用の接続の準備
- `destroyPhysicalConnection(): void`
このメソッドは、外部アプリケーション（EIS）への接続の切断を実装しなければなりません。

また、物理的な接続を経由して外部アプリケーション（EIS）のメタデータへアクセスするためのコードを、以下の方法で作成する必要もあります。

- `com.peregrine.resourceimpl.spi.APrgnManagedConnectionMetaData`を継承するクラスを書く

- このクラスのインスタンスを返すためにgetMetaData() : ManagedConnectionMetaDataメソッドをオーバーロードする

接続作成のパラメータ

APrgnManagedConnectionFactoryクラスの役割は、外部アプリケーション（EIS）へ物理的な接続を作成することです。

導入ファイル内で定義される接続の全プロパティでは、物理的な接続を作成するために、データのget（取得）やset（初期化）のパブリックメソッドを作成する必要もあります（例：データベース名、サーバ名など）。

コネクタのメタデータをエクスポートするためには、getMetaData() : ResourceAdapterMetaDataメソッドをオーバーロードする必要もあります。これによって、サポートされる相互作用のタイプを知ることができます。このメソッドは、com.peregrine.resourceimpl.cci.PrgnResourceAdapterMetaDataのインスタンスを戻さなければなりません。

クライアントの接続を取得するために必要なパラメータ

これらのJCAクラスを使用するコードは、PrgnConnectionRequestInfoオブジェクトを用いて接続を要求します。これは「ユーザ名」と「パスワード」の接続パラメータを使用するコンポーネントです。コードのこの場所に、ユーザ接続の取得用に必要な追加パラメータを置くことができます。

トランザクションの契約を定義する

トランザクションの契約を定義するという事は、外部アプリケーション（EIS）がトランザクションをサポートするということを意味します。

注意:

Connect-Itはローカルトランザクションのみをサポートします。

トランザクションの契約を定義するには、com.peregrine.resourceimpl.spi.APrgnManagedConnectionFactoryクラスの、getLocalTransaction() : LocalTransactionメソッドを使用する必要があります。このメソッドがサポートされていない場合は、例外javax.resource.NotSupportedExceptionを起動します。サポートされている場合は、APrgnLocalTransactionの派生クラスを書いて、以下のメソッドを使用します。

- begin() : void
新規のトランザクションを開始します。
- commit() : void

- トランザクションをコミットします。
- rollback() : void
トランザクションをロールバックします。
ステータス通知のメソッドを使用することも可能です。
- fireLocalTransactionStarted() : void
管理されリンクされた接続に、ローカルトランザクションが始動したことを通知します。
- fireLocalTransactionCommitted() : void
管理されリンクされた接続に、ローカルトランザクションがコミットされたことを通知します。
- fireLocalTransactionRolledBack() : void
管理されリンクされた接続に、ローカルトランザクションがロールバックされたことを通知します。

セキュリティの契約を定義する

セキュリティの契約を定義すると、外部アプリケーション (EIS) へのアクセスをセキュリティで保護できるようになります。

ベースクラス `APrgnManagedConnectionFactory` は、セキュリティで保護されたコンテキスト内で物理的な接続を作成します。

デフォルトの基本実装は、認証機構 `BasicPassword` です。多くの場合この実装で十分です。

認証用に使用されるセキュリティ情報は、`javax.resource.spi.security.PasswordCredential` オブジェクトです (ユーザ名と、パスワード)

`getPasswordCredential(Subject, ConnectionRequestInfo)` メソッドは、Factory用に `javax.security.auth.Subject` オブジェクトからセキュリティ情報を抽出します。サブジェクトがない場合 (Connect-Itはこの場合に当たります)、メソッドは作成された `PasswordCredential` を経由して、`com.peregrine.resourceimpl.spi.PrgnConnectionRequestInfo` オブジェクト内に含まれた情報を直接使用します。

以下のメソッドを作成する必要があります。

`createManagedConnection(PasswordCredential) : APrgnManagedConnection`

別の認証機構が必要な場合、1番高いレベルのメソッドをオーバーロードする必要があります。

`createManagedConnection(Subject, ConnectionRequestInfo) : ManagedConnection`

派生クラス `APrgnManagedConnection` 内に、次のメソッドをオーバーロードします。

`newConnectionHandle(Subject, ConnectionRequestInfo) : APrgnConnectionHandle`
 新規接続への参照を作成するには、

- クライアントが、外部アプリケーション（EIS）の物理的インスタンス上での再認証をサポートしない場合は、セキュリティ情報を無視しなければなりません。サポートする場合は、`APrgnManagedConnectionFactory`と同じセキュリティ機構を使用します。
- 再認証がサポートされている場合、コネクタは`APrgnManagedConnection`インスタンスのセキュリティコンテキストと、配置された`Subject`インスタンスのコンテキストを変更します。

データを定義する

Connect-Itで、コネクタはXML文書（ドキュメント）の形でデータを交換します。これらのドキュメントの内容は、Unified Document Content（UDC）規格に基づいています。各ドキュメントは1ドキュメントタイプに基づいており、ドキュメントタイプは更にUDC規格に準拠しています。ドキュメントは、1構造体（<STRUCTURE>ノード）で表現され、構造体はフィールド（<ATTRIBUTE>ノード）、構造体（<STRUCTURE>ノード）とコレクション（<COLLECTION>ノード）を含みます。

UDC規格の詳細については、本章の「UDC（Unified Document Content）規格」の節を参照してください。

データへの相互作用を定義する

データへの相互作用を定義すると、JCA規格のクライアント部分が実行する演算を定義できるようになります。最も重要なのは、外部アプリケーション（EIS）のデータにアクセスするメソッドに関連します。

Connect-Itのコネクタの相互作用は、XMLの相互作用です。ベースコンポーネントは、`com.peregrine.resourceimpl.APrgnXMLInteraction`クラスと共に、相互作用をサポートします。XML相互作用のメソッドの結果（1つまたは複数）は、以下の2つのタイプになります。

- XMLRecord
- XMLResultset

コネクタのベースコンポーネントは、設定、接続、自動記述などのために相互作用を定義します。これらの演算は全コネクタに共通しており、変更されたりオーバーロードされたりしてはなりません。

このため、コネクタ独自の相互作用には同じ名前を使用することができません。使用してはならない名前は以下の通りです。

- connect
- disconnect
- start
- stop
- setConfig
- getConfig
- getOperations
- describe
- getRecords
- viewRecords
- getRequests
- getRequest
- setEventFilter
- startTransaction
- endTransaction
- commitTransaction
- prepareTransaction
- rollbackTransaction
- forgetTransaction
- recoverTransaction
- reconc
- getDocumentChildrenName
- getDocumentsName

コネクタ独自の演算を定義するためには、`com.peregrine.conit.ra.APrgnDocumentInteraction`クラスが提供されています。これは、`APrgnXMLInteraction`の派生クラスです。このクラスの機能は、コネクタのXML演算として相互作用のメソッドをカプセル化することです。このクラスは更に、これらのメソッドの保存もサポートします。

3種類の演算があります。

- status
「status」演算は、取り込みモードにおけるコネクタの機能に相当します。この演算は「データ」(data)型のパラメータを取得し、このデータの処理ステータスを戻します。例：「失敗」または「成功」
例：データベース型のコネクタでは、「insert」は「status」タイプの演算です。
- resultSet

「resultSet」演算は、生成モードにおけるコネクタの機能に相当します。この演算は「データスキーマ」（ドキュメントタイプ）型のパラメータを取得し、データ集合を戻します。

例：データベース型コネクタで、「query」はresultSetタイプの演算です。この演算はテーブル名と、クエリの対象となるフィールドのリストをパラメータとして取得し、対応するレコードを戻します。

- document

この演算については、Connect-Itの開発チームが後日執筆するテクノートを参照してください。

具体メソッドのタイプは、以下の3つのパブリック署名に従う必要があります。

- method(PrgnDocument in) : PrgnMessage
status型の演算
- method(PrgnDocument in) : IPrgnXMLResultSet
resultSet型の演算
- method(PrgnDocument in) : PrgnDocument
document型の演算

コネクタの自動記述

クライアント部分が以下の情報を取得できるようにするため、コネクタは自動記述しなければなりません。

- コネクタの生成用または取り込み用ドキュメントタイプ
- 生成用または取り込み用のドキュメントタイプにコネクタが実行できる演算
- コネクタが適用できるルール（ディレクティブ）
- コネクタが使用するスケジュールのポインタ

コネクタの自動記述は、com.peregrine.conit.document.IPrgnDescribableインタフェースを介して実行されます。このインタフェースは、com.peregrine.conit.ra.APrgnDocumentInteractionクラスに実装されており、以下のメソッドを定義します。

- getSubFormats(PrgnPath path, String strMode) : PrgnDocument
- getOperations(): Vector
- getDirectives(): Vector
- getPointers(): Vector

これらのメソッドを作成すると、コネクタは自動記述できるようになります。APrgnDocumentInteractionに実装されるデフォルトの動作は、ドキュメントタイプも演算もないコネクタです。

Connect-Itの全コネクタに共通のdescribe演算は、この自動記述を検索できるようにします。

使用可能なドキュメントタイプ - Javaコネクタの自動記述

Javaコネクタの使用可能なドキュメントタイプを定義するには、`com.peregrine.conit.ra.APrgnDocumentInteraction`クラスの、`getSubFormat(PrgnPath path, String strMode) : PrgnDocument`メソッドをオーバーロードする必要があります。

`strMode`パラメータは、ドキュメントタイプのどの記述が必要であるかを指定します。このパラメータは以下の値をとります。

- `resultSet`
または、生成用ドキュメントタイプに対応する
`com.peregrine.conit.core.IPrgnOperation.RESULTSET`
- `status`
または、取り込み用ドキュメントタイプに対応する
`com.peregrine.conit.core.IPrgnOperation.STATUS`
- `document`
または、`com.peregrine.conit.core.IPrgnOperation.DOCUMENT`

注意:

このパラメータについては、Connect-Itの開発チームが後日執筆するテクノートを参照してください。

`getSubFormat`メソッドは、原則として外部アプリケーションのデータの単層記述（非階層型な記述）を戻します。完全な記述は、継続呼び出しによって取得されます。

パラメータ「`path`」が空の場合、メソッドは取り込み用または生成用で使用可能なドキュメントタイプのリストを戻さなければなりません。データベース型のコネクタでは、空の「`path`」パラメータはこの種のリストを戻すことがあります。

```
<schema>
<STRUCTURE name="docType1"/>
<STRUCTURE name="docType2"/>
<STRUCTURE name="docType3"/>
<STRUCTURE name="docType4"/>
(...)
</schema>
```

 **注意:**

返されるXMLドキュメントのルートノードに特に意味はありません。また、有効なXML要素であれば全てルートノードになり得ます。

「path」パラメータが空でない場合、メソッドは、このパラメータ値が指定するドキュメントタイプの1子ノードの内容を戻さなければなりません。パスが無効であると例外が発生する可能性があります。

例1

pathパラメータ値として「docType2」を使用する

```
<schema>
  <ATTRIBUTE name="a1" type="Long"/>
  <ATTRIBUTE name="a2" type="Byte"/>
  <STRUCTURE name="s1"/>
  <COLLECTION name="c2"/>
  <STRUCTURE name="s2"/>
  (...)
</schema>
```

例2

pathパラメータ値として「docType2.s1」を使用する。この場合、ドキュメントタイプ「docType2」の子ノード「s1」が、取得されるドキュメントです。

```
<schema>
  <ATTRIBUTE name="a5" type="String"/>
  <COLLECTION name="c1"/>
  (...)
</schema>
```

 **注意:**

- 生成用または取り込み用にコネクタがドキュメントタイプをエクスポートしない場合、NULLが戻されることもあります。
- PrgnPathクラスは、パスの要素上で処理を繰り返すパブリックサブクラス PrgnPathTokenizerを含みます。

演算 - Javaコネクタの自動記述

演算は以下の要素で定義されます。

- 名前
- 演算タイプ
- 演算を使用できるドキュメントタイプのリスト

コネクタは、取り込みまたは生成モードで複数の演算をエクスポートできます。

Javaコネクタの演算をエクスポートするには、`com.peregrine.conit.ra.APrgnDocumentInteraction`クラスの、`getOperations() : Vector`メソッドをオーバーロードする必要があります。

このメソッドは、`com.peregrine.conit.document.description.PrgnOperationDesc`のベクタを戻さなければなりません。戻されるベクタが空であると、コネクタは演算を使用できません。**注意**：イベントコネクタの`getRecords`演算の場合は、この限りではありません。

`PrgnOperationDesc`クラスコンストラクタでは以下のパラメータが必要です。

- 1演算名
- 1演算タイプ

このパラメータは文字列であり、`status`演算 (`com.peregrine.conit.core.IPrgnOperation.STATUS`クラス)、または`resultSet`演算 (`com.peregrine.conit.core.IPrgnOperation.RESULTSET`クラス) になり得ます。

`PrgnOperationDesc.addParam(String strParam) : void`メソッドでは、演算にパラメータを追加できます。`strParam`パラメータは、この演算の適用先となるドキュメントタイプの名前に当たります。

1つの演算は複数のドキュメントタイプに適用されることもあります。デフォルトで、1演算は全ドキュメントタイプに適用されます。

注意:

`status`と`resultSet`型の演算は、`status`と`resultSet`型のドキュメントタイプにそれぞれ適用されます。

シナリオビルダで、コネクタは演算のリストをコネクタのルール内に表示します。

演算のルール (ディレクティブ) - Javaコネクタの自動記述

コネクタのルール (ディレクティブ) は、コネクタがサポートする演算を分類します。

例：データベース型のコネクタでは、ソースデータベースにフィルタを適用するWHERE句とORDERBY句は、取り込み用ルールに当たります。

シナリオビルダでコネクタのルールを入力する方法については「コネクタのルール (ディレクティブ)」の章を参照してください。

ルールとルールグループ

ルールは機能ごとにまとめられています。例えば、整合性チェック用ルールのグループなどです。

ルールの定義

ルール（ディレクティブ）は以下の要素で定義されます。

- 名前
- タイプ
- デフォルト値（オプション）
- ルールの適用先となるノードタイプのリスト
- シナリオビルダ内でルールのタイトルを構成する記述（オプション）

ルールは、以下のXMLドキュメントのノードに適用されることができます。

- <STRUCTURE>
この要素は、コネクタの使用可能なドキュメントタイプ内の構造体に相当します。
ドキュメントタイプのルートノード（root）は、別の方法で処理される<STRUCTURE>です。
- <COLLECTION>
この要素は、コネクタの使用可能なドキュメントタイプ内のコレクションに相当します。
- <ATTRIBUTE>
この要素は、コネクタの使用可能なドキュメントタイプ内の属性に相当します。

ルールグループの定義

ルールグループは以下の要素により定義されます。

- 名前
- ルールのリスト
- ルールグループを使用する演算のリスト
- シナリオビルダ内でルールグループのタイトルを構成する記述（オプション）

データベース型のコネクタが使用するルールグループ「FILTER」と「RECONCILIATION」の例は、以下の表の通りです。

グループ名	演算	ルール (2)	タイプ	デフォルト 値	関連する導入 ファイルの ノード
FILTER	query	WHERE	Memo	なし(空の メモ)	ルートノード に適用される
		ORDERBY	文字列型	なし(空の 文字列)	(データベー スのテーブ ル)

グループ 名	演算	ルール (1)	タイプ	デフォルト 値	関連する導入 ファイルの ノード
RECONCATION	insert	RECONCKEY	ブール型	false	ATTRIBUTE ノードに適用 される (データベー スのフィール ド)

ルールの値の処理

演算が呼び出されると、ルールの値は、パラメータとして渡されたXMLドキュメントの<layer>要素内に渡されます。

例

```
<operation>
  <STRUCTURE name="format">
    <ATTRIBUTE name="a1" type="String"/>
    <ATTRIBUTE name="a2" type="Long"/>
    <STRUCTURE name="s1">
      <ATTRIBUTE name="sa1" type="Boolean"/>
    </STRUCTURE>
  </STRUCTURE>
  <layer>
    <STRUCTURE name="format">
      <dir1>value</dir1>
      <dir2>value</dir2>
      <ATTRIBUTE name="a1" type="String">
        <dir3>value</dir3>
      </ATTRIBUTE>
      <STRUCTURE name="s1">
        <dir2>value</dir2>
      </STRUCTURE>
    </STRUCTURE>
  </layer>
</operation>
```

```

<ATTRIBUTE name="sa1" type="Boolean">
  <dir3>value</dir3>
</ATTRIBUTE>
</STRUCTURE>
</STRUCTURE>
</layer>
</operation>

```

ルールのは、コネクタの使用可能なドキュメントタイプの特定のノードに関連付けられています。<layer>要素は、コネクタの使用可能なドキュメントタイプの構造を複製します。この際、これらのフィールドの値 (<ATTRIBUTE>要素) や、ルールが全く適用されていないノード、またはルールの値がルールのデフォルト値と同一であるノードは、複製されません。

com.peregrine.conit.ra.APrgnDocumentInteractionクラス内の、
getDirectiveValue(PrgnDocument doc, PrgnPath path, String strDirectiveName, int iDirectiveType) : Objectメソッドは、ルールの値を取得します。

docパラメータは、演算のパラメータとして渡されたXMLドキュメントを含まなければならない。

pathパラメータは、どのノードでルールの値を取得するかを指定します。このパスに、ドキュメントタイプのルートノードが含まれてはなりません。

pathDirectiveTypeh パラメータは、ルールの基本タイプを指定します。拡張タイプのルールでは、PrgnTypes.STRINGタイプを渡します。

戻されるオブジェクトクラスは、ルールの基本タイプにより変化します。ルールの値が見つからない場合は、NULL値が戻されます。

以下のコードの例では、「Format.s1.sa1」ノードの「dir3」値を検索します。dir3のタイプは「PrgnTypes.INT」であると仮定します。

例

```

public PrgnDocument operation(PrgnDocument doc) throws ResourceException
{
  (...)
  PrgnPath path = new PrgnPath();
  path.addRight("s1").addRight("sa1");
  int iVal = ((Integer)getDirectiveValue(doc, path, "dir3", PrgnTypes.INT)).intValue();
  (...)
}

```

ルールグループの表示

ルールグループをエクスポートするには、
com.peregrine.conit.ra.APrgnDocumentInteractionクラスの、getDirectiveGroups() : Vectorメソッドをオーバーロードする必要があります。

このメソッドは、com.peregrine.conit.document.description.PrgnDirectiveGroupDescのVectorを戻します。戻されたVectorが空である場合、コネクタで使用可能なルールはありません。

PrgnDirectiveGroupDescクラスコンストラクタには、パラメータが1つだけあります（ルールグループの名前）。


ルールグループを使用する演算を定義するには、以下のメソッドを使用します。

- addOperation(PrgnOperationDesc desc) : void
- addOperation(String strOperationName, String strOperationType) : void

グループにルールを追加するには、addDirectiveDesc(PrgnDirectiveDesc desc) : voidメソッドを使用します。

com.peregrine.conit.document.description.PrgnDirectiveDescクラスコンストラクタには2つのパラメータがあります。

- ルールの名前
- ルールのタイプ

ルールのタイプは、このルール用に予期されている値のタイプを決定します。例：ブール型ルールで、使用可能な値は「true」または「false」です。このタイプは、ルール編集のグラフィカルコントロールも指定します。例：「**ファイル名**」タイプのルールにアイコン  付きのフィールドを使用する。

可能なルール（ディレクティブ）のタイプは、com.peregrine.conit.document.PrgnTypesクラス内で定義されています。

これらのタイプに、com.peregrine.conit.document.description.PrgnDirectiveDescクラス内で定義される拡張タイプを追加する必要があります。

- ファイル名
TYPE_FILE_NAME
- フォルダ名
TYPE_DIR_NAME
- 変更不可能なリストデータ
TYPE_ENUM
- 変更可能なリストデータ
TYPE_FREE_ENUM

注意:

全ての拡張タイプの基本タイプは、PrgnTypes.STRINGです。

setDefaultVal(Object) : voidメソッドは、ルールのデフォルト値を定義できるようにします。パラメータとして渡されるオブジェクトクラスは、ルールのタイプに対応しなければなりません。またルールの拡張タイプの場合は、オブジェク

トクラスが基本タイプに対応しなければなりません。このデフォルト値は必須ではありません。

ルールの適用先のノードのタイプを定義するには、以下のメソッドを使用します。

- setApplyOnRoot(boolean bApply) : void
- setApplyOnStructures(boolean bApply) : void
- setApplyOnCollections(boolean bApply) : void
- setApplyOnAttributes(boolean bApply) : void

デフォルトでは、ルールはどのノードにも適用されません。

addPossibleValue(Object) : voidとsetPossibleValues(Collection) : voidメソッドは、以下の内容を定義します。

- ルールのタイプがPrgnDirectiveDesc.TYPE_ENUM (変更不可能なリストデータ) の場合は、ルールの使用可能な値
- ルールのタイプがPrgnDirectiveDesc.TYPE_FREE_ENUM (変更可能なリストデータ) の場合は、ルールに既に設定された値

スケジュールのポインタ - Javaコネクタの自動記述

シナリオがスケジュールモードで起動している場合、スケジュールのポインタとはソースコネクタが使用する一定の値を指します。ポインタは、最終セッションと同一のデータが処理されるのを回避するために使用されます。

スケジュールのポインタの概念は、resultSetタイプの演算でのみ有用です。

データベース型コネクタでは、スケジュールのポインタは、前回のセッション以降に変更または作成されたレコードのみを処理するために使用されます。この場合スケジュールのポインタは、一般的にレコードの最終変更の日付を記すフィールドになります。

キュー内に格納されるデータを処理するコネクタ (MQSeriesコネクタ) の場合、スケジュールのポインタは、最終セッション後に作成されたデータ集合のみを処理するために使用されます。データ集合は複数のメッセージ、文書、XMLファイルなどに相当します。この場合、スケジュールのポインタはデータ集合の番号に当たります。例：第1回目のセッションで、MQSeriesコネクタが番号0001から0999までのメッセージを処理したとします。第2回目のセッションでは、0999より大きな番号のメッセージのみが処理されます。

スケジュールのポインタの定義

ポインタは、以下の要素により定義されます。

- ポインタのデータ型
- デフォルト値

この値は、スケジュールモードでシナリオの最初のセッションが起動する時の値です。

- ドキュメントタイプのリスト

このリストは、ドキュメントタイプをまとめており、ドキュメントタイプの生成はこのスケジュールのポインタに左右されます。

ポインタ値の処理

resultSetタイプの演算がスケジュールモードで呼び出されると、ポインタ値はpointerルールの値として渡されます。

com.peregrine.conit.ra.APrgnDocumentInteractionクラスで、getPointerValue(PrgnDocument doc) : Objectメソッドはポインタ値を取得します。docパラメータは、演算のパラメータとして渡されたXMLドキュメントを含まなければなりません。

戻されるオブジェクトクラスは、ポインタのタイプにより変化します。以下の2つの場合はNull値が戻されます。

- スケジュールのポインタが定義されていない。
- シナリオがスケジュールモードで起動していない。

新規ポインタ値は、ルールのレポートを介してConnect-Itへ伝達されます。ルールは、getQueryReport() : IPrgnXMLRecordメソッドにより作成されます。レポートはXMLドキュメントの形を取ります。

新規ポインタ値は、以下の2つの属性を含む<pointer>ノード（<pointers>ノード内に含まれる）に渡されます。

- 'type'
ポインタのタイプ
- 'name'
現在のドキュメントタイプ名

クエリのレポートの例

```
<report>
  (...)
  <pointers>
    <pointer name="format" type="String">new pointer value</pointer>
    (...)
  </pointers>
  (...)
</report>
```

getQueryReportの実装例

```

public IPrgnXMLRecord getQueryReport() throws ResourceException
{
    PrgnDocument report = new PrgnDocument("report");
    (...)
    PrgnDocument pointers = report.createChildDocument(IPrgnTags.POINTERS);
    PrgnDocument pointer = pointers.createChildDocument(IPrgnTags.POINTER);
    pointer.setAttribute(IPrgnTags.ATTR_NAME, "tableName");
    pointer.setAttribute(IPrgnTags.ATTR_TYPE, PrgnTypes.getTypeName(PrgnTypes.LONG));
    PrgnXMLUtil.addTextNode(pointer.getElement(), m_pointerValue.toString());
    (...)
    return report;
}

```

注意:

新規ポインタ値がレポートに渡される場合に、シナリオがスケジュールモードで起動していないと、この値は無視されます。

スケジュールのポインタ値は、Connect-Itログ内に保存されます。

スケジュールのポインタのステータス（値）を確認するには、以下の手順に従います。

- 1 シナリオビルダを起動します。
- 2 生成モードで、シナリオまたはコネクタを開きます。
- 3 **[シナリオ/スケジュール]** を選択します。



スケジュールのポインタのエクスポーズ

スケジュールのポインタをエクスポーズするには、com.peregrine.conit.ra.APrgnDocumentInteractionクラスの、getPointers(): Vectorメソッドをオーバーロードする必要があります。

このメソッドはcom.peregrine.conit.document.description.PrgnPointerDescクラスのVectorを戻さなければなりません。戻されたVectorが空であると、コネクタはスケジュールのポインタを使用しません。

PrgnPointerDescクラスコンストラクタには、パラメータが1つだけあります。これは「ポインタのタイプ」です。

ポインタを使用するドキュメントタイプを定義するには、以下のメソッドを使用します。

- addSchema(String strDocTypeName) : void
- addSchemas(Collection cDocTypeNames) : void

デフォルトで、スケジュールのポインタは全ドキュメントタイプに適用されません。

Java仮想マシン（JVM）を設定する

Javaコネクタ用にConnect-Itが使用するJVMは、シナリオ図で直接設定可能です。

JMVの設定方法


- 1 シナリオビルダを起動します。
- 2 **[Java / JVMを設定する]** を選択します。
- 3 **[クラスパス]** フィールドにJavaクラスパスを入力します（jarまたはzipファイル）。
複数のパスを指定する場合は、セミコロン（;）を使用します。
- 4 **[オプション]** フィールドにJVMオプションを入力します。
例：-DmyDefine=value
- 5 **[デバッグモード]** オプションをオンまたはオフにします。
このオプションにより、デバッグのメッセージ（PrgnLog.debug）と例外のスタックトレース（stacktrace）を表示できるようになります。

Javaコネクタの導入

Javaコネクタを導入するには、コネクタを動的に導入するXML導入ファイル（多くの場合ra.xmlファイル）が必要です。このファイルは、JCA規格またはこの規格のイベント拡張に基づいています。導入ファイルの例については、本章の「イベントコネクタを作成する」の節の「Javaコネクタの導入-イベントコネクタ」を参照してください。

このファイルを作成した後に、コネクタをConnect-It内に導入します。

- 1 Connect-Itを起動します。

- 2 [Java / **コネクタを導入する**] を選択します。
- 3  をクリックします。
- 4 コネクタに名前を付けます。
[**コネクタ名**] フィールドは必須で、これはツールボックス内のコネクタの名前になります。
- 5 コネクタの導入ファイルを指定します。
- 6 コネクタのタイプを選択します。
以下の選択肢があります。
 - **イベント**
JCA規格のイベント拡張
 - **リソース**
JCA規格の実装
- 7 ツールボックス内のコネクタ用のアイコンを選択します。
このアイコンは16ピクセルx16ピクセルの寸法でなければなりません。
- 8 [**識別キー**] フィールドで、取り込み用ドキュメントタイプの編集ウィンドウに識別キーを表示するコネクタのルールを指定します。
例：付属のJDBCコネクタで、このルールはRECONCKEYルールです。
Javaコネクタのルールについては、本章の「**コネクタの自動記述**」節の「**演算のルール (ディレクティブ) - Javaコネクタの自動記述**」を参照してください。
- 9 オプションを選択します。
 - **取り込みモード**
コネクタがドキュメント取り込みの演算（書き込み、送信など）をサポートする場合は、このオプションを選択します。
 - **生成モード**
コネクタがドキュメント生成の演算（読み取り、受信など）をサポートする場合は、このオプションを選択します。
 - **スケジュールモードでポインタを使用する**
コネクタがスケジュールのポインタを使用する場合は、このオプションを選択します。
Javaコネクタでのスケジュールのポインタの実装については、本章の「**コネクタの自動記述**」節の「**スケジュールのポインタ - Javaコネクタの自動記述**」を参照してください。
 - **コネクタは要求を発信する**
コネクタが別のコネクタに向けて要求を送信する場合は、このオプションを選択します。例：XMLリスニングコネクタ、SCAutoリスニングコネクタ。このオプションは、イベントコネクタのみで使用可能です。

イベントコネクタを作成する

本節では、イベントコネクタの作成方法を段階を追って説明します。

トランザクションの契約を定義する - イベントコネクタ

`com.peregrine.resourceimpl.event.IPrgnEventAdapter` インタフェースは、コネクタアーキテクチャの主要なエントリポイントです。これは、接続契約とセキュリティに関して `IPrgnEventAdapter` クラスの役割を担当し、また `getConnection()` メソッドを使用します。このため具体的な実装は、導入ファイルで定義された設定パラメータに対応する接続を作成する際に必要なパラメータ用に、`Get`と`Set`メソッドを提供する必要があります。

デフォルトで提供されている実装は

`com.peregrine.resourceimpl.event.APrgnEventAdapter` です。この抽象クラスは、**接続**インタフェースを実装し、接続の操作を可能にします。`start()`と`stop()`関数は、イベント処理のコントロールをイベント接続上に追加します。

クライアントが接続を発生させると (`getConnection()`メソッド)、外部アプリケーションの特定のイベントリスナが、`createSourceListener()`メソッドにより動的に作成されます。作成されたオブジェクト

`com.peregrine.resourceimpl.event.APrgnSourceListener`は、`java.lang.Thread`のように動作し、外部アプリケーションの以下のタスクを実行します。

- 接続の初期化
`connect()`と`init()`メソッド
- 接続の切断
`disconnect()`と`shutdown()`メソッド
- 外部アプリケーションから来る入力イベントのリッスンと、XMLレコードの生成
`listen()`メソッド
- 接続イベントの通知
`addConnectionHandler()`と`fireConnectionClosed()`メソッド

`ManagedConnectionFactory`の`getMetaData()`メソッドのように、`getEventAdapterMetaData()`メソッドはコネクタのメタデータを返します。このメソッドでは、アクティブな接続が外部アプリケーションと確立される必要はありません。

データを定義する - イベントコネクタ

イベントタイプのコネクタはUnified Document Content規格をサポートします。イベントコネクタ用のデータの定義は、他のJavaコネクタと同じです。データの定義については、本章の「Javaコネクタの作成」の節の「データを定義する」を参照してください。

データへの相互作用を定義する - イベントコネクタ

IprgnEventAdapterのもう1つの役割は、生成されるイベントと要求のターゲットオブジェクトを保存することです。このオブジェクトはcom.peregrine.resourceimpl.event.IPrgnRecordHandlerインタフェースを実装し、応答せずにXMLファイルを取り込むことができます（イベントの受信）。また、入力レコードから出力レコードを送信する再呼び出しの相互作用として、動作できます（要求）。

保存されたハンドラへのXMLイベントのルーティングは、postRecord()メソッドを経由して実行されます。要求はpullExecute()とpullResultSet()メソッドを経由して実行されます。

相互作用のカスタムメソッドの書き込みも、イベントモードでは許可されています。しかしこれらのメソッドは、要求/応答モードでの機能（クライアントの要求によるドキュメントの生産または取り込み）の一部です。

データでの相互作用の詳細については、本章の「Javaコネクタの作成」の節の「データへの相互作用を定義する」を参照してください。

自動記述 - イベントコネクタ

イベントコネクタの自動記述は、他のJavaコネクタの自動記述と同じです。コネクタの自動記述の詳細については、本章の「Javaコネクタの作成」の節の「コネクタの自動記述」を参照してください。

イベントコネクタは、イベントを生成したり要求を発信したりするために、演算をエクスポートする必要はありません。エクスポートする必要があるのは、カスタマイズされた演算のみです。

イベントコネクタが生成するイベントのフォーマットは、コネクタの生成用ドキュメント（resultSetモード）に一致します。

イベントコネクタは、カスタマイズされたstatusタイプの演算をサポートしない限り、ドキュメントタイプを取り込むことはありません。

Javaコネクタの導入 - イベントコネクタ

Javaイベントコネクタを導入するには、このコネクタを動的に導入するXML記述子 (ra.xmlファイル) が必要です。

この記述子はJCA規格に類似しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<connector>
  <display-name> My Event Adapter< /display-name>
  <description> Event Adapter< /description>
  <vendor-name> Peregrine Systems< /vendor-name>
  <eis-type> My EIS Type< /eis-type>
  <version> 1.0< /version>
  <license>
    <description> < /description>
    <license-required> true< /license-required>
  </license>
  <resourceadapter>
    <eventadapter-interface> com.peregrine.resourceimpl.event.IPrgnEventAdapter< /eventadapter-interface>
    <eventadapter-class> MyEventAdapterClass< /eventadapter-class>
    <config-property>
      <description> < /description>
      <config-property-name> MyProperty< /config-property-name>
      <config-property-type> MyPropertyType< /config-property-type>
      <config-property-value> MyPropertyValue< /config-property-value>
    </config-property>
    ...
  </resourceadapter>
</connector>
```

JCAコネクタの作成 - 例

本節では、JDBCタイプの用例コネクタの作成方法を段階を追って説明します。Java用例コネクタ用のソースは以下のフォルダにあります。

- [Connect-Itインストール先フォルダ] \datakit\javasample\com\peregrine\sample\ra\jdbc

接続の契約を定義する - 用例コネクタ

コネクタの接続パラメータは以下の通りです。

- Driver
リレーショナルDBMSにアクセスするために使用するJDBCドライバ
- URL
次のフォームを使います： <Subprotocol> : <DatabaseName>
- UserName
接続作成用に使用される最初のセキュリティ情報
- Password
接続作成用に使用される2番目のセキュリティ情報

このパラメータは、com.peregrine.resourceimpl.spi.APrgnManagedConnectionFactoryクラスを拡張できます。

```
public class JDBCManagedConnectionFactory extends APrgnManagedConnectionFactory
{
// class must be public to be deployed by an external tool.
}
```

コードのこの部分は、接続パラメータ用にデータのget（取得）とset（初期化）パブリックメソッドを取得できるようにします。

```
/**
 * Sets the JDBCManagedConnectionFactory URL property.
 */
public void setURL(String strURL)
{
?
}

/**
 * Gets the JDBCManagedConnectionFactory URL property.
 */
public String getURL()
{
?
}
```

物理的な接続を考慮に入れるためには、com.peregrine.resourceimpl.spi.APrgnManagedConnectionFactoryクラスを派生させる必要があります。

トランザクションの契約を定義する - 用例コネクタ

用例コネクタはトランザクション境界をサポートしません。

ManagedConnectionFactoryクラスのgetResourceMetaDataメソッドは、falseに初期化されているResourceMetaDataオブジェクトを、supportsLocalTransactionDemarcationプロパティと共に戻さなければなりません。

ManagedConnectionの次のメソッドは、NotSupportedExceptionを生成しなければなりません。

- getLocalTransaction() : LocalTransaction
- getXAResource() : XAResource()
- getAutoCommit() : boolean
- setAutoCommit(boolean b) : void

セキュリティの契約を定義する - 用例コネクタ

セキュリティ機構は、ユーザ名/パスワードの認証に基づいています。

ManagedConnectionFactoryセキュリティに関連しているメソッドは以下の通りです。

```
createManagedConnection(,PasswordCredential pc) : APrgnManagedConnection
matchPasswordCredential(APrgnManagedConnection managedCnx, PasswordCredential pc) : boolean
```

再認証はサポートされていません。

ManagedConnectionセキュリティに関連しているメソッドは以下の通りです。

```
newConnectionHandle(Subject subject, PrgnConnectionRequestInfo cnxRequestInfo) : APrgnConnectionHandle
```

データを定義する - 用例コネクタ

用例コネクタは、リレーショナルデータベース型の外部アプリケーションに、SELECTを実行します。

データへの操作はXML技術に基づいているため、1演算は以下の形式の1ドキュメントを取り込みます。

```
<query>
<STRUCTURE name="Employee">
  <ATTRIBUTE name="LastName">
  <ATTRIBUTE name="FirstName">
```

```
</STRUCTURE>
</query>
```

このドキュメントの結果は、SQL宣言 (SELECT LastName,FirstName FROM Employee) となります。この宣言は、データベースのレコードに操作を施す XMLResultSet を戻します。

ManagedConnectionFactoryの実装時に、以下のメソッドを追加する必要があります。

```
public ResourceAdapterMetaData getMetaData() throws javax.resource.ResourceException
{
// must return a PrgnResourceAdapterMetaData instance
PrgnResourceAdapterMetaData raMetaData = new PrgnResourceAdapterMetaData();
?
return raMetaData;
}
```

データへの相互作用を定義する - 用例コネクタ

コードはAPrgnDocumentInteractionを派生させ、次のパブリックメソッドを書きます。

```
query(PrgnDocument document) : IPrgnXMLResultSet
```

設定ウィザードを改善する

「ra.xml」ファイルは、Connect-Itシナリオビルダでコネクタを展開します。またこのファイルは、非常に単純なデフォルトの設定ウィザードを作成できるようにします。

コネクタの設定ウィザードの使用については、「コネクタの設定」の章を参照してください。

本節では、デフォルトのウィザードを改善するDSDファイルを作成し、ウィザードに以下の複雑な機能を統合する方法を説明します。

- 複数選択肢のリスト
- デフォルト値の提示
- ラジオボタン (オプションボタン)
- オペレーティングシステムのフォルダ内の検索ボタン
- パスワードの暗号化入力ゾーン
- その他

警告:

無効なDSDファイルを使用すると、コネクタの実行に失敗することがあります。

DSDファイル導入前に作成された既存のシナリオは正常に起動します。しかし、DSDファイルと共に保存されたシナリオは、そのDSDファイルが削除されると起動不可能になります。

表記法: 本節のコードの例で、赤はDSD内に常に存在する要素を表します。これらの要素は変化しません。

青は、必須の属性を表します。

DSDファイルの作成

DSDファイルは、コネクタの「ra.xml」ファイルを含むフォルダ内に挿入されなければなりません。

例: コネクタの名前が「MyCon」である場合、コネクタを導入すると、対応するフォルダがConnect-Itインストール先フォルダ内に作成されます（[Connect-Itインストール先フォル]//config/java_MyCon」）。

DSDファイルは、導入ファイルと同名でなければなりません。

例: 導入ファイルの名前が「MyCon.xml」である場合、DSDの名前は「MyCon.dsd」になります。

DSDファイルの内容

ファイルには以下の5つのノードが含まれなければなりません。

- <DSD>ノード
このノードの属性は「rootnode」で、以下の4つのノードをまとめています。
- <module>ノード
このノードは、全DSDファイルで同一です。
- 3つの<node>必須ノード
 - 第1のルートノードは、設定のプロパティの宣言を含みます。
このノードの「name」属性の値は、<DSD>ノードの「rootnode」属性の値と同じでなければなりません。
「class」属性と「edit」属性は必須であり、値はそれぞれ「Folder」と「CWBAutoDet」でなければなりません。
 - defaultノードとadvancedノードの属性は「name」であり、属性の値はそれぞれ「default」と「advanced」です。

これらの2ノードは、単純モードと高度なモードの両方でウィザードのページを定義します。これらのノードの「class」属性の値は、ルートノードの「name」属性の値でなければなりません。

例

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DSD name="Wiz" rootnode="ConWiz" version="1.0" warning="true">
  <!-- You must specify the root node -->
  <module name="core.dsd" local="true"/>
  <!-- This module node is mandatory -->
  <node name="ConWiz" class="Folder" edit="CWBAutoDet">
    (...)
  </node>
  <node name="default" class="ConWiz">
    (...)
  </node>
  <node name="advanced" class="ConWiz">
    (...)
  </node>
</DSD>
```

advancedノードがdefaultノードと相違しない場合、以下の要領で単純にすることもできます。

```
<node name="advanced" class="default"/>
```

警告:

「name」属性はタグの第1属性でなければなりません。

コネクタ設定のプロパティの定義

コネクタの設定のプロパティごとに、ルートノードには子ノードが含まれなければなりません。この子ノードは、導入ファイル (ra.xml) で使用される名前と同じ名前になります。

警告:

ルートノードで定義される全プロパティは、導入ファイル内に含まれなければなりません。

必須属性

設定の各プロパティには2つの必須属性「name」と「class」があります
サポートされるクラスは以下の通りです。

- Enum (変更不可能なリストデータ)
- FreeEnum (変更可能なリストデータ)
- Boolean (ブール)
- Long (倍長整数)
- Double (倍精度)
- Date (日/時)
- Duration (期間)
- String (文字列)
- Password (暗号化)
- FileName (ファイル名)
- DirName (フォルダ名)
- Memo (複数行のテキスト)

オプション属性

設定の各プロパティには、3つのオプション属性があります。

- 'mandatory'
ブール型
- 'inshortdesc'
このブール値がtrueである場合、パラメータの値はコネクタの要約記述内で使用されます。
- 'default'
この属性は、設定プロパティのデフォルト値を定義します。

属性「inshortdesc」または「mandatory」が指定されていない場合、デフォルト値は「false」になります。

設定のプロパティ

各設定プロパティには2つの子ノードがあります。

- <shortdesc>
このプロパティは、設定ウィザードがシナリオビルダ内で起動される時にウィザードに表示されるフィールドの説明を定義します。このプロパティは必須です。
- <longdesc>

このプロパティは、コネクタのヘルプ用テキストを定義します。

例

```
<node name="ConWiz" class="Folder" edit="CWBAutoDet">
  <node name="field1" class="String" default="test" mandatory="true" inshortdesc="true">
    <shortdesc>Field 1</shortdesc>
    <longdesc>Field 1's help</longdesc>
  </node>
  <node name="field2" class="Boolean" default="true" mandatory="false" inshortdesc="false">
    <shortdesc>Field 2</shortdesc>
    <longdesc>Field 2's help</longdesc>
  </node>
  <node name="field3" class="Long" default="0">
    <shortdesc>Field 3</shortdesc>
    <longdesc>Field 3's help</longdesc>
  </node>
  <node name="field4" class="Double" default="1.52">
    <!-- Double nodes have a precision of two digits after the decimal.-->
    <shortdesc>Field 4</shortdesc>
    <longdesc>Field 4's help</longdesc>
  </node>
  <node name="field5" class="Date" default="yyyy-mm-dd">
    <shortdesc>Field 5</shortdesc>
    <longdesc>Field 5's help</longdesc>
  </node>
  <node name="field6" class="Duration" default="1y 2mon 3d 4h 5m 6s">
    <shortdesc>Field 6</shortdesc>
    <longdesc>Field 6's help</longdesc>
  </node>
  <node name="field7" class="Password">
    <shortdesc>Field 7</shortdesc>
    <longdesc>Field 7's help</longdesc>
  </node>
  <node name="field8" class="FileName" default="c:/tmp/test.txt">
    <shortdesc>Field 8</shortdesc>
    <longdesc>Field 8's help</longdesc>
  </node>
  <node name="field9" class="DirName" default="c:/tmp">
    <shortdesc>Field 9</shortdesc>
    <longdesc>Field 9's help</longdesc>
  </node>
```

```

<node name="field10" class="Enum" default="op1">
<shortdesc>Field 10</shortdesc>
<longdesc>Field 10's help</longdesc>
<enum name="op1" value="0"><caption>Option 1</caption></enum>
<!-- The value of an enum is equal to the name attribute value-->
<enum name="op2" value="1"><caption>Option 2</caption></enum>
</node>
<node name="field10" class="FreeEnum" default="Option 1" enum="Option 1|Option
2">
<!-- Values are separated by | -->
<shortdesc>Field 10</shortdesc>
<longdesc>Field 10's help</longdesc>
</node>
<node name="field11" class="Memo">
<shortdesc>Field 11</shortdesc>
<longdesc>Field 11's help</longdesc>
<default>this is a multiline default value</default>
<!-- In the case of a Memo, the default value is set in a default node.-->
</node>
</node>

```

設定ページ

default ノードと advanced ノードは、<layout>要素を統括しています。この要素は、Java コネクタの設定ウィザードの [コネクタの導入] ページのレイアウトを定義します。

ページの内容は枠にまとめられることもあります。各枠は同種のプロパティをまとめます。例：接続プロパティ用の枠、など。

各<frame>要素は、

- 名前で定義されます。
- 枠に説明を加える<caption>要素を含みます。

各<frame>要素は、表示する設定プロパティの参照を含んでいます。

設定プロパティが枠で参照されていない場合は、プロパティの値をウィザード内でパラメータ設定することはできません。高度な設定モードのみでプロパティが表示される場合は、プロパティを advanced ノード内で参照します (default ノードではありません)。

例

```

<node name="default" class="ConWiz">
  <layout>
    <frame name="frame1">
      <caption>Frame #1</caption>

```

```
<attribute name="field1"/>
<attribute name="field2"/>
<attribute name="field5"/>
</frame>
<frame name="frame2">
  <caption>Frame #2</caption>
  <attribute name="field6"/>
</frame>
<frame name="frame3">
  <caption>Frame #3</caption>
  <attribute name="field3"/>
  <attribute name="field4"/>
</frame>
</layout>
</node>
```

DSD - 頻発する問題点

以下の問題点はDSDファイルを無効にします。

- 終了タグの欠如
- 無効な属性値
- 「name」属性がタグの第1属性として宣言されていない。
- ルートノードが指定されていない。
- 不明なクラスが使用されている。
- defaultまたはadvancedノードが不在である。または、defaultまたはadvancedノードはルートノードから継承しない。
- 不明な属性が参照されている。
- ルートノードが定義するプロパティは、導入ファイル (ra.xml) 内に存在しない。

12 | コネクタの設定


コネクタのインスタンスをシナリオ内で設定すると、以下の操作を実行できます。

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

本章では、コネクタの設定ウィザードへのアクセス方法を説明した後、複数のコネクタに共通な設定ウィザードのページについて説明します。「Peregrine Systemsコネクタ」、「アプリケーションコネクタ」、「プロトコルコネクタ」、「インベントリコネクタ」と「Javaコネクタ」の章では、コネクタの設定に関する節は、本章の説明を参照しています（例: 「Peregrine Systemsコネクタ」章の「Asset Managementコネクタ」節の「Asset Managementコネクタの設定」）。一部のコネクタの設定では、特定のオプションが共通な設定ウィザードページに含まれることがあります。これらのオプションは、各コネクタの節で説明されています。

コネクタを設定する

Connect-Itコネクタの設定はウィザードで実行できます。
コネクタの設定ウィザードを起動するには、以下の方法があります。

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

コネクタに名前を付け、役割を説明する

このページでは以下の内容を入力します。

- コネクタの名前
- コネクタの役割の説明



[名前]

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

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

警告:

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

[説明]

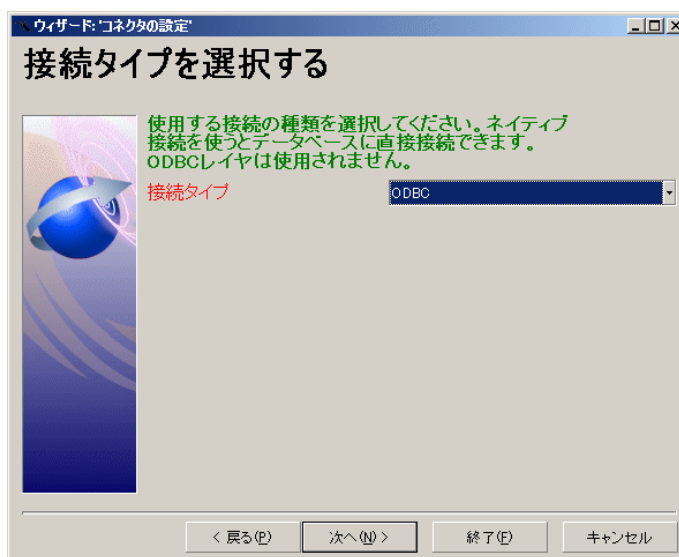
コネクタを説明するテキストを入力します（例：データソースAの読み取り、データソースAへの書き込み、など）。

これは必須フィールドではありません。

接続タイプを選択する

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

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



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などに、ネイティブ接続の方が高性能を發揮します。

注意:

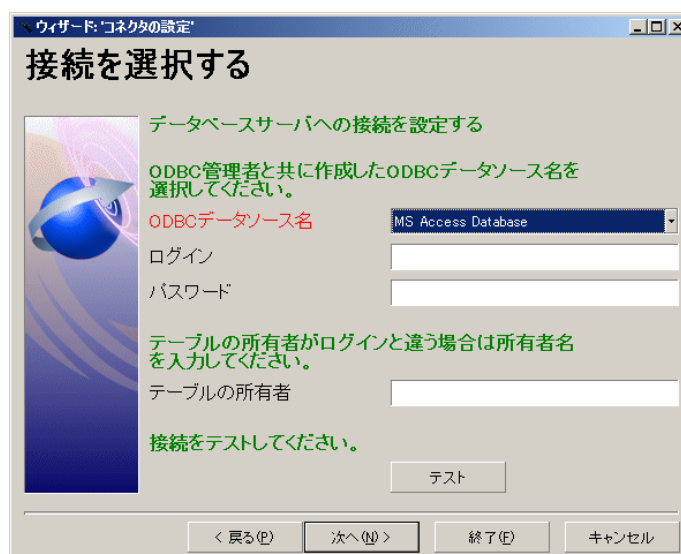
MySQLネイティブ接続を使用するには、MySQLのクライアントのダイナミックライブラリ (DLL) を、Connect-Itインストール先フォルダの「bin」フォルダ内（またはオペレーティングシステムの「system32」フォルダ内）にコピーする必要があります。

接続を選択する

このページでは接続を選択します。

前のページ [接続タイプを選択する] 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）



The screenshot shows a Windows-style dialog box titled 'ウィザード: コネクタの設定' (Wizard: Connection Settings). The main heading is '接続を選択する' (Select Connection). Below the heading, there is a blue globe icon on the left and instructional text on the right. The text reads: 'データベースサーバへの接続を設定する' (Set up connection to database server), 'ODBC管理者と共に作成したODBCデータソース名を選択してください。' (Select the ODBC data source name created with the ODBC administrator.), 'ODBCデータソース名' (ODBC data source name) with a dropdown menu showing 'MS Access Database', 'ログイン' (Login) with an input field, 'パスワード' (Password) with an input field, 'テーブルの所有者がログインと違う場合は所有者名を入力してください。' (If the table owner is different from the login, enter the owner name.), 'テーブルの所有者' (Table owner) with an input field, and '接続をテストしてください。' (Test the connection.). A 'テスト' (Test) button is located below the input fields. At the bottom of the dialog, there are four buttons: '< 戻る(B)' (Back), '次へ(N) >' (Next), '終了(F)' (Finish), and 'キャンセル' (Cancel).

このページでは以下の内容を指定します。

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

[ODBCデータソース名]

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

【ログイン】

ODBCデータソースへのアクセスを許可するログインを入力します。

【パスワード】

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

【テーブルの所有者】

ログインがテーブルの所有者と違う場合は、テーブルの所有者名を入力します。

【テスト】

このボタンを押すと接続をテストできます。

接続をテストするには、

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

接続を選択する（Oracleネイティブ接続）

ウィザード: コネクタの設定

接続を選択する

データベースサーバへの接続を設定する

データベースサーバ

ログイン

パスワード

テーブルの所有者がログインと違う場合は所有者名を入力してください。

テーブルの所有者

接続をテストしてください。

テスト

< 戻る(B) 次へ(N) > 終了(E) キャンセル

このページでは以下の内容を指定します。

- データベースサーバ
- ログイン
- パスワード
- テーブルの所有者

[データベースサーバ]

Oracleサーバの名前を指定します。

[ログイン]

データベースへのアクセスを許可するOracleのユーザ名を入力します。

[パスワード]

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

【テーブルの所有者】

ログインがテーブルの所有者と違う場合は、テーブルの所有者名を入力します。

【テスト】

このボタンを押すと接続をテストできます。

接続をテストするには、

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

接続を選択する - Sybaseネイティブ接続

ウィザード: 'コネクタの設定'

接続を選択する

データベースサーバへの接続を設定する

データベースサーバ

ログイン

パスワード

サーバデータベース

テーブルの所有者がログインと違う場合は所有者名を入力してください。

テーブルの所有者

接続をテストしてください。

テスト

< 戻る(B) 次へ(N) > 終了(F) キャンセル

このページでは以下の内容を指定します。

- データベースサーバ
- ログイン
- パスワード
- サーバデータベース

- テーブルの所有者

[データベースサーバ]

Sybaseデータベースのサーバの名前を指定します。

[ログイン]

データベースへのアクセスを許可するSybaseのユーザ名を入力します。

[パスワード]

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

[サーバデータベース]

接続先のデータベースの名前を入力します。

[テーブルの所有者]

ログインがテーブルの所有者と違う場合は、テーブルの所有者名を入力します。

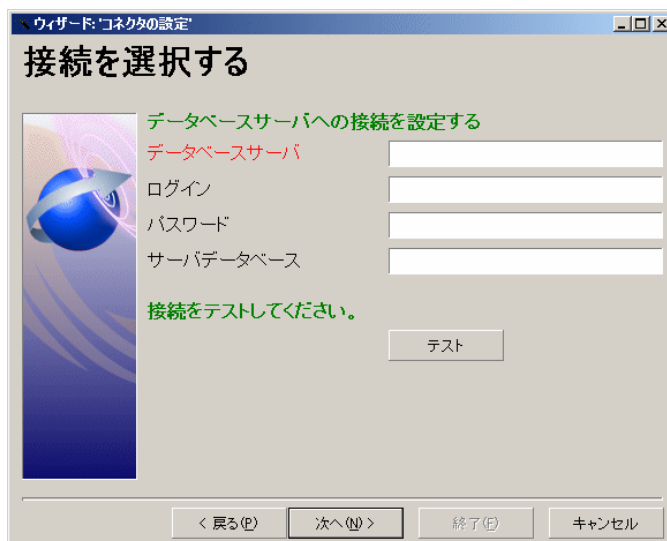
[テスト]

このボタンを押すと接続をテストできます。

接続をテストするには、

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

接続を選択する - ネイティブMySQL



ウィザード: 'コネクタの設定'

接続を選択する

データベースサーバへの接続を設定する

データベースサーバ

ログイン

パスワード

サーバデータベース

接続をテストしてください。

テスト

< 戻る(B) 次へ(N) > 終了(F) キャンセル

このページでは以下の内容を指定します。

- データベースサーバ
- ログイン
- パスワード
- サーバデータベース

[データベースサーバ]

MySQLデータベースのサーバの名前を指定します。

[ログイン]

データベースへのアクセスを許可するユーザ名を入力します。

[パスワード]

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

[サーバデータベース]

接続先のデータベースの名前を入力します。

[テスト]

このボタンを押すと接続をテストできます。

接続をテストするには、

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

高度な設定

このページでは、高度な設定のパラメータを定義できます。



データベース記述ファイル

このファイルには以下の内容が含まれています。

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

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


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

SQL92をサポート

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

詳細オプション

この編集用ゾーンには、コネクタ専用の高度なオプションを入力できます。詳細オプションを入力するには、

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

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

オプションを削除するには、オプションに対応する行を選択してから **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

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

警告:

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

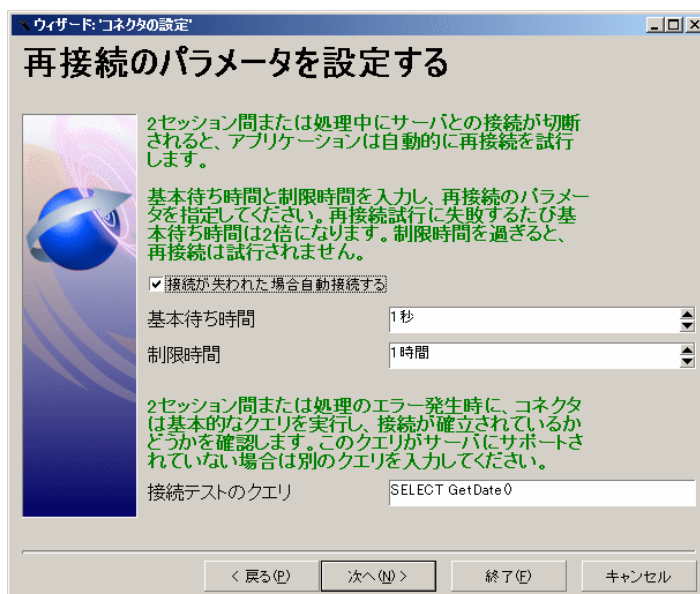
表 12.1. インベントリコネクタとデータベースコネクタ - 設定の詳細オプション

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

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

再接続のパラメータを設定する

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



接続が失われた場合自動接続する

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

注意:

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

再接続用パラメータは以下の通りです。

- **[基本待ち時間]**
[基本待ち時間] では、Connect-It が何秒後にサーバへの再接続を実行するかを指定します。再接続に失敗すると、Connect-Itは基本待ち時間を2倍にして接続を再試行します。例えば、基本待ち時間が2秒の場合、2回目の再接続は4秒後に実行され、3回目の再接続は8秒後に実行されます。
- **[制限時間]**
この [制限時間] で、Connect-Itがサーバへの再接続を何分（時間）後に中止するかを指定できます。

重要項目:

再接続に成功すると、

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

接続テストのクエリ

新規セッション時や処理エラーが発生した場合は、クエリがサーバに送信され、データベースへの接続が切断されたかどうかを確認します。クエリはサーバにサポートされていないければなりません。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。

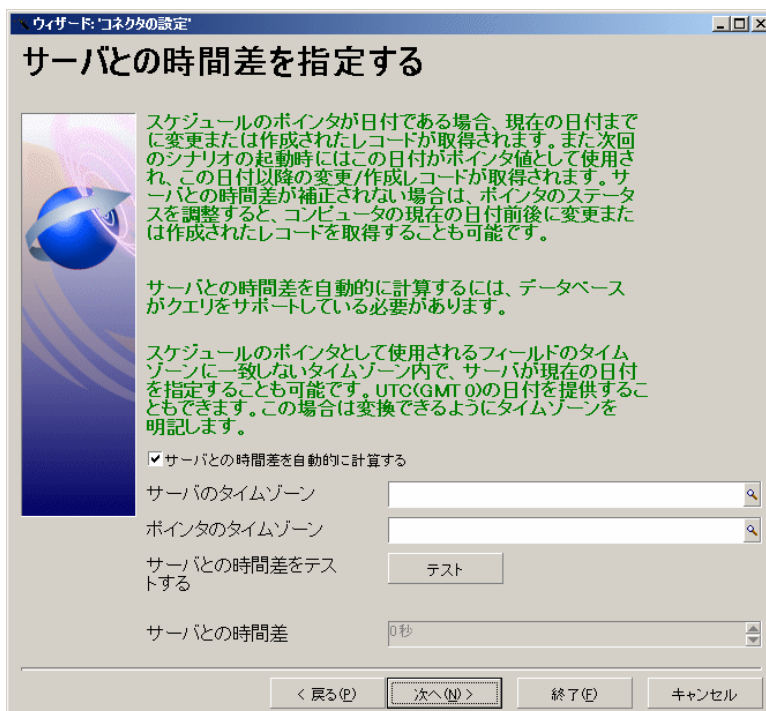
重要項目:

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。

サーバとの時間差を指定する

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



ウィザード: 'コネクタの設定'

サーバとの時間差を指定する

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

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

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

サーバとの時間差を自動的に計算する

サーバのタイムゾーン

ポイントのタイムゾーン

サーバとの時間差をテストする

サーバとの時間差

< 戻る(B) 終了(F) キャンセル

[サーバとの時間差を自動的に計算する] オプションが選択されている

このオプションはデフォルトで選択されています。

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

[サーバとの時間差を自動的に計算する] オプションが選択されていない

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

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

タイムゾーンの使用

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

例

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

タイムゾーンの指定方法

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

- 表示されるウィンドウでタイムゾーンを選択します。



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


- [OK]** をクリックします。

外部アプリケーションの現在の日付と時刻を表示する

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

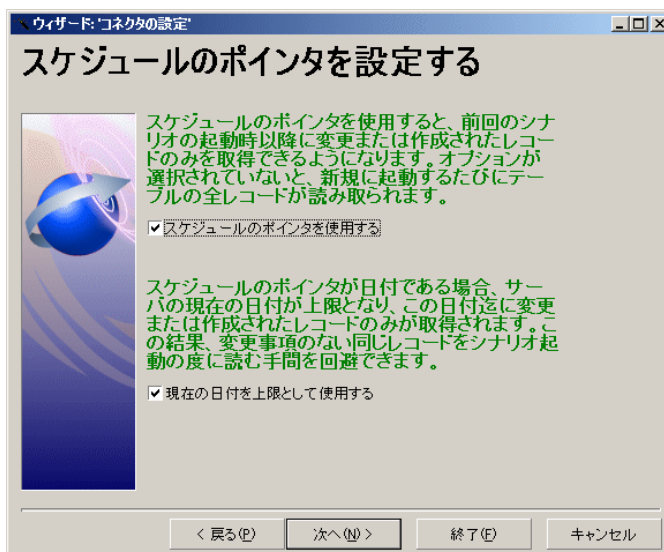
- 外部アプリケーションとの接続
- 外部アプリケーションの現在の日付と時刻が取得され、表示されます。
- タイムゾーンが指定されている場合は、タイムゾーンと時間調整をした後の現在の日付と時刻が表示されます。

タイムゾーンのパラメータをテストする

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

スケジュールのポイントを設定する

このページでは、コネクタが使用するスケジュールのポイントを設定できます。



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

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

スケジュールのポイントを使用する

このオプションはデフォルトで選択されています。

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

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

現在の日付を上限として使用する

このオプションはデフォルトで選択されています。

! 警告:

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

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

[現在の日付を上限として使用する] オプションが選択されている

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

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

[現在の日付を上限として使用する] オプションが選択されていない

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

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

トランザクションを設定する

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



コネクタは以下の操作を実行できます。

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

注意:

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

キャッシュを設定する

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



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

重要項目:

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

キャッシュファイルの詳細については、「[統合シナリオのテストとデバッグ](#)」章の「[キャッシュファイルを使用する](#)」の節を参照してください。

以下の2つの選択肢があります。

• **【キャッシュファイルを使用する】**

このオプションを使用すると、Connect-Itインストール先フォルダ内の「cache」フォルダ内にあるシナリオと同名のサブフォルダに、キャッシュファイルが作成されます。各キャッシュファイル（「.cch」）の名前は固有です。このため同一コネクタの他のインスタンスのキャッシュには、別の名前がつけます。一般的にファイルにはサーバ名が使用されます。

- **【特定のキャッシュファイルを使用する】**

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

このキャッシュファイルが存在しない場合、Connect-Itは、コネクタを最初に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。

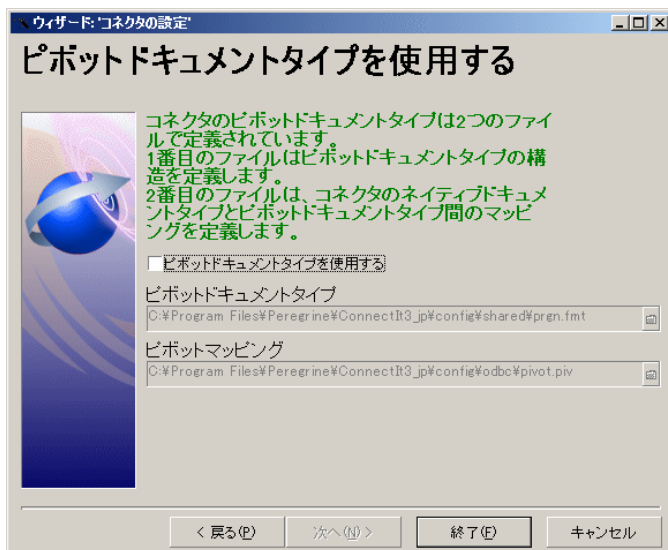
コネクタのキャッシュの同期をとるには、

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

ピボットドキュメントタイプの使用

このページでは、コネクタ用のピボットドキュメントタイプの使用を設定します。

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



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

 **注意:**

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

13 | コネクタのルール（ディレクティブ）

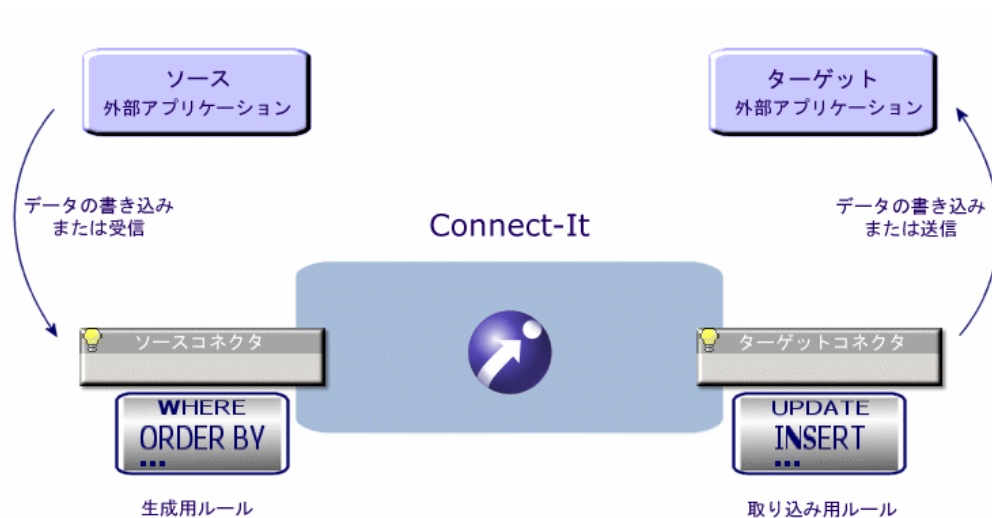
ルール（ディレクティブ）とはユーザが定義するコネクタ用の指示内容のことです。コネクタは、ドキュメントの生成または取り込み時にこの指示に従います。

ルールを作成するのは以下の場合です。

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

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

図 13.1. コネクタ - 生成用ルールと取り込み用ルール




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

生成用ルール

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

生成用ルールを定義する

コネクタの生成用ルールを定義するには、

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

作業枠が表示され、生成用ドキュメントタイプを編集作成できるようになります。

- 6 作業枠内に生成用ルールを入力します。



WHERE句とORDER BY句

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

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

WHERE句シンタックスの詳細については、「[AQLのシンタックス](#)」の章の「[WHERE句](#)」の節を参照してください。

ORDER BY句シンタックスの詳細については、「[AQLのシンタックス](#)」の章の「[ORDER BY句](#)」の節を参照してください。

WHERE句とORDER BY句のシンタックス

! 警告:

データベース型のコネクタでは、SQLシンタックスは、データベース作成に使用したデータベースエンジンに使われているシンタックスと同一でなければなりません。

SQLシンタックスに加えて、Connect-Itには以下の規則があります。

- フィールド名は角括弧で囲む

例

```
[フィールド]
```

- 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

配列の取り出しを使用しない

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

取り込み用ルール

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


コネクタの取り込み用ルールを定義する

コネクタの取り込み用ルールを定義するには、以下の2つの方法があります。

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


ドキュメントタイプ編集用ウィンドウ内で作成する取り込み用ルール

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

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

マッピングの編集用ウィンドウ内で作成する取り込み用ルール

マッピングの編集用ウィンドウでコネクタの取り込み用ルールを定義するには、

- 1 このコネクタを含むシナリオを読み込むか、または作成します。
マッピングは、シナリオ内でこのコネクタと、別のコネクタの間に作成されていなければなりません。
- 2 シナリオのコネクタを開きます。
- 3 [マッピング]タブを選択します。
- 4 をクリックするか、または既存のマッピングを編集します。
マッピングの編集用ウィンドウが開きます。
- 5 マッピングウィンドウの作業枠内に取り込み用ルールを入力します。

整合性

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

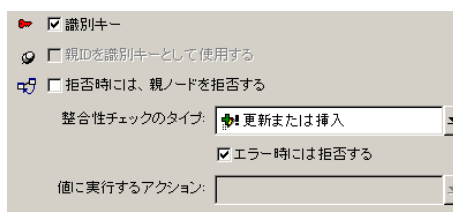
パラメータには以下の内容を入力します。

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

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

【整合性】タブ

図 13.2. 取り込み用ルール - 【整合性】タブ



☑ 識別キー

このチェックボックスでは、現在選択されている要素が、要素の所属するノード用の識別キーであるかどうかを指定できます。

フィールドまたは構造体を識別キーとして指定するには、

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

例：

Desktop Discovery - Asset ManagementシナリオのamAssetドキュメントタイプでは、**【AssetTag】**フィールドと、Product構造体の**【Brand】**と**【Model】**フィールドが識別キーとして選択されています。

構造体が識別キーとして選択されている場合

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

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

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



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

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

親IDを識別キーとして使用する

選択されている要素が、親ID（識別子）を識別キーとして使用するかどうかを指定するチェックボックス

【大文字小文字の区別】

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

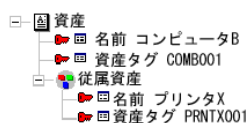
- **【大文字小文字の区別】オプションを選択する**
このオプションが選択されていると、識別キー値の大文字と小文字は区別して取り扱われます。例：従業員のEメールアドレスを含むフィールドが識別キーとして使用されている場合、'jmartin@company.com'、'JMARTIN@company.com' と 'jMartin@company.com' は、ターゲットアプリケーション内で3つの別々のレコードになります。
- **【大文字小文字の区別】オプションを選択しない**
このオプションを選択しないと、識別キー値の大文字と小文字は区別されません。例：従業員のEメールアドレスを含むフィール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

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。例えばAssetCenterでは、資産のテーブルは、従属資産のテーブル（実際は資産のテーブルと同一）に [lparentId] フィールドによりリンクされています。

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

例 :

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



例 13.1. [親IDを識別キーとして使用する] オプション

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

1 [親IDを識別キーとして使用する] を選択した場合、

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

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

1 [親IDを識別キーとして使用する] が選択されていない場合、

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

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

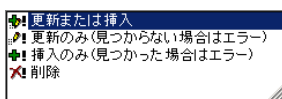
拒否時には、親ノードを拒否する

このチェックボックスでは、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ことができます。

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

例：

あるレコードが [従業員] 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 (資産) が [資産] のテーブルに作成されないよう設定できます。これにより、Asset Managementアプリケーションのデータベースに、ユーザの関連付けられていない資産が作成されることを回避できます。



【整合性チェックのタイプ】ドロップダウンリストでは、各ルートノード、構造体またはコレクションの整合性チェックのタイプを指定できます。

ルートノード、構造体、またはコレクションに整合性チェックのオプションを指定するには、

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



【値に実行するアクション】フィールドのオプションでは、整合性チェックされたレコードの値が、Asset Managementコネクタが取り込むドキュメント内のフィールド値に置き換えられるのか、または整合性チェックされたレコードの値に、Asset Managementコネクタが取り込むフィールド値が追加されるのか、を指定できます。

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

【高度な整合性チェック】タブ

【高度な整合性チェック】タブでは、以下の操作を実行できます。

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

図 13.3. 取り込み用ルール - [高度な整合性チェック] タブ

コレクションの整合性チェック

コレクションの整合性チェックを実行する

ソースドキュメントのコレクションに不在の構成要素に対して実行するアクション(デフォルトでは削除)

コレクションの整合性チェックのフィルタ

次の場合実行するアクション

複製の矛盾:

[コレクションの整合性チェック]

この枠では、ソースコネクタに生成されたドキュメント内のデータに応じて、Asset Managementデータベースのレコードを更新するよう設定できます。

コレクションの整合性チェックを実行する

このチェックボックスをオンにすると、コレクションの整合性チェックが有効になります。

ソースドキュメントのコレクションに不在の構成要素に対して実行するアクション(デフォルトでは削除)

このフィールドでは、ソースコレクションに不在の構成要素に関するアクションを設定できます。

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

例えば、ターゲットアプリケーション内のコレクションの【ステータス】フィールドに、「**削除予定**」という値を割り当てます。これにより、ターゲットアプリケーションの管理者は【ステータス】フィールドが「**削除予定**」であるレコードを検索できるようになります。

フィールドに使用するシンタックスは次の通りです。

[フィールド名] = "[選択したフィールド値]"

例

ステータス = "削除予定"

コレクションの整合性チェックのフィルタ

Name <> "WINNT"

このフィルタ（WHERE句に相当）では、あるレコードを削除しないように設定できます。（以下参照）

Connect-It内のコレクションの更新には、以下の方法があります。

- 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、Asset Managementアプリケーションのデータベースから削除する。
- Asset Management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 - Asset Managementシナリオ（「iddac.scn」ファイル）のAsset informationマッピングでは、ソースコレクションSoftwareはターゲットコレクションSoftInstallにマップされています。これらの2つのコレクションは資産（この場合はコンピュータ）にインストールされているソフトウェアを表しています。

時にはいくつかのソフトウェアがアンインストールされる場合もあります。【**コレクションの整合性チェックを実行する**】チェックボックスをオンにすると、

SoftInstallコレクションに不在のソフトウェアは、Asset Managementアプリケーションのデータベースから削除されます。

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

図 13.4. 整合性チェック - コレクションのフィルタ

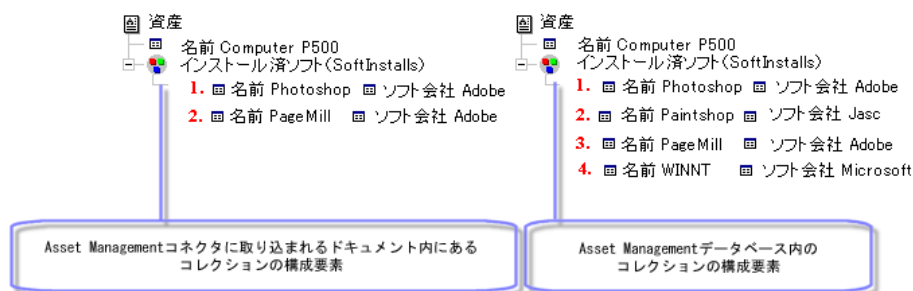
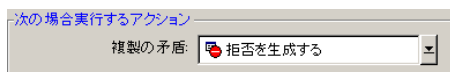


図 13.5. 整合性チェック - ターゲットコレクションの構成要素の削除



このドロップダウンリストでは、複製（レプリケーション）の矛盾が起きた場合にConnect-Itが実行するアクションを設定します。

複製の矛盾が起きた場合、以下の3つのオプションがあります。

- [拒否を生成する]
- [ログに警告を記録する]

- [上書きする]

識別キーに関するエラーメッセージ

シナリオのテスト時または始動時に、Connect-Itログは識別キーの性質に関するメッセージを生成します。

識別キーを不適切に選択すると、整合性に以下の影響を及ぼします。

- 遅い整合性チェック
例：外部アプリケーションのデータベース内で、インデックスとして使用されていないフィールドを選択したとします。このフィールドの検索に時間がかかるため、整合性チェックが遅れます。
- 不安定な整合性
例：重複する値の入力可能なフィールドを選択したとします。インポートする際、ターゲットアプリケーションのデータベースでは、1つのレコードしか更新されない恐れがあります。
例2：複合インデックスのうち、識別キー用に一部のフィールドしか選択しなかったとします。この場合レコードは適切に識別されない恐れがあります。

14 | Peregrine Systemsコネクタ

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

Action Request Systemコネクタ

ベースコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

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

Action Request Systemコネクタの互換性

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

- 3.xx
- 4.xxから4.5
- 5.0

Action Request Systemコネクタ：ダイナミックライブラリ

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

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

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

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

Action Request Systemコネクタの既知の制約点

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

- テキスト（text）
- ダイアリ（diary）

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

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

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

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

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

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

- FileName
添付ファイルの完全パスを含みます。

- AttachMemoVal
添付テキストファイルの場合。このフィールドはファイルの内容を含みます。
- 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おく必要はなくなりました。

読み取り（生成）モードでは、以下のマッピングを実行します。





- AttachMemoVal フィールドをmemoタイプのフィールドへマップする。
 - AttachBlobValフィールドをblobタイプのフィールドへマップする。
 - FieldNameフィールドをblobタイプのフィールドへマップする。
- マッピングの編集用ウィンドウで、**[ファイルからBLOBを読み込む]** オプションが選択されているかどうか確認します。
このオプションの詳細については、「ドキュメントタイプのマッピング」の章の「ターゲット要素をソース要素にマップする」の節、「Blobタイプのフィールドのマッピング」を参照してください。

Action Request Systemコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 14.1. Action Request Systemコネクタの設定

	コネクタに名前を付け、役割を説明する
	接続用パラメータを定義する
	高度な設定
	キャッシュを設定する

- ☒ _____ 再接続のパラメータを設定する
- ☒ _____ サーバとの時間差を指定する
- ☒ _____ スケジュールのポイントを設定する
- ☒ _____ ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

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

接続用パラメータを定義する

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

[サーバ名]

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

[ログイン]

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

[パスワード]

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

[使用するダイナミックライブラリ]

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

[スキーマの最終変更フィールドのリスト] (高度な設定)

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

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


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

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


警告:

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

スキーマと最終変更日用のフィールドを新規に関連付けるには、

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

既存のフィールドを削除するには、

- 1 フィールドに対応する行を選択します。
- 2  をクリックします。

高度な設定

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

スケジュールのポイントを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

生成用ルール - Action Request Systemコネクタ

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

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

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

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

追加情報 - Action Request Systemコネクタ

本節ではコネクタに関する追加情報を説明します。

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

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

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

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

用例シナリオ - Action Request Systemコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- itsm-am/cmpo.scn
- itsm-am/sharedat.scn
- rsm/idd2rsm.scn
- rsm/ind2rsm.scn

Asset Managementコネクタ

ベースコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

本節ではAsset Managementコネクタについて説明します。

Asset Managementコネクタの互換性

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

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

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

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

Asset Managementアプリケーションの設定に関する注意事項

「amdb.ini」ファイル

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。

任意管理項目のテーブル

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

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

例：

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

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

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

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

Asset Managementデータベースの必須フィールド

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

整合性チェック時に、Asset Managementコネクタの使用可能なドキュメントタイプ内の各構造体は、1つのレコードに当たります。この構造体内の1要素が必須フィールドであるにも関わらず入力されていない場合は、要素の所属する構造体は拒否されます。

例：

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

API : DLL

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

Asset Managementコネクタの制約点

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










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

Asset Managementコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 14.2. Asset Managementコネクタの設定

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

コネクタに名前を付け、役割を説明する

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

接続を選択する

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

高度な設定



警告:

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

[使用するダイナミックライブラリ]

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

[テーブルの主キーを表示する]

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

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

Connect-Itのセッションの終了時に接続を切断する

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

スケジュールのポイントを設定する

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

トランザクションを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

Asset Managementコネクタの生成用ルール

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

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

これらの句はAQL言語のシンタックスに従う必要があります、詳細については「AQL言語」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)

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)」章の「取り込み用ルール」の節を参照してください。

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

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

用例シナリオ - Asset Managementコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- 1 itsm-am /cmpo.scn
- 2 itsm-am /sharedat.scn
- 3 scac /sc3ac36 /scac.scn
- 4 scac /sc3ac40 /scac.scn
- 5 scac /sc3ac41 /scac.scn
- 6 scac /sc4ac40 /scac.scn
- 7 scac /sc4ac41 /scac.scn
- 8 ind /indac36 /indac.scn
- 9 ind /indac40 /indac.scn

10 ind /indac41 /indac.scn
11 idd /iddac36 /iddac.scn
12 idd /iddac40 /iddac.scn
13 idd /iddac41 /iddac.scn
14 amo /amo2am.scn
15 ild /ild2am.scn
16 ild /ild65am.scn
17 ldap /ac /complete.scn
18 ldap /ac /simple.scn
19 ac36 /finreque.scn
20 ac36 /finconfi.scn
21 ac36 /hewemplo.scn
22 ac40 /finconfi.scn
23 ac40 /finreque.scn
24 ac40 /hewemplo.scn
25 mqseries /acmq /mqtoprgn.scn
26 mqseries /acmq /prgntomq.scn
27 ntac36 /addcpu.scn
28 ntac36 /adduser.scn
29 ntac36 /adduser.scn
30 ntac40 /addcpu.scn
31 ntac40 /adduser.scn
32 ntac41 /addcpu.scn
33 ntac41 /adduser.scn
34 scac /sc3ac40 /scac.scn
35 scac /sc3ac41 /scac.scn
36 scac /sc4ac40 /scac.scn
37 scac /sc4ac41 /scac.scn
38 acscups /dataprop /01-SCcompany2ACamCompany.scn
39 acscups /dataprop /02-SClocation2ACamLocation.scn
40 acscups /dataprop /03-SCdept2ACamEmplDept.scn
41 acscups /dataprop /04-SCcontacts2ACamEmIpDept.scn
42 acscups /dataprop /05-SCdevtype2ACamNature.scn
43 acscups /dataprop /06-SCocmlcat2ACamNature.scn
44 acscups /dataprop /07-SCmodel2ACamModel.scn
45 acscups /dataprop /08-SCdevtype2ACamModel.scn
46 acscups /dataprop /09-SCdevice2AACamPortfolioAndamComputer.scn

47 acscups \dataprop\10-SClaptop2ACamPortfolioAndamComputer.scn
48 acscups \dataprop\11-SCworkstation2ACamPortfolioAndamComputer.scn
49 acscups \dataprop\12-SCpc2ACamAndamComputer.scn
50 acscups \dataprop\13-SCserver2ACamAndamComputer.scn
51 acscups \dataprop\14-SCphone2ACamPortfolioAndamPhone.scn
52 sms \sms1am.scn
53 sms \sms1amlt.scn
54 sms \sms2am.scn
55 sms \sms2amlt.scn
56 sms \sms2ac41.scn
57 tim \tim2am
58 uagw \uagw.scn

InfraTools Desktop Discoveryコネクタ

ベースコネクタ

• 生成 (ソース)

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

InfraTools Desktop Discoveryコネクタの互換性

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

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

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

FSFファイル

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

SAIファイル

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

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

Desktop DiscoveryコネクタのCDTファイル

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

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

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

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

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

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

• 学習モードを使用する

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

• 学習モードを使用しない

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

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








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

InfraTools Desktop Discoveryコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 14.3. InfraTools Desktop Discoveryコネクタの設定

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

コネクタに名前を付け、役割を説明する





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

FSFファイル

このページでは、データソースと学習モードのパラメータを選択できます。

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

FSFファイルが保存されているフォルダを指定するには、

- 1  をクリックします。
 - 2 コンピュータでのフォルダの完全パスを入力します。
フォルダのパスを変更するには、パスを直接ダブルクリックし、新規のパスを入力します。
フォルダを削除するには、フォルダに対応する行を選択してから  をクリックします。
-  と  ボタンを使うと、フォルダの順番を変更できます。
- [SAIファイルフォルダ]** : SAIファイルが保存されているフォルダを指定します。
- [ユーザ追加項目ファイル]** : ユーザが追加する特定の項目が保存されている「.cdt」ファイルに名前をつけます。

警告:

FSFファイルを含むフォルダのパス

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

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、「[Connect-Itサービスの作成](#)」の章、「[WindowsでConnect-Itサービスを作成する](#)」 / 「[シナリオのパス](#)」を参照してください。

学習モード

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

- IDDのスキャナの作成時にユーザが設定する【UserAndAssets】構造体の項目
- ユーザのIT資産に応じた【SMBIOS】と【DMI】構造体の項目
上記2つの構造体にある要素は、学習モードでコネクタを使用する際に識別されます。
- ユーザが保存したファイルを含む【StoredFiles】構造体の項目

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

学習モードは以下の場合に使用します。

- InfraTools Desktop Discoveryコネクタが一番最初に開かれ使用される時。この後、マッピングの段階でユーザが追加した新規項目を使用できるようになります。

- ユーザがスキャンの内容を変更したため、この新規項目に応じてマッピングを変更する必要がある場合。

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

- ドキュメントを拒否する
- ドキュメントログ内に警告を作成する

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

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

このエラーを訂正するには、

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

FSFファイルから抽出するデータ

このページでは、FSFファイルから抽出する要素を選択します。

例：

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

[格納されたファイル]

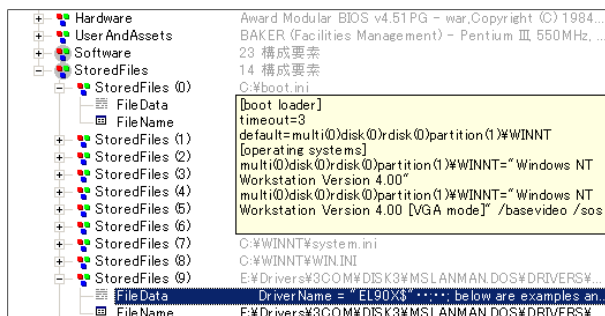
[格納されたファイル] を選択すると、InfraTools Desktop Discoveryコネクタは格納されたファイルの内容を取得します。

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

例 14.1. 格納されたファイルの例

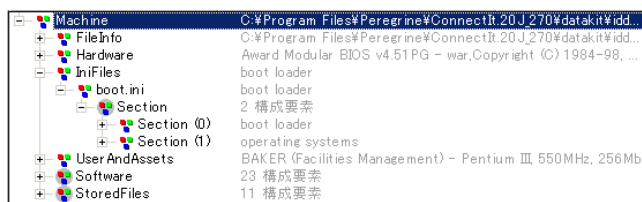
以下は、データキットの「london01.fsf」ファイルに含まれている、格納されたファイル「boot.ini」の例です。

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



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

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



FSFデータをフィルタで抽出する

【ソフトウェア認識フィルタ】



警告:

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

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

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

スキャンされたファイルのフィルタ

警告:

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

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

以下の3つのオプションがあります。

- **【何も取得しない】**
このオプションが選択されていると、スキャンされたファイルはFSFファイルから抽出されません。
- **【全ファイルを取得する】**
このオプションが選択されていると、スキャンされたファイルは全てFSFファイルから抽出されます。
- **【次のフィルタを使用する】**
このオプションが選択されていると、下のフィールドで指定するファイルの拡張子（セミコロンで区切って記入）に一致するファイルが、FSFファイルから抽出されます。

警告:

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

処理後のアクションを定義する

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



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

- FSFファイルをフォルダに保存する
- ファイルを削除する
- パスを指定したフォルダ内にファイルを移動させる

FSFファイルを削除または移動する場合、InfraTools Desktop Discoveryコネクタは、FSFファイルが正常に処理されたことを示す処理レポートを取り込む必要があります。この処理レポートは、InfraTools Desktop Discoveryコネクタが生成し

たドキュメント（FSFファイルに相当）を正常に取り込んだ別のコネクタから、InfraTools Desktop Discoveryコネクタに転送されます。

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



警告:

処理後にファイルを削除する

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

例：

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

FSFファイルの基準日とポインタ

[フォルダに保存する] オプションが選択されている場合

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

例：

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

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

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

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

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

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

[フォルダの相対パスを使用する]

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

スケジュールのポイントを設定する

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

ピボットドキュメントタイプの使用

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

InfraTools Desktop Discoveryコネクタのルール

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

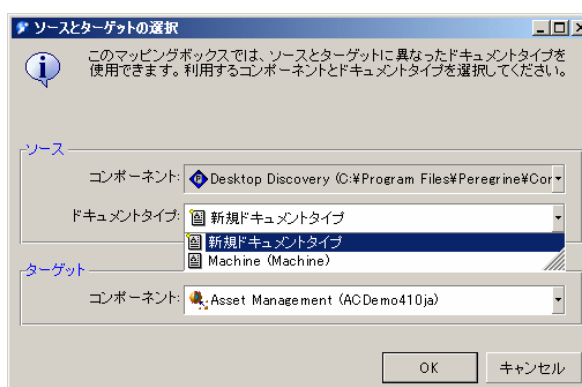
追加情報

本節ではコネクタに関する追加情報を説明します。

複数のドキュメントの生成

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

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



用例シナリオ - Infratools Desktop Discoveryコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- `idd /ddac36 /ddac.scn`
- `idd /ddac40 /ddac.scn`
- `idd /ddac41 /ddac.scn`

- `idd /dditsm /ddam.scn`
- `idd /dditsm /ddhd.scn`
- `idd /dds /ddsc.scn`
- `rsm /dd2rsm.scn`

InfraTools Managementコネクタ

ベースコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

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

InfraTools Managementコネクタの互換性

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

以下のバージョンが使用可能です。

- InfraTools Remote Controlバージョン5.xx






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

InfraTools Managementコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 14.4. InfraTools Managementコネクタの設定

	コネクタに名前を付け、役割を説明する
	接続を選択する
	高度な設定
	再接続のパラメータを設定する
	サーバとの時間差を指定する

- ☒ _____ スケジュールのポイントを設定する
- ☒ _____ トランザクションを設定する
- ☒ _____ キャッシュを設定する
- ☒ _____ ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

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

接続を選択する

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

高度な設定



警告:

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

[Connect-Itのセッションの終了時に接続を切断する]

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

[サーバとの時間差]

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

[使用するダイナミックライブラリ]

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

サーバへの再接続用オプション

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

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

再接続オプション用のフィールドは次の2つです。

- **[基本待ち時間]**

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

- **[制限時間]**

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

 **注意:**

再接続に成功すると、

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

スケジュールのポイントを設定する

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

トランザクションを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

InfraTools Managementコネクタの生成用ルール

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

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

これらの句はAQL言語のシンタックスに従う必要があります、詳細については「AQL言語」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)

InfraTools Managementコネクタの取り込み用ルール

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

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

用例シナリオ - InfraTools Managementコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- im/ntsec2im.scn

InfraTools Network Discoveryコネクタ

ベースコネクタ

- 生成 (ソース)

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

InfraTools Network Discoveryコネクタの互換性

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

- Network Discovery 4.0X、4.10と4.20

注意事項

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

InfraTools Network Discoveryコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 14.5. InfraTools Network Discoveryコネクタの設定

コネクタに名前を付け、役割を説明する

接続用パラメータを定義する

スケジュールのポイントを設定する

キャッシュを設定する

ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

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

接続用パラメータを定義する

[サーバ名]

Network Discoveryサイトのアドレスを入力します。

イベントのURL (高度な設定オプション)

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

 注意:

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

周辺機器のURL (高度な設定オプション)

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

[ログイン]

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

[パスワード]

ログイン用のパスワードを入力します。

スケジュールのポイントを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

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

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

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

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

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

例：

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

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

以下のWHERE句を作成します。

```
category = "error"
```

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

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

例：

イベントの数を1クエリにつき50に指定する場合（最大値100）、

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

```
nrows=50
```

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

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

```
Limit = 50
```

 **注意:**

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

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

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

非スケジュールモード

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

スケジュールモード

EVENT

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

DEVICE

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

追加情報 - InfraTools Network Discoveryコネクタ

本節ではコネクタに関する追加情報を説明します。

マッピングスクリプト内で使用されるフィールド

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

- EventID
- NMID
- MACAddress


例

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

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

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

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

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

例：EVENT.EventIDフィールド

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

用例シナリオ - InfraTools Network Discoveryコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- ind /ndac36 /ndac.scn
- ind /ndac40 /ndac.scn
- ind /ndac41 /ndac.scn
- rsm /nd2rsm.scn

Peregrine Desktop Inventory (PDI) コネクタ

ベースコネクタ

• 生成 (ソース)

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

PDIコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 14.6. PDIコネクタの設定

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



警告:

ファイルまたはフォルダのパス

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

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、「Connect-Itサービスの作成」の章、「WindowsでConnect-Itサービスを作成する」/「シナリオのパス」を参照してください。

コネクタに名前を付け、役割を説明する

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

接続のプロトコルを選択する

以下のオプションがあります。

- [FTPサーバ]
- [ローカルファイル/ネットワークファイル]

FTPサーバ

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

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

FTPサーバ接続

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

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

[ファイルを読み取る]

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

[フォルダを読み取る]

このオプションを選択する場合は、以下の2フィールドに入力します。

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

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

[サブフォルダを読み取る]

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

注意:

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

処理後のアクションを定義する

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

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

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

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

ローカルファイル/ネットワークファイル

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

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

ファイルまたはフォルダを選択する

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

- [ファイルを読み取る]
- [フォルダを読み取る]

[ファイルを読み取る]

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

[フォルダを読み取る]

このオプションを選択する場合は、以下の2フィールドに入力します。

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

[サブフォルダを読み取る]

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

処理後のアクションを定義する

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

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

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

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

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

DTDを選択する

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

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

キャッシュを設定する

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

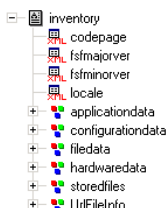
ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定」章の「ピボットドキュメントタイプの使用」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 14.1. PDIコネクタ - 生成用ドキュメントタイプ



用例シナリオ - PDIコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- pdi/pdiac41/pdiac.scn

ゲートウェイ3.xコネクタ

ベースコネクタ

• 生成 (ソース)

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

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

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

これらのゲートウェイでは以下のアプリケーションからデータを取得できます。

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

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

ゲートウェイコネクタのODBC互換性

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

重要項目:

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

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

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

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739

- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301
- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500

ゲートウェイコネクタの既知の制約点

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

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

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

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

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

ゲートウェイコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 14.7. ゲートウェイコネクタの設定

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

前提条件

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

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

コネクタに名前を付け、役割を説明する

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

接続タイプを選択する

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

接続を選択する

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

高度な設定

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

スケジュールのポイントを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

ゲートウェイ3.xコネクタの生成用ルール

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

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

ゲートウェイ3.xコネクタの使用可能なドキュメントタイプ

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

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

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

用例シナリオ - ゲートウェイ3.xコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- uagw/uagw.scn

ServiceCenterコネクタ

ベースコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

本節ではServiceCenterコネクタについて説明します。

ServiceCenterコネクタの互換性

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

コネクタは以下のバージョンと互換性があります。

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

ServiceCenterの設定に関する注意事項

入力イベントの作成

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

例：

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

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

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

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

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

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

入力イベントの同期をとる

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

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

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

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

ServiceCenterコネクタのUNIXでの使用

ServiceCenterコネクタをUNIXで使用するには、









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

ServiceCenterコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 14.8. ServiceCenterコネクタの設定

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

コネクタに名前を付け、役割を説明する

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

接続を設定する

【サーバ名】：以下のように名前を入力します。

例：

[サーバ名].[ServiceCenterクライアントサーバのポート番号]

SC.peregrine.com.12670

 注意:

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

【ログイン】

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

【パスワード】

ログイン用のパスワードを入力します。

【テスト】

このボタンを押すと接続をテストできます。

接続をテストするには、

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

高度な設定

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

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

スケジュールのポイントを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

ServiceCenterコネクタの生成用ルール

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

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

演算子

=

≠

>

<

AND

OR

NOT

#

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

例：

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

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

以下のWHERE句を作成します。

例：

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

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

追加情報

本節ではコネクタに関する追加情報を説明します。

ServiceCenterコネクタが適切に処理したイベントを削除する

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

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

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

処理レポートの詳細については、「[処理レポート](#)」の章を参照してください。

ServiceCenterと共に処理レポートを使用する例

この例を模倣するには、Peregrine Systemsのサポートから以下の3つのファイル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 **[オプション/インポート/読み込み]**を選択します。
- 4 以下の2ファイルを読み込みます。

- delevreg.unl
- delevmap.unl

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

ServiceCenter内にPMOタイプの出力イベントを作成するには、

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

処理レポート

「delete.scn」シナリオは以下の内容を使用します。

- PMOタイプの出力イベントを処理するServiceCenterソースコネクタ
- ServiceCenterコネクタが生成するドキュメントを変換するマッピングボックス

- ヘルプデスクチケットのテーブル (amTicket) 内にレコードを作成するAsset Managementターゲットコネクタ
- Asset Managementコネクタが生成する処理レポートを取り込むマッピングボックス

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

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

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

表 14.9. Asset Managementコネクタの処理レポートと、ServiceCenterコネクタのConnectItDelドキュメントタイプ間のマッピング

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

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

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

\$ParentDoc\$変数の詳細については、「処理レポート」章の「既製シナリオ内で処理レポートを使用する」節の「\$ParentDoc\$変数の使用」を参照してください。

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

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

用例シナリオ - ServiceCenterコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- idd /dds /ddsc.scn
- scac /sc3ac36 /scac.scn
- scac /sc3ac40 /scac.scn
- scac /sc3ac41 /scac.scn
- scac /sc4ac40 /scac.scn
- scac /sc4ac41 /scac.scn
- scauto /scacfg /scacfg.scn
- tim /tim4sc4

15 | アプリケーションコネクタ

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

Lotus Notesコネクタ

オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

Lotus Notesコネクタでは、以下の操作が可能です。

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

Lotus Notesコネクタの互換性

Lotus Notesコネクタは、Lotus Notesのバージョン5.0とのテストに成功していません。

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

 **注意:**

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

Lotus Notesコネクタの既知の制約点

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

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





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

Lotus Notesコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 15.1. Lotus Notesコネクタの設定

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

コネクタに名前を付け、役割を説明する

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

接続用パラメータを定義する

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

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

[Lotus Notes IDファイル]

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



警告:

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

[パスワード]

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

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

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

[サーバ名]

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

[データベース]

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

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

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

スケジュールのポイントを設定する

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

ピボットドキュメントタイプの使用

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

Lotus Notesコネクタの生成用ルール

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

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

WHERE句

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

以下の表はWHERE句で使用可能な演算子と、オペランドのリストです。

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

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

フォーム名

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

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

Lotus Notesコネクタの取り込み用ルール

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

[整合性] タブの詳細

フォーム名

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

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

【識別キー】

☞ 「識別キー」

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

フィールドまたは構造体を識別キーとして指定するには、

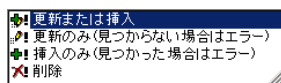
- 要素を作業枠内で選択します。
- 【識別キー】チェックボックスをオンにするか、または取り込み用ドキュメントタイプが表示される枠内の透明なキーを、直接クリックします。

【大文字小文字の区別】

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

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

[整合性チェックのタイプ]



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

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

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

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

Lotus Notesコネクタに関する追加情報

本節ではコネクタに関する追加情報を説明します。

バイナリLotus Notesフィールド

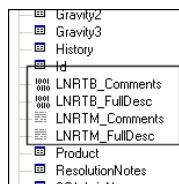
Lotus Notesデータベースには、バイナリ型のフィールドを含むドキュメントがあります。バイナリフィールドは、テキストの他、色やフォントなどの書式情報などが含まれています。

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

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

注意:

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



スケジュールモードでのデータ処理

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

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

Notes文書のプロパティには2つの変更日があります。

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

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

用例シナリオ - Lotus Notesコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- `ldap/notes/notes.scn`

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。

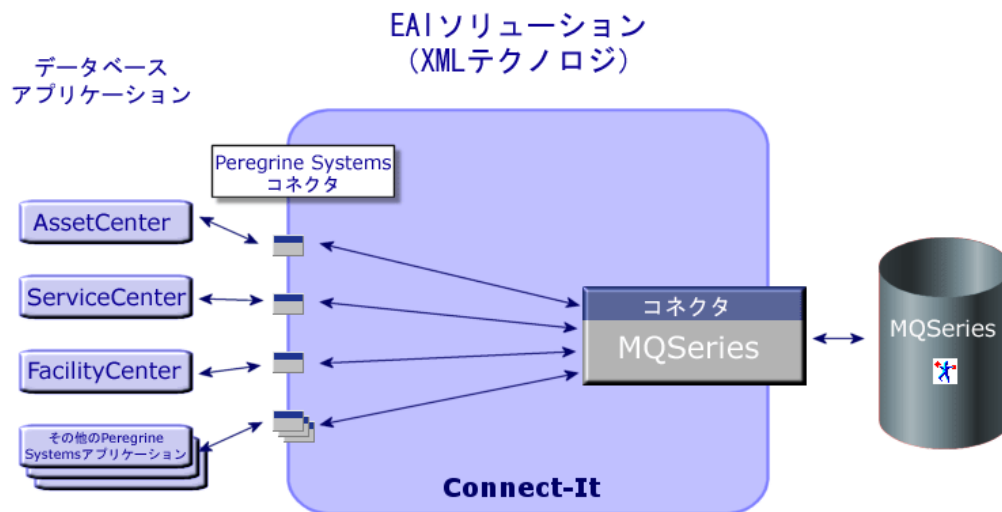
MQSeriesの新機能は、システムやネットワークインフラストラクチャに欠陥がある場合でも、データの送信を保証します。

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のアプリケーション間で交換されます。

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

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

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



互換性

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

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」章の「コネクタを設定する」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 15.2. MQSeriesコネクタ（読み取り）の設定

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

コネクタに名前を付け、役割を説明する

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

処理モードを選択する

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

キューマネージャへ接続する

このページでは以下の内容を入力します。

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

キューマネージャへ接続する

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

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

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

- **【キューマネージャ名】**

キューマネージャ名を入力します。MQSeriesコネクタ付属の既製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。



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

MQSeriesサーバへの再接続用パラメータ

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

再接続オプション用のフィールドは以下の通りです。

- **【基本待ち時間】**

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

- **【制限時間】**

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



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

【サーバとの時間差】

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

キューの選択

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

[モデルキュー] オプション

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

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

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

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

処理後のアクションを定義する

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

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

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

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

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

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

DTDを選択する

XMLファイル进行处理するために、MQSeriesコネクタはDTD（文書型定義）を使用する必要があります。このページでは、使用するDTDを【DTD】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】フィールドには空のファイル名を指定します。学習モードでデータ処理が終わった時点で、このファイルの内容が充填されます。このフィールドに記入された名前が既存ファイルの名前と一致すると、Connect-ItはこのDTDファイルに自動的に番号を付けて保存します（例：request_01.dtd）。コネクタが数回学習モードで起動されると、起動の度に作成されるそれぞれのDTDには、作成の順番に従って番号が付けられます。（例：request_01.dtd、request_02.dtd、request_03.dtd。「request_03.dtd」が最も最近に保存されたDTDファイルです。）

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

ファイルの場所

以下の3つのオプションがあります。

- 【ローカル/ネットワーク】
Connect-Itがインストールされているコンピュータやネットワーク上の、DTDの完全パス名を入力します。
- 【FTP】
【サーバ名】、【ログイン】、【パスワード】と【パス】フィールドにFTPパラメータを入力します。
【パス】フィールドでは、DTDのサーバ上の場所を指定します。例えば、DTDがサーバ「ftp.mycompany.com」上の「myfolder/dtd」フォルダ内にある

場合、**【サーバ名】**フィールドに「ftp.mycompany.com」を入力し、**【パス】**フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。

 **注意:**

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

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

 **注意:**

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

[第1レベルの各コレクションに使用可能なドキュメントタイプを発行する]

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

例:

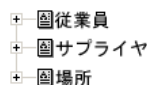
あるDTDのルート要素が「**会社**」で、第1レベルに3つのコレクション「**従業員**」、「**サプライヤ**」と「**場所**」があるとします。上記のオプションが選択されていないと、コネクタが発行する使用可能なドキュメントタイプのルート要素は「**会社**」になります。

注意:

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



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



ルート要素を指定する

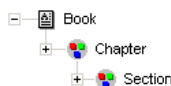
DTDは複数の要素から構成されており、これらの要素は更に別の要素を含んでいます。（例：文書編集用のDTDでは、複数の「Section」要素が複数の「Chapter」要素に含まれており、更に「Chapter」要素は1つの要素「Book」に含まれています。）他の要素に含まれることがない要素は、ルート要素と呼ばれます。DTDは1つまたは複数のルート要素から構成されることもあれば、DTDに全くルート要素が存在しないこともあります。

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

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

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

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



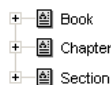
注意:

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

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

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

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



注意:

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

学習モード

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

注意:

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

学習モードを使用するには、


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

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

XMLドキュメントとDTD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内に定義された要素に一致するはずですが、メッセージに新規の要素が現れることもあります。要素の少ないメッセージを学習モードで処理した時に得られたDTDを使用する際に、このようなXMLとDTDの矛盾が頻繁に起こります。

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

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

スケジュールのポイントを設定する

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

ピボットドキュメントタイプの使用

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 15.3. MQSeriesコネクタ（書き込み）の設定

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

コネクタに名前を付け、役割を説明する

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

処理モードを選択する

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

キューマネージャへ接続する

このページでは以下の内容を入力します。

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

キューマネージャへの接続

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

- **【サーバ名】**
このフィールドには、ネットワーク上のMQSeriesサーバのDNS名、またはIPアドレスを入力します。
- **【接続ポート】**
キューマネージャのポート番号を指定します。このポートのデフォルト値は「1414」です。
- **【チャンネル名】**
キューマネージャが通信パスとして使用する接続チャンネル名を入力します。デフォルト値は「SYSTEM.DEF.SVRCONN」です。
- **【キューマネージャ名】**
キューマネージャ名を入力します。MQSeriesコネクタ付属の既製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。

 **注意:**

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

MQSeriesサーバへの再接続用パラメータ

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

再接続オプション用のフィールドは以下の通りです。

- **【基本待ち時間】**
【基本待ち時間】は、コネクタが何秒後にMQSeriesサーバへの再接続を実行するかを指定します。再接続に失敗すると、Connect-Itは基本待ち時間を2倍

にして接続を再試行します。例えば、基本待ち時間が2秒の場合、2回目の再接続は4秒後に実行され、3回目の再接続は8秒後に実行されます。

- **[制限時間]**

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

 **注意:**

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

[サーバとの時間差]

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

キューの選択

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

[モデルキュー] オプション

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

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

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

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


DTDを選択する

XMLファイル进行处理するために、MQSeriesコネクタはDTD（文書型定義）を使用する必要があります。このページでは、使用するDTDを【DTD】フィールドで指定できます。

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

ファイルの場所

以下の3つのオプションがあります。

- 【ローカル/ネットワーク】
Connect-Itがインストールされているコンピュータやネットワーク上の、DTDの完全パス名を入力します。
 - 【FTP】
【サーバ名】、【ログイン】、【パスワード】と【パス】フィールドにFTPパラメータを入力します。
【パス】フィールドでは、DTDのサーバ上の場所を指定します。例えば、DTDがサーバ「ftp.mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、【サーバ名】フィールドに「ftp.mycompany.com」を入力し、【パス】フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。
-
-  **注意:**
スラッシュ（/）はサーバ名の最後には入れませんが、DTDパスの最初に入れます。
-
- 【HTTP】
【サーバ名】、【ログイン】、【パスワード】と【パス】フィールドにHTTPパラメータを入力します。
【パス】フィールドでは、DTDのサーバ上の場所を指定します。例えば、DTDがサーバ「http://mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、【サーバ名】フィールドに「http://mycompany.com」を入力し、【パス】フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。

 **注意:**

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

[XMLファイルで取得したDTDを参照しない]

このオプションはデフォルトのオプションです。

[取得したDTDをXMLファイル内に書き込む]

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

[取得したDTDを外部ファイルに書き込み、XMLファイル内で参照する]

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

 **警告:**

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

DTD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が使用されることになります。

新規のDTDが外部ファイルに書き込まれるようにするには、シナリオを停止し、再起動しなければなりません。

ルート要素を指定する

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

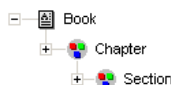
ルート要素（つまり、MQSeriesコネクタに発行される使用可能なドキュメントタイプに当たります）を指定するためには、以下の2つのオプションがあります。

- [DTD内に見つかった各ルート要素に使用可能なドキュメントタイプを発行する(推奨オプション)]
- [ユーザの選択する各ルート要素に使用可能なドキュメントタイプを発行する(コンマで値を区切る)]

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

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

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



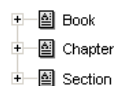
注意:

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

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

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

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



注意:

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

スケジュールのポイントを設定する

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

ピボットドキュメントタイプの使用

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

MQSeriesコネクタの生成用ルール

MQSeriesコネクタの生成用ルールにより、設定で指定されたキュー内でコネクタが読み取るメッセージに、フィルタをかけることができます。



以下の内容を指定してルールを作成します。

- WHERE句
- 取得用のオプション

ルールを作成するには、

- 1 MQSeriesコネクタをダブルクリックします。

- 2 ダイアログボックスの [生成用ドキュメントタイプ] 枠内で、ドキュメントタイプを1つ選択します。
- 3 生成用ドキュメントタイプのルート要素を選択します。
- 4 生成用ルールを入力します。

WHERE句

WHERE句では、MQSeriesコネクタに読み取られるメッセージにフィルタをかけます。WHERE句は以下のMQSeriesシンタックスで作成されます。

[キーワード]=[値]

以下のキーワードがあります。

- MsgID (メッセージのID)
- GroupID (メッセージグループのID)
- CorrelID (相関ID)

キーワードは16進数形式で入力します。

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

- CONNIT.MQAM.REQUEST.ACK
(434f4e4e49542e4d51414d2e524551554553542e41434b) : ExtRequestAcksドキュメントタイプ用
- CONNIT.MQAM.RECEIPT.ACK
(434f4e4e49542e4d51414d2e524543454950542e41434b) : ExtReceiptAcksドキュメントタイプ用
- CONNIT.MQAM.VENDOR (434f4e4e49542e4d51414d2e56454e444f52) : Vendorsドキュメントタイプ用
- CONNIT.MQAM.COSTCENTER
(434f4e4e49542e4d51414d2e434f535443454e544552) : CostCentersドキュメントタイプ用

MQSeriesコネクタ付属のMQSeries - ServiceCenterシナリオでは、相関ID (CorrelID) 1つがExternal Contacts生成用ドキュメントタイプ用に使用されます : CONNIT.MQSC.CONTACT (434f4e4e49542e4d51414d2e434f535443454e544552)

これらの相関IDは、外部アプリケーションがPEREGRINE.INキューに置くMQSeriesメッセージに適用されます。

取得オプション

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。

以下の4つのオプションがあります。

- [メッセージを論理順序で取得する]
このオプションを選択すると、全ての物理メッセージを論理順序で取得できます。
- [完全な論理メッセージを取得する]
このオプションを選択すると、同じ論理メッセージにある全ての物理メッセージを取得できます。
このオプションが選択されていないと、以下のオプションが自動的に選択されます。
- [論理メッセージの全セグメントが使用可能な場合にのみメッセージを取得する]
- [グループの全メッセージが使用可能な場合にのみメッセージを取得する]

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

ルールを作成するには、

- 1 MQSeriesコネクタをダブルクリックします。
- 2 ダイアログボックスの [取り込み用ドキュメントタイプ] 枠内で、ドキュメントタイプを1つ選択します。
- 3 取り込み用ドキュメントタイプのルート要素を選択します。
- 4 取り込み用ルールを入力します。

または

- 1 書き込みモードのMQSeriesコネクタにリンクしているマッピングボックスを、ダブルクリックします。
- 2 MQSeriesコネクタがターゲットコネクタとしてリンクされているマッピングを、編集または作成します。
- 3 [メッセージのオプション] タブを選択します。
- 4 取り込み用ルールを入力します。

自動セグメント化

メッセージのセグメント化を許可すると、キューで許容されている最大サイズよりも大きいメッセージは、複数の物理メッセージに分割されます。これらの物理メッセージは1つの論理メッセージ内にまとめられます。

メッセージのセグメント化を禁止すると、キューで許容されている最大サイズよりも大きいメッセージは、MQSeriesコネクタに拒否されます。

ID指定

このオプションでは特定のIDを指定できます。

- グループID
- 関連ID

IDは16進数形式で入力します。

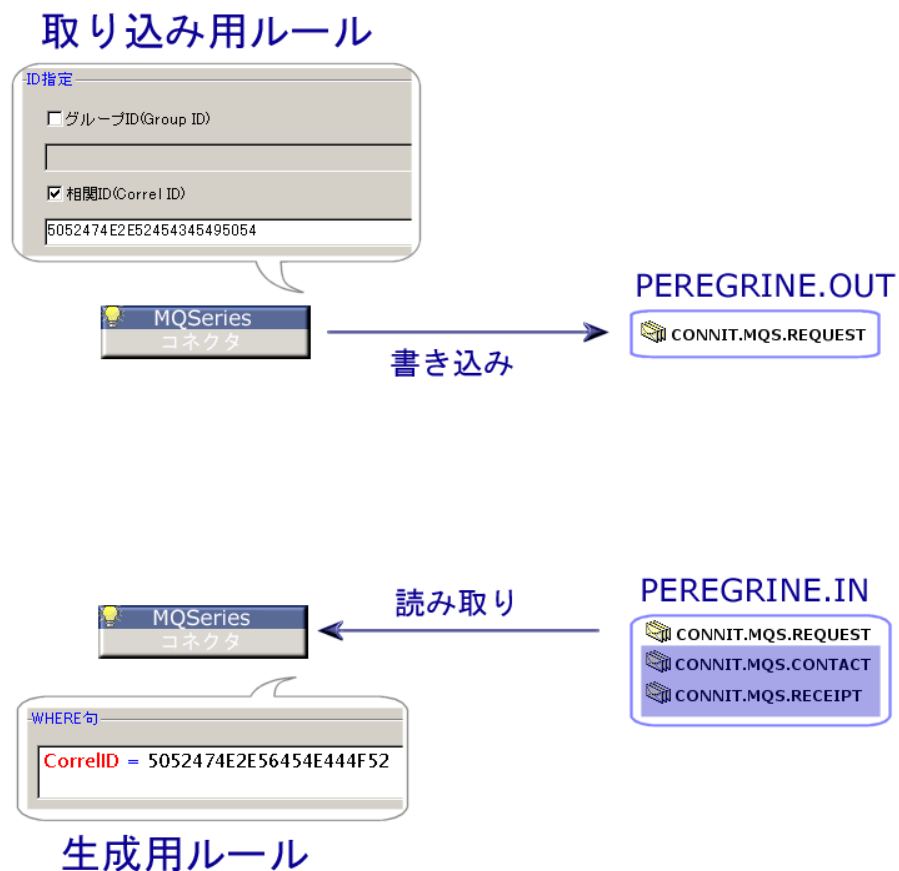
MQSeriesコネクタ付属のMQSeries - Asset Managementシナリオでは、取り込み用ドキュメントタイプ用の関連ID (CorrelID) は以下の通りです。

- CONNIT.MQAM.REQUEST (434f4e4e49542e4d51414d2e52455155455354) : Requestドキュメントタイプ用
- CONNIT.MQAM.RECEIPT (434f4e4e49542e4d51414d2e52454345495054) : Receiptドキュメントタイプ用

MQSeriesコネクタ付属のMQSeries - ServiceCenterシナリオでは、関連ID (CorrelID) 1つがContacts from ServiceCenter取り込み用ドキュメントタイプ用に使用されます : CONNIT.MQSC.CONTACT (434f4e4e49542e4d51414d2e434f535443454e544552)

指定されたIDは、PEREGRINE.OUTキュー内の発注や受領伝票に対応するメッセージを識別します。メッセージを読み取る外部アプリケーションは、メッセージを取得するために同じIDを使用しなければなりません。

図 15.1. MQSeriesコネクタ - 相関IDによるMQSeriesメッセージの識別



用例シナリオ - MQSeriesコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- mqseries /acmq /mqtoprgn.scn
- mqseries /acmq /prgntomq.scn
- mqseries /scmq /mqsc.scn

NTセキュリティコネクタ

ベースコネクタ

• 生成 (ソース)

本節ではNTセキュリティコネクタについて説明します。このコネクタは、Connect-Itがインストールされているコンピュータの、NTドメインに関する情報を取得します。この情報はコンピュータ、ユーザや、NTドメインのユーザグループに関するものです。

このコネクタはドキュメントを生成できますが、取り込むことはできません。

NTセキュリティコネクタの互換性

NTセキュリティコネクタは、Windows 32ビットのオペレーティングシステムを使用するコンピュータで機能します。

 **注意:**




新規のMS Active Directoryセキュリティモデルを使用する場合は、LDAPコネクタの使用をお勧めします。このセキュリティモデルはNTセキュリティコネクタにはサポートされていません。

NTセキュリティコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 15.4. NTセキュリティコネクタの設定

	コネクタに名前を付け、役割を説明する
	NTドメインを入力する
	ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

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

NTドメインを入力する

[ドメイン]フィールドで、NTセキュリティ情報をどのドメインから取得するかを指定します。デフォルトで表示されるドメイン名は、Connect-It起動中のNTセッションのドメインに一致します。

注意:

複数ドメインの情報を取得する場合は、処理するドメインの名前をセミコロン(;)で区切って入力します。

コンピュータからアクセス可能な全NTドメインを取得する場合は、アスタリスク(*)を入力します。この操作を行うとコンピュータのメモリに負荷がかかるため、Connect-Itの実行が中断される可能性があります。

警告:

シナリオ図内でコネクタ名の下に表示される短い説明は、常にConnect-Itの現在のドメイン名とユーザ名です。このページで別のドメイン名を入力しても、コネクタの短い説明内には表示されません。

ピボットドキュメントタイプの使用

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

NTセキュリティコネクタの使用可能なドキュメントタイプ

NTセキュリティコネクタには3つの使用可能なドキュメントタイプがあります。

- Machine
- NtDomain
- User

注意:

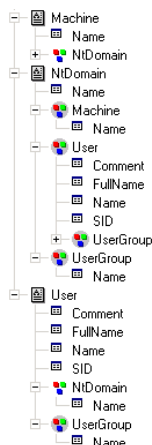
「NtDomain」ドキュメントタイプは、コネクタの設定時に指定されたNTドメインに一致します。

NTセキュリティコネクタの使用可能なドキュメントタイプには、以下のサブノード(構造体とコレクション)があります。

- ドメインのコンピュータ (Machineコレクション)

- ドメインのユーザ (Userコレクション)
- ドメインのユーザグループ (UserGroupコレクション)

図 15.2. NTセキュリティコネクタ - 使用可能なドキュメントタイプ



NTセキュリティコネクタの生成用ルール

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

NTセキュリティコネクタの生成用ルールでは、ドキュメントタイプのコレクションの構成要素にフィルタを適用できます。

NTドメインの要素に、名前フィルタをかける

[名前フィルタをかける] 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 15.5. NTセキュリティコネクタ - 生成用ルールの例

コレクション	フィルタ	取得する値
Machine	Plateforme1	Plateforme1コンピュータ
Machine	a*	名前が「a」で始まるドメインの全コンピュータ
User	Dupon?	ドメインのユーザの内、名前の始めが「Dupon」で、名前の終わりの文字が不明である全ユーザ

用例シナリオ - NTセキュリティコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- im /ntsec2im.scn
- ntac36 /addcpu.scn
- ntac36 /adduser.scn
- ntac40 /addcpu.scn
- ntac40 /adduser.scn
- ntac41 /addcpu.scn
- ntac41 /adduser.scn

16 | プロトコルコネクタ

プロトコルコネクタは、特定の情報プロトコルで処理されるデータ用のコネクタです。

データベースコネクタ

オプションコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

データベースコネクタは、ODBCデータソース、またはOracle、SybaseおよびMySQLデータベースから直接データを処理します。

 **警告:**

ODBC接続を使用するには、ODBCアドミニストレータが先ずインストールされていなければなりません。

データベースコネクタの互換性

データベースコネクタは、以下の表のドライバとODBC Administratorバージョン3.520.4403.2とのテストに成功しています。

ODBC	Oracle
Accessバージョン4.00.4403.02	Oracle 7バージョン2.05.0301
Excelバージョン4.00.4403.02	Oracle 8.0バージョン8.00.0500
Sybase 11.9バージョン3.11.0001	Oracle 8iバージョン8.01.0500
Sybase SQL Anywhereバージョン5.05.052739	
Textバージョン4.00.4403.02	
Microsoft SQL Serverバージョン3.70.08.20	
IBM DB2バージョン7.01.00.00 (バージョン6より後のDB2で)	
INFORMIX 3.33 32 BIT バージョン3.33.00.10688	

データベースコネクタは、全てのODBCドライバのバージョン3.xxと互換性があります。

データベースコネクタの既知の制約点

データベースコネクタを適切に使用するためには、ODBCドライバとODBCアドミニストレータのDLLバージョンが、同一である必要があります。

例：

ODBC Administratorバージョン3.0と、ODBC Microsoft Accessドライババージョン4.00を使用していると、データベースコネクタは機能しません。

サポートされているフィールドのタイプ

データベースコネクタの使用可能なドキュメントタイプは、フィールドから構成されています。各フィールドには、それぞれ特定のフィールドタイプ（テキストフィールド、整数フィールド、Blobフィールドなど）があります。

使用されているデータベースドライバの種類により、データベースコネクタにサポートされるフィールドタイプは異なります。

Oracle 8.0 - Oracle用Microsoft ODBCドライバ、バージョン2.573.6526.00

フィールドタイプ	サポート可/不可
bfile	不可
blob (可変長バイナリフィールド)	可
char	可
clob	可
Date (日付)	可
Float (浮動小数点)	可
integer	可
long整数 (32ビット)	可
long raw	可
nchar	可
nchar	不可
nvarchar	不可
number	可
ms_label	可

MSSQL 7.0 - SQL Server **ドライバ、バージョン** 2000.80.194.00

フィールドタイプ	サポート可 / 不可
binary	可
bit	可
char	可
datetime	可
decimal	可
Float (浮動小数点)	可
image	可
int	可
money	可
nchar	不可
ntext	可
numeric	可
nvarchar	不可
real	可
smalldatetime	可
smallint	可
smallmoney	可
text	可
timestamp	可
tinyint	可
uniqueidentifier	不可
varbinary	可
varchar	可

Sybase12 : Sybase System 11バージョン3.11.00.01







フィールドタイプ	サポート可 / 不可
binary	可
char	可
datetime	可
decimal	可
Float (浮動小数点)	可
image	可
int	可
money	可
nchar	可
ntext	可
numeric	可
nvarchar	可
real	可
smalldatetime	可
smallint	可
smallmoney	可
text	可
tinyint	可
varbinary	可
varchar	可

データベースコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 16.1. データベースコネクタの設定

	コネクタに名前を付け、役割を説明する
	接続タイプを選択する
	接続を選択する
	高度な設定
	トランザクションを設定する
	再接続のパラメータを設定する

- ☒ _____ サーバとの時間差を指定する
- ☒ _____ スケジュールのポイントを設定する
- ☒ _____ キャッシュを設定する
- ☒ _____ ピボットドキュメントタイプの使用

接続条件

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

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

コネクタに名前を付け、役割を説明する

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

接続タイプを選択する

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

接続を選択する

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

高度な設定

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

トランザクションを設定する

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

スケジュールのポイントを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

データベースコネクタの生成用ルール

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

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

これらの句はAQL言語のシンタックスに従う必要があります、詳細については「AQL言語」を参照してください。

データベースコネクタの取り込み用ルール

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

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

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

追加情報

本節では以下の内容について説明します。

- ODBCソースの宣言に関する詳細

ODBCソースの宣言に関する詳細

データベースコネクタを含むシナリオが、Windows 32ビット下のサービスとして起動する場合は、ODBCデータソースは、ユーザデータソース（ユーザDNS）ではなく、システムデータソース（システムDNS）として宣言されていなければなりません。

ODBCデータソースがシステムデータソースであることを確認する（サービスモードの場合）

データソースがシステムデータソースであることを確認するには、

- 1 ODBCアドミニストレータを起動します。
- 2 コネクタに使用されているODBCデータソースが、[システムDNS]タブ内にあるかどうか確認します。

[システムDNS]タブ内にはない場合はデータソースを削除し、新規のODBCソースを作成します。

用例シナリオ - データベースコネクタ

このコネクタ用の用例シナリオはありません。

Eメールコネクタ

オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

Eメールコネクタにより、Eメールメッセージの受信と送信が可能になります。Eメールコネクタ（受信）により受信されるメッセージの場合、メッセージの情報は、Connect-Itで処理（マッピング）された後外部アプリケーション内に統合されます。例：E-mail - ServiceCenterタイプのシナリオでは、メールサーバに受信されたメッセージの情報を使用して、ServiceCenter内にチケットを作成できます。

Eメールコネクタ（送信）により送信されるメッセージの場合、Connect-It内で、外部アプリケーションのデータがEメールのフィールドにマップされます。例：Asset Management - E-mailのシナリオ（finreque.scn）では、[依頼] テーブルの各レコードごとにメッセージが作成され、依頼者の責任者にメッセージが送信されます。

本節では以下の内容について説明します。

- コネクタの互換性
- Eメールコネクタの制約点

Eメールコネクタの互換性

Eメールコネクタは、以下のアプリケーションとのテストに成功しています。

プロトコル	例
Lotus VIM	Lotus Notesバージョン4.6と5 Lotus cc:Mail
Microsoft MAPI	Outlookバージョン98 と2000 Exchange 5.0
POP 3-SMTP	Outlook Express Eudora Internet mail

警告:

Connect-ItがインストールされているコンピュータにLotus NotesとOutlook Expressがインストールされていると、Eメールコネクタは両アプリケーションと共に正常に機能します。

Eメールコネクタの既知の制約点

Eメールコネクタをシナリオ内で使用する前に、以下の内容を熟読してください。

サポートされているコードページ

Eメールコネクタにサポートされているコードページは以下の通りです。

- Connect-ItのANSIバージョンでは、

- Latin-1言語（英語、フランス語、など）と互換性のあるオペレーティングシステム：ISO-8859-1、ASCII 7
- Latin-2言語（ポーランド語などの中央ヨーロッパ言語）と互換性のあるオペレーティングシステム：ISO-8859-2、ASCII 7
- Connect-ItのUNICODEバージョン：ISO-8859-1、ISO-8859-2、ASCII 7、SHIFT-JS、EUC-JP、UTF-8、UNICODE

その他の制約点

Eメールコネクタの様々な制約点は以下の通りです。

- Eメールの「プライオリティ」フラグを処理できるのはSMTPサーバのみです。
- コネクタはBase64とQuoted-Printableのエンコードのみをサポートします。
- UnicodeサポートはSMTPとMAPIサーバとのみ使用可能です。
- Quoted-PrintableでエンコードされたEメールが適切に処理されていても、エラーメッセージが表示されます。

警告:






Connect-It クライアントコンピュータの設定：Eメールコネクタが適切に機能するためには、Connect-It がインストールされているコンピュータに、Microsoft Internet Explorerバージョン5.xxがインストールされていなければなりません。

Eメール（受信）コネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 16.2. Eメール（受信）コネクタの設定

	コネクタに名前を付け、役割を説明する
	接続用パラメータを定義する
	処理後のアクションを定義する
	メッセージの内容を処理する
	ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

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

接続用パラメータを定義する

このページでは、Eメール（受信）コネクタのメッセージプロトコルを選択します。

このページに表示されるフィールドの数は、[メールエンジンの種類]フィールドで選択されたプロトコルの種類により変化します。Eメール（受信）コネクタで利用できるプロトコルは以下の通りです。

- Lotus VIM (Vendor Independent Messaging)
- POP3 (Post Office Protocol)
- Microsoft MAPI (Messaging Application Program Interface)

Lotus VIM

VIMプロトコルを選択する場合は、2つのフィールドに入力します。

- [ログイン]
コンピュータにインストールされているVIMメッセージシステムへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

POP3

POP3プロトコルを選択する場合は、4つのフィールドに入力します。

- [POP3サーバ]
POP3サーバの名前を指定します。
- [接続ポート]
サーバに使用されているコンピュータの接続ポートを指定します。このフィールドのデフォルト値は「110」です。
- [ログイン]
POP3サーバへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

Eメール処理のオプション

POP3プロトコルを使用する場合、以下のオプションがあります。

- [Eメールを取得した後サーバから削除しない]
- [Eメールを取得した後サーバから削除する]
このオプションを選択すると、Connect-Itにメッセージが取り出された後、メッセージは削除されます。これにより、処理済みのメッセージが受信トレイ内にたまるのを回避できます。
- [DOS/Windowsテキスト形式を使用する。CR + LF (復帰+改行) を使用する。]
外部アプリケーションが、DOSまたはWindows32ビットをオペレーティングシステムとして使用している場合、このオプションを選択しなければなりません。両システムでは、復帰 (Carriage Return - CR) の次に改行 (Line feed - LF) が来ます。ターゲットアプリケーションが、UNIXをオペレーティングシステムとして使用している場合は、このオプションは使用しません。

MAPI

MAPIプロトコルを選択する場合は、2つのフィールドに入力します。

- [プロファイル]
コンピュータにインストールされたMAPIメッセージシステムへのアクセスを許可するログインを入力します。
- [パスワード]
オペレーティングシステム用のユーザパスワードを入力します (例 : Windows NTパスワード)。パスワードは入力時に非表示になり、暗号化されてから保存されます。

処理後のアクションを定義する

このページでは、コネクタが生成したドキュメントの処理後に、コネクタが適用するアクションを指定します。

[メッセージの取得直後にアクションを適用する]

このオプションを選択すると、シナリオの他のコネクタが送信する処理レポートに応じて、アクションが適用されます。

処理レポートの詳細については、「[処理レポート](#)」の章を参照してください。

処理後のアクションの種類

[[メッセージ処理に成功した場合](#)] と [[メッセージ処理に失敗した場合](#)] の枠内で、以下の2つのオプションのいずれかを選択します。

- [サーバのEメールを変更しない]
- [Eメールをサーバから削除する]

メッセージの内容を処理する

Eメールコネクタでは、DTDファイルに基づいて受信するEメールの本文を解析できます。このファイルは、Eメール本文の構成をXML要素で説明するものです。

DTDファイルを使ってメッセージの内容を処理するには、

- 1 [Eメール本文の内容を処理する] オプションを選択します。
- 2 [DTDファイル] フィールドに、Eメール本文の処理を可能にするDTDファイルのパスを指定します。

ピボットドキュメントタイプの使用





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

Eメール（送信）コネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 16.3. Eメール（送信）コネクタの設定

	コネクタに名前を付け、役割を説明する
	接続を設定する
	送信の頻度を定義する
	ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

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

接続を設定する

[コネクタの設定] ウィザードの2ページ目では、Eメール（送信）コネクタのメッセージプロトコルを選択します。

このページに表示されるフィールドの数は、[メールエンジンの種類]フィールドで選択されたプロトコルの種類により変化します。Eメール(送信)コネクタで使用できるプロトコルは以下の通りです。

- Lotus VIM (Vendor Independent Messaging)
- Microsoft SMTP (Simple Mail Transfer Protocol)
- MAPI (Messaging Application Program Interface)

Lotus VIM

VIMプロトコルを選択する場合は、2つのフィールドに入力します。

- [ログイン]
コンピュータにインストールされているVIMメッセージシステムへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。パスワードは入力時に非表示になり、暗号化されてから保存されます。

SMTP

SMTPプロトコルを選択する場合は、4つのフィールドに入力します。

- [SMTPサーバ]
SMTPサーバの名前を指定します。
- [接続ポート]
SMTPサーバに使用されている接続ポートを指定します。このフィールドのデフォルト値は「25」です。
- [表示名]
メッセージの受信時に、メッセージの[送信者]フィールドに表示される名前を指定します。
- [返信アドレス]
メッセージの受信者が、返信する際に使用できるEメールアドレス(例えば Reply@peregrine.comのようなアドレス)を入力します。

MAPI

MAPIプロトコルを選択する場合は、2つのフィールドに入力します。

- [ログイン]
コンピュータにインストールされているMAPIメッセージシステムへのアクセスを許可するログインを入力します。
- [パスワード]

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

送信の頻度を定義する

[コネクタの設定] ウィザードの3ページ目では、一定数のメッセージがメッセージサーバに送信される毎に、送信を一時停止するよう指定できます。これにより、メッセージが継続的にサーバをふさぐことを回避できます。

このページには2つのフィールドがあります。

- [メッセージ数]

2回の一時停止時間の間にサーバに送信されるメッセージの数を指定します。例えば、メッセージ数として「100」を入力し「30s」の一時停止時間を入力すると、100のメッセージがコネクタに送信されると30秒停止し、また100メッセージが送信される、というようになります。

 注意:

[一時停止時間] フィールドと [メッセージ数] フィールドの適切な値は、メッセージシステムの性能に左右されます。

- [一時停止時間]

一時停止時間を入力する際には「s」を秒、「m」を分、「h」を時間用に使います。例えば、「100」のメッセージ用に「30s」の一時停止時間を入力すると、コネクタは100メッセージをメッセージサーバに送信してから30秒停止し、また100メッセージを送信する、というようになります。

ピボットドキュメントタイプの使用

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

Eメール（受信）コネクタの生成用ドキュメントタイプ

Eメール（受信）コネクタはEメールを受信します。受け取った情報を基に、コネクタは使用可能なドキュメントタイプInMailMessageを発行します。このドキュメントタイプはEメールの様々なコンポーネントを含んでおり、別のコネクタの取り込み用ドキュメントタイプにマップされます。

Eメール（受信）コネクタの生成用ドキュメントタイプ InMailMessage

Eメール（受信）コネクタの生成用ドキュメントタイプのコンポーネントは次の表の通りです。

表 16.4. Eメール（受信）コネクタの生成用ドキュメントタイプ

生成用ドキュメントタイプの構成要素	要素
InMailMessageルートノード	メッセージ本文（【Body】フィールド） メッセージの日付（【Date】フィールド） メッセージのプライオリティ（優先順位）（【Priority】フィールド） メッセージの件名（【Subject】フィールド）
From構造体	メッセージ送信者のEメールアドレス（【Address】フィールド） メッセージ送信者の名前（【Name】フィールド） メッセージ送信者のタイプ（【Type】フィールド）
MailInfo構造体	【UniquelD】フィールドを含みます。このフィールドは、他のコネクタが送信する処理レポートによって使用されます。 これにより、コネクタが生成したメッセージを一意的な方法で識別できるようになります。
Attachmentコレクション	添付ファイルの内容は3つのフィールドに分かれます。 <ul style="list-style-type: none"> 添付ファイルのデータを含む【BlobFromMail】フィールド ファイル名を含む【FileFromMail】フィールド 添付ファイルのタイプを指定する【MimeType】フィールド

生成用ドキュメントタイプの構成要素	要素
Bccコレクション	メッセージのBcc受信者のEメールアドレス ([Address] フィールド) メッセージのBcc受信者の名前 ([Name] フィールド) Bccのタイプ ([Type] フィールド)
Ccコレクション	メッセージのCc受信者のEメールアドレス ([Address] フィールド) メッセージのCc受信者の名前 ([Name] フィールド) Ccのタイプ ([Type] フィールド)
Toコレクション	メッセージ受信者のEメールアドレス ([Address] フィールド) メッセージ受信者の名前 ([Name] フィー ルド) メッセージ受信者への送信のタイプ ([Type] フィールド)

Eメール (受信) コネクタに生成されるメッセージ本文の処理

DTDファイルをコネクタの設定中に指定すると (本節内の「Eメール (受信) コネクタの設定」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 16.1. DTDにより拡張されたドキュメントタイプの例

Eメール (受信) - Asset Managementシナリオ (「newemplo.scn」ファイル) では、DTD「newemplo.dtd」が、InMailMessageドキュメント内に新規の構造体 (amEmplDept構造体) を追加します。

図 16.1. 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

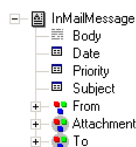
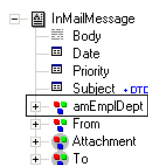


図 16.2. Eメールコネクタ - DTDにより拡張されたInMailMessageドキュメントタイプ



Eメール（送信）コネクタの取り込み用ドキュメントタイプ

Eメール（送信）コネクタはEメールを送信します。コネクタは、Eメールの様々なコンポーネントを含む使用可能なドキュメントタイプOutMailMessage、を発行します。このドキュメントタイプの情報は、別のコネクタの生成用ドキュメントタイプのフィールドにマップされます。（例えばAsset Management - Eメールシナリオ（finreque.scn）の、Asset Managementコネクタ）

Eメール（送信）コネクタの取り込み用ドキュメントタイプ

Eメール（送信）コネクタの取り込み用ドキュメントタイプのコンポーネントは次の表の通りです。

表 16.5. Eメール（送信）コネクタの取り込み用ドキュメントタイプ

取り込み用ドキュメントタイプの構成要素	要素
OutMailMessageルートノード	メッセージ本文（【Body】フィールド） メッセージのプライオリティ（優先順位）（【Priority】フィールド） メッセージの件名（【Subject】フィールド）
Attachmentコレクション	添付ファイルの内容は3つのフィールドに分かれます。 <ul style="list-style-type: none"> 添付ファイルのデータを含む【BlobToSend】フィールド ファイル名を含む【FileToSend】フィールド
Bccコレクション	メッセージのBcc受信者のEメールアドレス（【Address】フィールド） メッセージのBcc受信者の名前（【Name】フィールド） Bccのタイプ（【Type】フィールド）
Ccコレクション	メッセージのCc受信者のEメールアドレス（【Address】フィールド） メッセージのCc受信者の名前（【Name】フィールド） Ccのタイプ（【Type】フィールド）
Toコレクション	メッセージ受信者のEメールアドレス（【Address】フィールド） メッセージ受信者の名前（【Name】フィールド） メッセージ受信者への送信のタイプ（【Type】フィールド）

添付ファイルの処理

添付ファイルは、メッセージに添付されたファイル（画像、ビデオ、音声、実行可能ファイルなど）を指します。Eメールコネクタの使用可能なドキュメントタイプでは、添付ファイルはAttachmentコレクションのフィールドとして表されています。

添付ファイルのマッピング

本節では、受信または送信される添付ファイルのマッピング方法を説明します。

受信される添付ファイルのマッピング

添付ファイルは、Eメールコネクタの生成用ドキュメントタイプ内の3つのフィールドで表されています。

- 添付ファイルのデータを含む [BlobFromMail] フィールド
- ファイル名を含む [FileFromMail] フィールド
- 添付ファイルのタイプを指定する [MimeType] フィールド

受信される添付ファイルをマップするには、[BlobFromMail] フィールドを、ターゲットコネクタのバイナリタイプのフィールドへマップします。

Eメール (受信) - Asset Management シナリオ (newemplo.scn) では、[BlobFromMail] フィールドは、Asset Management コネクタの amEmpDept ドキュメントタイプの [Photo.blobData] フィールドへマップされています。

amEmpDept	
bDepartment	[amEmpDept.bDepartment]
E-Mail	[amEmpDept.E-Mail]
FirstName	[amEmpDept.FirstName]
FullName	[amEmpDept.FullName]
IDNo	[amEmpDept.IDNo]
MiMrs	[amEmpDept.MiMrs]
Name	[amEmpDept.Name]
Phone	[amEmpDept.Phone]
Title	[amEmpDept.Title]
Photo	IF [Attachment(0).MimeType] <> "image"
blobData	[Attachment(0).BlobFromMail]
Name	[Attachment(0).FileFromMail]

送信される添付ファイルのマッピング

送信される添付ファイルは、Eメールコネクタの取り込み用ドキュメントタイプの2つのフィールドで表されています。

- 添付ファイルのデータを含む [BlobToSend] フィールド
- ファイル名を含む [FileToSend] フィールド

送信される添付ファイルのフィールドをマップするには、

- 1 ソースコネクタのバイナリフィールドを、[BlobToSend] フィールドへマップします。
- 2 ソースコネクタのテキストフィールドを、[FileToSend] フィールドにマップします。このフィールドは、添付ファイルの名前を含んでいなければなりません。

Asset Management コネクタの amAsset ドキュメントタイプの [Photo.blobData] フィールドと [Photo.Name] フィールドは、[BlobToSend] フィールドと [FileToSend] フィールドにマップされます。

用例シナリオ - Eメールコネクタ

このコネクタ用の用例シナリオはありません。
このコネクタを使用する用例シナリオは以下の通りです。

- ac36/finconfi.scn
- ac36/finreque.scn
- ac36/hewemplo.scn
- ac40/finconfi.scn
- ac40/finreque.scn
- ac40/hewemplo.scn

LDAPコネクタ

オプションコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

LDAPコネクタは、LDAPプロトコルを使用するサーバから送信されるデータを処理できます。例えば、コネクタはX500電子ディレクトリにアクセスできます。現バージョンのLDAPコネクタでは、LDAPソースから来るデータの読み取りと、LDAPソースへの書き込みが可能になっています。

LDAPコネクタの互換性

LDAPコネクタは、LDAP v2とLDAP v3プロトコルと互換性があります。

LDAPコネクタは、LDAP v3プロトコルの自動記述機能を使用し、ユーザに様々な情報 (オブジェクトクラスの定義、属性タイプの定義、名前付けコンテキスト、サポートされているコントロール、など) を提供します。

LDAPコネクタは、以下のアプリケーションとのテストに成功しています。

- Novell NetWare Directory Service
- OpenLDAP
- Microsoft ActiveDirectory
- iPlanet / Netscape Directory Server

LDAPコネクタ使用上の注意

LDAPコネクタの制約点は、使用するLDAPのリソース（割り当てられたCPUやメモリ、など）に左右されます。

コネクタのテスト中やシナリオの実行中にLDAPリソースの限界を超えると、データ一部または全体の処理に失敗する可能性があります。

LDAPコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 16.6. LDAPコネクタの設定

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

コネクタに名前を付け、役割を説明する

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

LDAPコネクタの設定

[基本設定] ウィザードの2ページ目では、LDAP接続を設定します。

このページのフィールドでは、次の内容を指定できます。

- LDAPサーバ名
- サーバの接続ポート
- ログイン
- パスワード
- サーバタイプ
- コードページ

[サーバ名]

LDAPサーバ名を入力します。サーバ名（例：ldap-server.unil.ch）または数値のIPアドレス（例：207.68.137.42）を入力できます。

[接続ポート]

LDAPサーバの接続ポートを入力します。デフォルト値「389」は多くの場合適切な値です。（この値はプロトコルに推奨されているデフォルト値です。）

[ログイン]

LDAPサーバへのアクセスを許可するログインを入力します。

[パスワード]

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

[サーバタイプを指定してください]

以下の3タイプのLDAPサーバが使用可能です。

- Microsoft Exchange、Novell Directory Service、Netscape Directory Server、など
- Microsoft Active Directory
- OpenLDAP

[サーバのコードページを指定してください]

ドロップダウンリストから、LDAPサーバのコードページを選択します。

[テスト]

このボタンを押すと接続をテストできます。

接続をテストするには、

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

高度な設定

このページでは以下の内容を指定します。

- ページサイズ
デフォルト値は500です。
このオプションの値は、MicrosoftのLDAPサーバでのみ使用されます。
- スキーマのDN (distinguished name、識別名)
稀に、スキーマの識別名がLDAPコネクタによって自動的に取得されないことがあります。
識別子が取得されなかった場合は、Connect-Itログ内に警告が表示されます。
名前を取得するには、ルートDSE (Directory Service Entries) の subschemaSubentryエントリを参照してください。
- 日付形式
日付の形式については、本節内の「追加情報」の節の「LDAP日付形式」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

% [数] [記号] 文字列は、記号の内容を表現するために使用される数字の数を指定します。例：「%4Y」は、年が4つの数字で表現されることを指定します (2003、1997など)。

サーバがミリ秒をサポートする場合、ミリ秒はピリオドで区切られた小数で表示されます。ピリオドの後の数字は、小数点以下の桁数を表しています。
例：「%4Y%2M%2D%2H%2N%2.1SZ」は、ミリ秒の精度が小数点以下1桁であることを表しています。

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

[ルートDSEエントリ]

[ルートDSEエントリ] フィールドでは、LDAPツリー構造内のサーバの現在の日付に対応するエントリを入力できます。

デフォルトのエントリは「CurrentTime」です。

スケジュールのポイントを設定する

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

[最終変更のフィールド]

このフィールドには、LDAPコネクタがスケジュールのポイントとして使用する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。

キャッシュを設定する

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

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定」章の「ピボットドキュメントタイプの使用」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）」章の「取り込み用ルール」の節を参照してください。

コネクタの取り込み用ルールを定義するには、以下の操作を行います。

- [整合性]と[高度な整合性チェック]タブ内で整合性用のパラメータを入力します。
これらの整合性用タブの詳細については、「コネクタのルール(ディレクトティブ)」章の「取り込み用ルール」節内「整合性」を参照してください。
- SCOPE句とSEARCH DN句を作成します。

SCOPE句

SCOPE句は、LDAPディレクトリのツリー構造内で、DN(識別名)により識別されるエントリポイントに対して、クエリのスコープ(適用範囲)を指定します。

- [基本スコープ]
このオプションを選択すると、クエリの対象はエントリポイントに含まれるデータのみになり、このエントリポイント下のサブツリー内は検索されません。
- [1レベルスコープ]
このオプションを選択すると、クエリの対象はエントリポイントに含まれるデータと、このエントリポイントすぐ下の子ノード内のデータになります。
- [サブツリースコープ]
このオプションを選択すると、クエリの対象はエントリポイントに含まれるデータと、このエントリポイント下のすべての子ノード内のデータになります。

SEARCH DN句

SEARCH DN句は、LDAPサーバの接続先のディレクトリのツリー構造内で、エントリポイントのDN(識別名)を指定します。コネクタの起動時に、ディレクトリは複数のエントリポイントを提案します。これらのエントリポイントは、LDAPサーバに表示される名前付けコンテキストに当たります。

名前付けコンテキストは、[SEARCHDN句]下のドロップダウンリスト内に表示されます。このフィールドに直接別のエントリポイントを入力することも可能です。各DNはRDN(相対的識別名)から成っています。以下の表は使用頻度の高いRDNをまとめています。

RDN (相対的識別名)	キー
CommonName	CN
LocalityName	L
StateOrProvinceName	ST
OrganizationName	O
OrganizationalUnitName	OU
CountryName	C
StreetAddress	STREET



注意:

【SEARCHDN句】フィールドに値を入力しておくことをお勧めします。フィールドが空白であると全ディレクトリ内でクエリを実行することになり、LDAPサーバのダウンロード許容量を超過する恐れがあります。

LDAPコネクタの生成用ルール

LDAPコネクタの生成用ルールにより、LDAPコネクタは特定のクエリに従ってLDAPソースからデータを抽出します。

LDAPクエリには、他の生成用ドキュメントタイプのルールタブ内で作成する句とは違う種類の3つの句があります。

- SCOPE句
- SEARCH DN句
- FILTER句

SCOPE句

SCOPE句は、LDAPディレクトリのツリー構造内で、DN (識別名) により識別されるエン트리ポイントに対して、クエリのスコープ (適用範囲) を指定します。

- [基本スコープ]
このオプションを選択すると、クエリの対象はエン트리ポイントに含まれるデータのみになり、このエン트리ポイント下のサブツリー内は検索されません。
- [1レベルスコープ]
このオプションを選択すると、クエリの対象はエン트리ポイントに含まれるデータと、このエン트리ポイントすぐ下の子ノード内のデータになります。
- [サブツリースコープ]

このオプションを選択すると、クエリの対象はエン트리ポイントに含まれるデータと、このエン트리ポイント下のすべての子ノード内のデータになります。

SEARCH DN句

SEARCH DN句は、LDAPサーバの接続先のディレクトリのツリー構造内で、エン트리ポイントのDN（識別名）を指定します。コネクタの起動時に、ディレクトリは複数のエン트리ポイントを提案します。これらのエン트리ポイントは、LDAPサーバに表示される名前付けコンテキストに当たります。

名前付けコンテキストは、[SEARCHDN句]下のドロップダウンリスト内に表示されます。このフィールドに直接別のエン트리ポイントを入力することも可能です。各DNはRDN（相対的識別名）から成っています。以下の表は使用頻度の高いRDNをまとめています。

RDN（相対的識別名）	キー
CommonName	CN
LocalityName	L
StateOrProvinceName	ST
OrganizationName	O
OrganizationalUnitName	OU
CountryName	C
StreetAddress	STREET

注意:

[SEARCHDN句]フィールドに値を入力しておくことをお勧めします。フィールドが空白であると全ディレクトリ内でクエリを実行することになり、LDAPサーバのダウンロード許容量を超過する恐れがあります。

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 16.7. LDAPフィルタ

フィルタ

filter= "(" filtercomp ")"
filtercomp = and / or / not / item
and = "&" filterlist
or = " " filterlist
not = "!" filter
filterlist = 1*filter
item = simple / present /substring / extensible
simple = attr filtertype value
filtertype = equal / approx / greater / less
equal = "="
approx = "-="
greater = ">="
less = "<="
extensible = attr [":dn"] [":" matchingrule] ":@" value / [":dn"] ":" matching rule ":@" value
present = attr "=*"
substring = attr "=" [initial] any [final]
initial = value any = "*" *(value "**")
final = value
attr = AttributeDescription (RFC 2251のセクション4.1.5)
matchingrule = MatchingRuleId (RFC 2251のセクション4.1.9)

フィルタ

value = AttributeValue (RFC 2251のセクション4.1.6)

LDAPコネクタが処理するドキュメントの値の中で、一部の文字を取得する方法は以下の表の通りです。

表 16.8. マップテーブルファイル (ASCII値の文字)

必要な文字	ASCII値
*	0x2a
(0x28
)	0x29
/	0x5c
NULL	0x00

文字は、「/」 (ASCII 0x5c) とエンコードされた文字のASCII値を表す2数値でコード化されなければなりません。16進数である2数値は大文字小文字のどちらでも構いません。例：RDNcnが文字*を含むかどうかを調べるには、以下のフィルタを作成します。

```
"(cn=* /2a*)"
```

フィルタの例

フィルタの目的	作成方法
全クラス内を検索する	(objectclass=*)
名前がAで始まる人をフィルタで抽出する	(& (objectclass=person)(cn=A*))
LDAP内で全タイプの人をフィルタで抽出する	((objectclass=person)(objectclass=organizationalPerson)(objectclass=inetOrgPerson)(objectclass=residentialPerson)(objectclass=newPilotPerson))
名前がAで始まる人を抽出し、そのうちArで始まる人は抽出しない	(& (objectclass=person)(& (cn=A*)(!(cn=Ar*))))
名前がH、Y、Eで始まらない人を抽出する場合、次のようには書けません。	(& (objectclass=person)(!(cn=H*)(cn=Y*)(cn=E*))) "not" ("!") は単項の演算子です。
名前がH、Y、Eで始まらない人を抽出する場合は、次のように書かなければなりません。	(& (objectclass=person)(& (!(cn=H*)(!(cn=Y*)(!(cn=E*)))))

追加情報

本節ではコネクタに関する追加情報を説明します。

LDAPプロトコルに関する情報元

LDAPプロトコルの詳細に関しては、以下のRFC (requests for comments) を参照してください。

- RFC 1274 : The COSINE and Internet X.500 Schema
- RFC 1777 : Lightweight Directory Access Protocol
- RFC 1778 : The String Representation of Standard Attribute Syntaxes
- RFC 1617 : Naming and Structuring Guidelines for X.500 Directory Pilots
- RFC 2253 : Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names
- RFC 2251 : Lightweight Directory Access Protocol (v3)
- RFC 2252 : Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions
- RFC 2254 : The String Representation of LDAP Search Filters
- RFC 2256 : A Summary of the X.500(96) User Schema for use with LDAPv3
- RFC 1823 : The LDAP Application Program Interface
- RFC 1798 : Connection-less Lightweight X.500 Directory Access Protocol
- RFC 2259 : Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2
- RFC 2279 : UTF-8, a transformation format of ISO 10646 RFC 2116 : X.500 Implementations Catalog-96
- RFC 2255 : The LDAP URL Format

コレクションの一意のフィールドを別のアプリケーションのフィールドにマップする

LDAPディレクトリのエント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、「ドキュメントタイプのマッピング」章の「ターゲット要素をソース要素にマップする」の節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 **[マッピングスクリプト]** フィールドに入力します。（注意：Connect-Itのコレクションの構成要素は0から番号が付けられています。第1の構成要素には番号0が、第2の構成要素には番号1がついています。）

例：「Person.cn」（人の通称）一意フィールドのコレクション内にある第1の構成要素を指定するには、[cn(0).cn]をマッピングスクリプト内に入力します。

Basicスクリプトの使用

Basicスクリプトを使用するには、

- 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 **[マッピングスクリプト]** フィールドにマッピングスクリプトを入力します。

スクリプトの前半部分ではコレクション内の構成要素の数を数え、後半部分では属性を連結して1つの文字列にします。シンタックスは以下の通りです。

```
Dim iCollectionCount As Integer
iCollectionCount = PifGetItemCount("cn")
Dim strCollapse As String
Dim iltem As Integer
For iltem = 0 to iCollectionCount - 1
    strCollapse = strCollapse + [cn(iltem)]
Next iltem
RetVal = strCollapse
```

最大のデータを含むLDAP要素を識別する


LDAPコネクタは、以下の情報を含む使用可能なドキュメントタイプを発行します。

- オブジェクトのクラス
- これらのクラスのフィールド
- クラス間の継承

LDAPデータソースの使用において最も重要な情報は以下の通りです。

- データを含むオブジェクトクラスが必要かどうか。
例：レコードを含むテーブル
- クラス間の継承を使うと、高性能なクエリを作成できるか。
例：Top、PersonまたはOrganizationalPersonクラスのうち、どれを使用すべきか。

これらの情報を取得するには、

- 1 シナリオ図部分でLDAPコネクタを選択するかまたは作成します。
- 2 **[ドキュメントタイプ]** タブを選択します。
- 3 ルート要素がTopである生成用ドキュメントタイプを作成します。
全LDAPディレクトリは、全サーバクラスでルート要素の役割を果たすオブジェクトクラスTopを発行します。
- 4  をクリックし、LDAPソースのデータを表示します。
- 5 最多のデータを返すクラスを探し、このクラスをドキュメントタイプの定義内で使用します。

LDAP日付形式

LDAPディレクトリ用の「日付と時刻」（timestamps）型フィールドの形式は、以下のシンタックスに従います。

```
[year][month][day][hour][minute][second]Z
```

例：2003年4月5日22時40分34秒は、文字列「20030405224034Z」で表現されます。

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、日付/時刻フィールド用に別の形式を使用します。例えば、ミリ秒に対応する数字を追加するサーバもあります（この数字はピリオド「.」で区切られます）。例：20030405224034.5Z

用例シナリオ - LDAPコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- `ldap /ac /complete.scn`
- `ldap /ac /simple.scn`

コマンドラインコネクタ

ベースコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

本節ではコマンドラインコネクタの内容を説明します。

- データの取り込み時に、コマンドラインコネクタは、Connect-Itがインストールされているコンピュータ上にコマンドラインを実行します。
- データの生成時に、コマンドラインコネクタは、Connect-Itのインストール先コンピュータで実行されたコマンドライン用に以下の要素を取得します。

- 1 戻り値
- 2 標準出力
- 3 標準入力

コネクタはドキュメントを自動的に生成します。シナリオでは、コマンドラインコネクタがドキュメントを取り込む (他のコネクタがコマンドラインを発信することに相当します) たびに、対応するドキュメントを生成します。

コマンドラインコネクタの互換性

コマンドラインコネクタにより、Connect-Itがインストールされているオペレーティングシステム上にコマンドラインを起動できるようになっています。




UNIXでは、コネクタはコマンドの戻り値のコードのみを取得します。標準出力と標準エラーは取得されません。

コマンドラインコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 16.9. コマンドラインコネクタの設定

	コネクタに名前を付け、役割を説明する
	実行の頻度
	ピボットドキュメントタイプの使用

Connect-Itのコネクタの設定は、ウィザードで実行できます。

コネクタに名前を付け、役割を説明する

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

実行の頻度

このページでは、一定数のドキュメントが処理される毎に一時停止するよう指定できます。これにより、コンピュータやサーバに過度の負担がかかるのを防ぐことができます。

このページには以下のフィールドがあります。

- [ドキュメントの数]

2回の一時停止時間の中にコンピュータが処理するドキュメントの数を指定します。例えば、ドキュメントの数として「100」を入力し「30s」の一時停止時間を入力すると、100ドキュメントがコネクタに処理されると30秒停止し、また100ドキュメントが処理される、というようになります。

 **注意:**

入力できる[一時停止時間]のフィールド値と[ドキュメントの数]のフィールド値は、コンピュータの性能に左右されます。適切な値を入力するにはテストを実行する必要があります。

- [一時停止時間]

一時停止時間を入力する際には「s」を秒、「m」を分、「h」を時間用に使います。例えば、「100」のドキュメント用に「30s」の一時停止時間を入力すると、100ドキュメントがコネクタに処理されると30秒停止し、また100ドキュメントが処理される、というようになります。

- [タイムアウト]

[タイムアウトを過ぎたらプロセスを強制終了する]オプションを選択すると、このフィールドに入力できるようになります。タイムアウトを秒数で入力します。

 **警告:**

UNIXでは、[タイムアウトを過ぎたらプロセスを強制終了する] オプションは使用不可能です。

ピボットドキュメントタイプの使用

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

使用可能なドキュメントタイプ

コマンドラインコネクタには2つの使用可能なドキュメントタイプがあります。

- **Commandドキュメントタイプ** (取り込み用)
このドキュメントタイプは、Connect-Itがインストールされているコンピュータ上に実行されるコマンドラインに相当します。この使用可能なドキュメントタイプにより、コマンドラインコネクタの取り込み用ドキュメントタイプを指定できます。
- **CommandReturnドキュメントタイプ** (生成用)
このドキュメントタイプは、Connect-Itがインストールされているコンピュータ上に実行されるコマンドラインの戻り値に相当します。この使用可能なドキュメントタイプにより、コマンドラインコネクタの生成用ドキュメントタイプを指定できます。

Commandドキュメントタイプ

このドキュメントタイプには3つのフィールドがあります。

- **[Arg] フィールド**
このフィールドはコマンドラインの引数に当たります。
- **[Name] フィールド**
このフィールドは実行可能ファイル (ターゲット) の完全パスに当たります。例: `C:\Program Files\Peregrine\AssetCenter\`
実行可能ファイルがオペレーティングシステムのPath環境変数内に定義されている場合は、実行可能ファイル名を直接指定できます。例: `notepad.exe`
- **[Path] フィールド**
このフィールドはコマンドラインの実行フォルダに当たります。フィールド **[作業フォルダ]** に当たります。

CommandReturn ドキュメントタイプ

このドキュメントタイプには3つのフィールドがあります。

- [ReturnCode] フィールド
このフィールドは、実行されたコマンドの戻り値のコードに当たります。
- [StdErr] フィールド
このフィールドにより、標準のエラー出力に関するデータを取得できます。
- [StdOut] フィールド
このフィールドにより、標準の出力のデータを取得できます。

重要項目:

UNIXでは「StdErr」と「StdOut」フィールドは使用不可能です。

取り込み用ルール (ディレクティブ)

コマンドラインコネクタには、ルール (ディレクティブ) が1つあります。このルールは、コネクタの取り込み用ドキュメントタイプの編集用ウィンドウにある【同期実行】オプションです。同期モード (デフォルトのモード) では、コマンドラインは前のコマンドラインの実行が終了してから次のコマンドラインを実行します。コマンドライン実行から発生する全メッセージ (エラーメッセージも含む) は、コネクタに処理されます。

用例シナリオ - コマンドラインコネクタ

このコネクタ用の用例シナリオはありません。

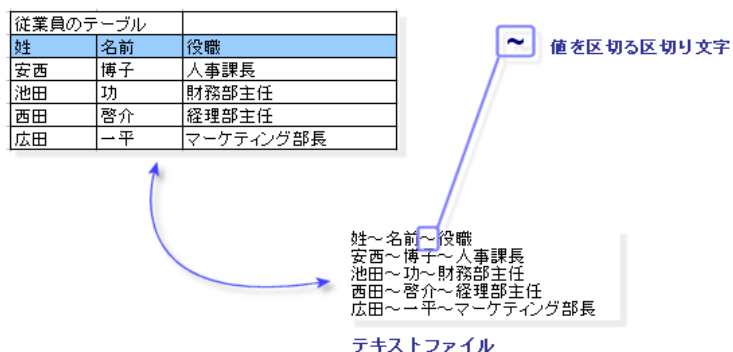
テキストコネクタ

ベースコネクタ

- 生成 (ソース)
- 取り込み (ターゲット)

テキストコネクタは、データベースのレコードを単純なテキスト形式で記録するファイル処理します。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 16.3. テキストコネクタ - テキストファイル内のデータの記録法



テキストコネクタは以下のモードで機能します。

- 読み取りモード
読み取りモードでは、テキストコネクタはテキストファイルのデータを読み取ります。そして、Connect-Itコネクタ（データベースコネクタ、Asset Managementコネクタ、ServiceCenterコネクタなど）が使用できるデータベース内に、テキストファイルのデータを送信します。
- 書き込みモード
書き込みモードでは、テキストコネクタはテキストファイルを作成します。作成されるテキストファイル内のデータは、Connect-Itコネクタ（データベースコネクタ、Asset Managementコネクタ、ServiceCenterコネクタなど）が使用できるデータベースから来るものです。

テキストコネクタでは、以下の操作が可能です。

- Connect-Itがインストールされているコンピュータやネットワーク上にあるテキストファイルを処理する。
これらのファイルはgzipコマンドで圧縮できます。
- FTPサイトやWebサイトにあるテキストファイルを処理する（Webサイトのテキストファイルでは読み取りのみ可能）。

テキストコネクタの既知の制約点

テキストコネクタでの唯一の制約点は、「日付」型と「日付+時刻」型フィールドのフォーマットに関するものです。これらのフィールドを適切に処理するには、フィールドのフォーマットが、Connect-Itがインストールされているコンピュータの地域設定に一致しなければなりません。

「文字列」型のフィールドを使用し、マッピング時にこれをユーザフォーマットに関連付けると、この制約点を回避できるようになります。

ユーザフォーマットの詳細については、「ドキュメントタイプのマッピング」の章の「スクリプトの作成」の節、「ユーザフォーマット」を参照してください。

テキストコネクタの設定（読み取り）

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 16.10. テキストコネクタの設定（読み取り）

コネクタに名前を付け、役割を説明する		
処理モードを選択する		
接続のプロトコルを選択する		
ファイルまたはフォルダを選択する	HTTP Webサイトへの接続	FTPサーバ接続
処理後のアクション	処理後のアクションを定義する	
書式設定ファイル（DSCファイル）を選択する		
ピボットドキュメントタイプの使用		

警告:

ファイルまたはフォルダのパス

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

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、「Connect-Itサービスの作成」の章、「WindowsでConnect-Itサービスを作成する」/「シナリオのパス」を参照してください。

コネクタに名前を付け、役割を説明する

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

処理モードを選択する

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

接続のプロトコルを選択する

以下の3つのオプションがあります。

- [HTTP Webサイト]
- [FTPサーバ]
- [ローカルファイル/ネットワークファイル]

ファイルまたはフォルダを選択する

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

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

ファイルまたはフォルダを選択する

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

- [ファイルを読み取る]
- [フォルダを読み取る]

[ファイルを読み取る]

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

[フォルダを読み取る]

このオプションを選択する場合は、以下の2フィールドに入力します。

- [フォルダ名]
ファイルを含むフォルダのパスを入力します。
- [拡張子]

コネクタが読み取るファイルの拡張子を入力します。例：「txt」

[サブフォルダを読み取る]

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

処理後のアクション

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

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

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

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

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

HTTP Webサイトへの接続

WebサイトにあるXMLファイルの読み取りを選択する場合は、HTTP接続パラメータを入力します。

- [アドレス]
[プロトコル]://[アドレス]:[ポート]/[パス]のようにアドレスを入力します。
[アドレス]の部分のみが必須の場合もあります。
HTTPサーバ用の通常のポート番号は80です。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

[セキュリティで保護された接続 (HTTPSプロトコル)]

このオプションでは、セキュリティで保護された接続 (HTTPS) でサイトに接続するかどうかを指定できます。

🔑 重要項目:

Webサイトアドレスの [プロトコル] の部分がHTTPSプロトコルでない場合でも、[**セキュリティで保護された接続 (HTTPSプロトコル)**] オプションを使用すると、HTTPSプロトコルが強制的に使用されます。

HTTPSサーバ用の通常のポート番号は443です。

HTTPアドレスの例

アドレス	セキュリティで保護された接続 (HTTPS) のオプション	接続モード
https://a.b.c.d/	選択	https port 443
https://a.b.c.d:9000/	選択	https port 9000
http://a.b.c.d:9000/	非選択	https port 9000
http://a.b.c.d:9000/	選択	https port 9000
https://a.b.c.d:9000/	非選択	https port 9000
http://a.b.c.d:443/	非選択	https port 443
https://a.b.c.d:80/	選択	https port 80

[クライアント証明書]

このフィールドでは、コンピュータに既存のHTTPS証明書の中から1つを選択します。

重要: コネクタがドキュメントを読み込んだ後またはドキュメントに書き込んだ後に、このフィールドに指定されているクライアント証明書を変更したり削除したりする場合は、シナリオビルダを一度終了してから再起動して、この変更事項を有効にします。

クライアント証明書のリストを管理する

設定ウィザードにあるクライアント証明書のリストは、Microsoft Managementアプリケーションの「コンソールルート / 証明書 - 現在のユーザー / 個人 / 証明書」フォルダにあるリストと一致します。

Windows 2000でこのフォルダに証明書を追加するには、

- 1 Windowsの [スタート] メニューで [**ファイル名を指定して実行**] を選択します。
- 2 [**名前**] フィールドに「MMC」と入力します。
- 3 [**コンソール / スナップインの追加と削除**] を選択します。
- 4 表示されるウィンドウで [**追加**] をクリックします。

- 5 表示されるウィンドウで **[証明書]** を選択します。
- 6 **[追加]** をクリックします。
- 7 表示されるダイアログボックスで **[ユーザーアカウント]** オプションを選択します。
- 8 **[完了]** をクリックします。
- 9 **[OK]** をクリックします。
- 10 「**証明書 - 現在のユーザー / 個人 / 証明書**」フォルダ内でファイルを追加または削除します。

FTPサーバ

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

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

FTPサーバ接続

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

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

[ファイルを読み取る]

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

[フォルダを読み取る]

このオプションを選択する場合は、以下の2フィールドに入力します。

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

[サブフォルダを読み取る]

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

注意:

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

処理後のアクションを定義する

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

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

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

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

書式設定ファイル (DSCファイル) を選択する

テキストファイルを処理する際、コネクタは書式設定ファイルを使用します。

- 書式設定ファイルが既に存在する場合
[DSCファイル] フィールド内に完全パスを入力します。📁をクリックしてファイルの場所を検索することも可能です。
- 書式設定ファイルが存在しない場合
作成する書式設定ファイルの名前とパスを [DSCファイル] フィールド内に入力し、📁をクリックします。すると、書式設定ファイル作成用のウィザードが起動します。このウィザードについては、本節の「書式設定ファイル (DSCファイル) の作成」を参照してください。

ピボットドキュメントタイプの使用

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

テキストコネクタの設定（書き込み）

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

コネクタの設定ウィザードの各ページは以下の通りです。

テキストコネクタの設定（書き込み）の内容は、選択する接続プロトコルにより変わります。

表 16.11. テキストコネクタの設定（書き込み）

コネクタに名前を付け、役割を説明する		
処理モード		
接続のプロトコルを選択する		
ファイルまたはフォルダを選択する	HTTP Webサイト	FTPサーバ接続
2セッション間の設定		2セッション間の設定
書式設定ファイル（DSCファイル）を選択する		
ピボットドキュメントタイプの使用		



警告:

ファイルまたはフォルダのパス

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

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、「[Connect-Itサービスの作成](#)」の章、「[WindowsでConnect-Itサービスを作成する](#)」 / 「[シナリオのパス](#)」を参照してください。

コネクタに名前を付け、役割を説明する

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

処理モード

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

接続のプロトコルを選択する

以下の3つのオプションがあります。

- [HTTP Webサイト]
- [FTPサーバ]
- [ローカルファイル/ネットワークファイル]

HTTP Webサイト

HTTPサイトにファイルを書き込む場合は、以下の操作を実行します。

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

HTTP Webサイトへの接続

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

- [アドレス]
HTTPサーバへの接続用アドレスを指定します。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

HTTPSプロトコル

セキュリティで保護されたプロトコルを使用するサーバへ接続する場合は、このオプションを選択します。

HTTPSサーバ用の通常のポート番号は443です。

HTTPサーバ用の通常のポート番号は80です。

[クライアント証明書]

このフィールドでは、コンピュータに既存のHTTPS証明書の中から1つを選択します。

重要：コネクタがドキュメントを読み込んだ後またはドキュメントに書き込んだ後に、このフィールドに指定されているクライアント証明書を変更したり削除したりする場合は、シナリオビルダを一度終了してから再起動して、この変更事項を有効にします。

クライアント証明書の一覧を管理する

設定ウィザードにあるクライアント証明書の一覧は、Microsoft Management アプリケーションの「コンソールルート / 証明書 - 現在のユーザー / 個人 / 証明書」フォルダにある一覧と一致します。

Windows 2000でこのフォルダに証明書を追加するには、

- 1 Windowsの [スタート] メニューで [ファイル名を指定して実行] を選択します。
- 2 [名前] フィールドに「MMC」と入力します。
- 3 [コンソール / スナップインの追加と削除] を選択します。
- 4 表示されるウィンドウで [追加] をクリックします。
- 5 表示されるウィンドウで [証明書] を選択します。
- 6 [追加] をクリックします。
- 7 表示されるダイアログボックスで [ユーザーアカウント] オプションを選択します。
- 8 [完了] をクリックします。
- 9 [OK] をクリックします。
- 10 「証明書 - 現在のユーザー / 個人 / 証明書」フォルダ内でファイルを追加または削除します。

書き込み用のコマンド

HTTPサーバへの書き込みには、以下の2タイプのコマンドが使用されます。

- POST
指定されたアドレスに位置するプログラムヘッダを送信します。PUTメソッドと異なり、送信されたデータは処理されなければなりません。
- PUT
データを送信し、指定されたアドレスにデータが保存されるようにします。HTTPサーバ用のコマンドを選択します。

ファイルまたはフォルダを選択する

ローカルまたはネットワーク上にファイルを書き込む場合、以下の操作を実行します。

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

ファイルまたはフォルダを選択する

このページでは、コネクタが書き込むテキストファイルを選択します。以下の2つのオプションがあります。

【1つのファイルにドキュメントを書き込む】

ローカルコンピュータまたはネットワーク上のテキストファイルのパス名を指定します。このファイルが既に存在すると、コネクタの起動時に既存ファイルの内容は上書きされます。

【各ドキュメントごと別々のファイルに書き込む】

ファイルの書き込み先フォルダ名を指定します。ファイル名は、テキストコネクタの取り込み用ルール内に指定した汎用名に一致します（本節の「テキストコネクタ（書き込み）の取り込み用ルール」を参照してください）。

FTPサーバ

FTPサイトにテキストファイルを書き込む場合は、以下の操作を実行します。

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

FTPサーバ接続

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

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

【1つのファイルにドキュメントを書き込む】

テキストコネクタで作成するファイルのパス名を指定します。このファイルが既に存在する場合は、既存ファイルの内容が上書きされます。

【各ドキュメントごと別々のファイルに書き込む】

ファイルの書き込み先フォルダ名を指定します。ファイル名は、テキストコネクタの取り込み用ルール内に指定した汎用名に一致します（本節の「テキストコネクタ（書き込み）の取り込み用ルール」を参照してください）。

2セッション間の設定

このページでは、データ書き込みの2セッション間のドキュメント処理を選択します。1つのファイルにデータを書き込むか、または1つのフォルダの複数のファイルにデータを書き込むかにより、動作オプションは変化します。

[1つのファイルにドキュメントを書き込む]

1つのファイルに書き込む場合は、以下のオプションがあります。

- [同じファイルの後に書き込む]
このオプションを選択すると、コネクタは、前回のセッションが中断した時に開いていたファイル内に、データの書き込みを再開します。
- [前回の全ファイルに上書きする]
コネクタは、前回のセッション時にデータを書き込んでいたファイルを削除します。
- [異なるファイルに番号を付ける]
コネクタはファイルを作成し、前回のファイル名を基に名前を付けます。
例：ファイル.xml、ファイル1.xml、ファイル2.xmlなど。

[各ドキュメントごと別々のファイルに書き込む]

各ドキュメントごと別々のファイルに書き込む場合は、以下のオプションがあります。

- [前回の全ファイルを削除し、番号付けを再開する]
フォルダ内の全ファイルが削除されます（前回のセッションのみでなく、前回までの全セッションのファイルが削除されます）。
コネクタは、書き込む全ファイルへの番号付けを開始します。
- [ファイルの番号付けを続行する]
コネクタはファイルへの書き込みを再開し、既存ファイルを削除せずに番号を付けます。

書式設定ファイル（DSCファイル）を選択する

テキストファイルを処理する際、コネクタは書式設定ファイルを使用します。

- 書式設定ファイルが既に存在する場合
【DSCファイル】フィールド内に完全パスを入力します。📁をクリックしてファイルの場所を検索することも可能です。
- 書式設定ファイルが存在しない場合
作成する書式設定ファイルの名前とパスを【DSCファイル】フィールド内に入力し、📁をクリックします。すると、書式設定ファイル作成用のウィザード

ドが起動します。このウィザードについては、本節の「**書式設定ファイル (DSCファイル) の作成**」を参照してください。

ブラインドモード

[**ブラインドモードを使用する**] オプションを選択すると、テキストコネクタは、書式設定ファイルなしでテキストファイルを作成します。作成されるテキストファイル内の値は、選択した区切り文字で区切られます。

区切り文字には以下のものがあります。

- [**タブ**]
- [**コンマ**]
- [**セミコロン**]
- [**スペース**]
- [**その他**]

[**その他**] フィールドに、区切り文字として使用する記号を入力します。

ブラインドモードでのテキストコネクタの機能

ブラインドモードでテキストコネクタを使用するには、

- [**ブラインドモードを使用する**] オプションを選択し、テキストコネクタを設定します。
- データベース型のコネクタ（例：Asset Managementコネクタ）を選択し、このコネクタ用に生成用ドキュメントタイプを作成します。ターゲットのテキストファイル内に作成する値を含むフィールドを、この生成用ドキュメントタイプ内で選択します。

注意:

ブラインドモードでは、生成用ドキュメントタイプのルートノードのすぐ下にあるフィールド値のみが、ファイルテキストに書き込まれます。生成用ドキュメントタイプのルートノード下に、構造体やコレクションがあると、構造体とコレクション内の値はテキストファイルには書き込まれません。

- シナリオ図内で、**マッピングボックスを bypass せずに**データベース型のコネクタをテキストコネクタに接続します（ [Shift] キーを押したままコネクタ間のリンクを作成すると、マッピングボックスなしの直接リンクが作成されます）。
- データベース型のコネクタを生成モードにします（ [**ツール / 生成する**] コマンド）。

ブラインドモードで生成されるテキストファイル

[1つのファイルにドキュメントを書き込む] オプションが [基本設定] ウィザードで選択されている場合、

- ブラインドモードで作成されるファイル内の各行は、テキストコネクタに取り込まれた各ドキュメントに対応します。
- ブラインドモードで作成されるファイルには、ソースコネクタの生成用ドキュメントタイプと同じ名前が付けられます。

[各ドキュメントごと別々のファイルに書き込む] オプションが [基本設定] ウィザードで選択されている場合、

- テキストコネクタに取り込まれる各ドキュメントの値は、ブラインドモードで作成される各ファイルに書き込まれます。
- ブラインドモードで作成されるファイルには、以下の2要素から成る名前が付けられます。
 - ソースコネクタの生成用ドキュメントタイプの名前 (例: Test)
 - ドキュメントが生成される順番に対応する番号 (例: test_01、test_02、test_03)

書式設定ファイル (DSCファイル) を書き込む

[DSCファイルを書き込む] オプションを使用すると、ブラインドモードで作成されたテキストファイルのデータに基づいて、DSCファイルを作成、保存できます。[ブラインドモードを使用する] が選択されていると、このオプションはデフォルトで選択されます。

既存するファイル名を使用すると、Connect-Itは、自動的に「_01」を追加してから保存します。書式設定ファイルは、ブラインドモード用に選択した区切り文字を使用します。

ピボットドキュメントタイプの使用

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

書式設定ファイル (DSCファイル) の作成

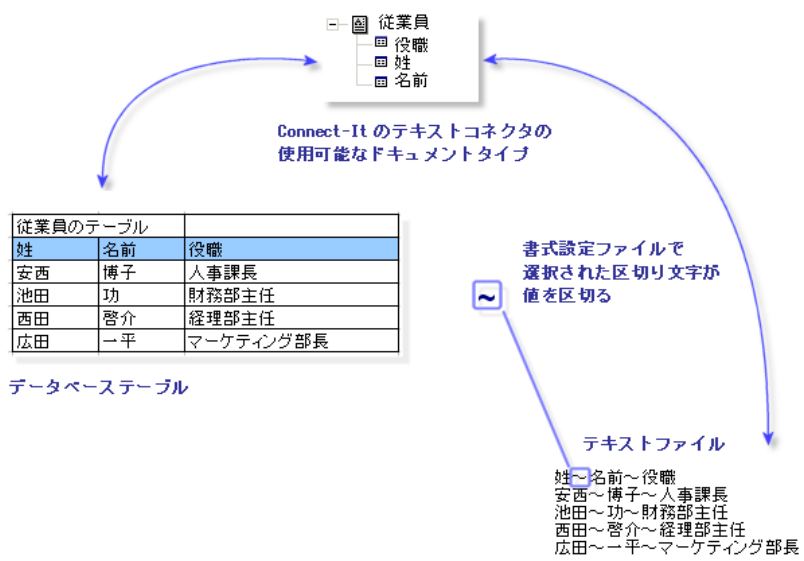
書式設定ファイル (ファイルの拡張子「.dsc」) は、テキストファイル内のデータがどのように構成されているかを説明するファイルです。ファイル内のデータは、データベースのテーブル内のフィールド値に当たります。


テキストコネクタの使用可能なドキュメントタイプは、書式設定ファイル内で作成されたドキュメントタイプです。

例：書式設定ファイルで以下の内容を指定します。

- テキストファイルは1つのテーブル（従業員のテーブル）のデータを含みます。
- 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 16.4. テキストコネクタ - データ処理



書式設定ファイル作成用のウィザードにアクセスするには、【コネクタの設定】ウィザードの【書式設定ファイル（DSCファイル）を選択する】ページで、をクリックします。


書式設定ファイルの設定ウィザードの各ページは以下の通りです。

- ドキュメントタイプ
- プレビューするファイルの選択
- 区切り文字または固定幅
- データ処理のオプション
- 列


ドキュメントタイプ

このページでは、テキストコネクタの使用可能なドキュメントタイプを作成できます。各ドキュメントタイプはデータベース内の1テーブルに当たります。

ドキュメントタイプを作成するには、

-  をクリックします。
- [ドキュメントタイプ] 列内のテキストゾーンをクリックし、作成するドキュメントタイプの名前を入力します。
- [次へ] をクリックして、次のページへ進みます。

ドキュメントタイプを削除するには、

- ドキュメントタイプを選択します。
-  をクリックします。

プレビューするファイルの選択

このページでは、区切り文字で区切られたテキストファイルを指定します。ウィザードは、書式設定ファイルがこのファイルのデータを処理する方法をプレビュー表示します。

処理するプレビューファイル

スペース、タブ、特殊文字などで区切られたデータを含むファイルを指定します。このファイルは、シナリオ内のコネクタが処理するファイルに相当します。

ヒント:

データのプレビュー表示を効率よく実行するには、以下のファイルをプレビューに使用するようお勧めします。

- シナリオの実行時にコネクタが処理するファイルのローカルコピー
- 小さなサイズのファイル（サイズに特に制限はありません）

プレビューする行の数

ウィザードの次のページの [データのプレビュー] で表示する行の数を指定します。

区切り文字または固定幅

このページでは、テキストファイル内のデータを区切る方法を指定します。以下の2つのオプションがあります。

- [固定幅]

このオプションを選択すると、ウィザードの【列】ページの【幅】フィールドで、テキストファイルの各列用に固定幅を指定できます。

列を作成するには、プレビューゾーンを直接クリックします。

列の幅を調整するには、列をクリックしドラッグして移動させます。

列を削除するには、列をクリックしてプレビューゾーン外へドラッグします。

- 【区切り文字】

このオプションを選択すると、テキストファイル内のデータ値用に区切り文字を指定できます。以下の区切り文字が使用可能です。

- 【タブ】
- 【コンマ】
- 【セミコロン】
- 【スペース】
- 【区切り文字として使用する文字のリスト】

区切り文字として使用する文字を指定します。このリスト内で文字を区切るための句読点記号はありません。

データ処理のオプション

このページでは、区切られたテキストファイルのデータを処理する際のオプションを指定できます。

ここでは、セミコロンで区切られた次のテキストを使って、ウィザードの処理オプションを説明します。

```
名前;メーカー;ID
Inspiron;Dell;Comp111
"Inspiron";Dell;Comp112
'Inspiron';Dell;Comp113
"Inspiron" ;Dell ;Comp114
//Inspiron";Dell;Comp115
Inspiron /;Dell;Comp /,116
```


第1行目から列のタイトルをインポートする

このオプションを選択すると、区切られたテキストファイルの1行目の値は、列のタイトルとして使用されます。

	名前	メーカー	ID	列4
1	Inspiron	Dell	Comp111	Comp\
2	"Inspiron"	Dell	Comp112	
3	'Inspiron'	Dell	Comp113	
4	"Inspiron"	Dell	Comp114	
5	\\"Inspiron\"	Dell	Comp115	
6	Inspiron\		Dell	

引用符（クォーテーションマーク）

このフィールドには、テキスト内で使用されている引用符の文字を指定します。

- 一重引用符（'）
- 二重引用符（"）
- 区切り文字として選択した文字以外の全記号

二重引用符を指定すると、次の結果が得られます。

	名前	メーカー	ID	列4
1	Inspiron	Dell	Comp111	Comp\
2	Inspiron	Dell	Comp112	
3	'Inspiron'	Dell	Comp113	
4	Inspiron	Dell	Comp114	
5	\\"Inspiron\"	Dell	Comp115	
6	Inspiron\		Dell	

[値の前後の引用符を保持する] オプションを選択すると、上記で選択した引用符はプレビュー枠内に表示されます。

コメント行の開始

区切られたテキストファイル内のコメントに、印をつけるための文字列を指定します。デフォルト値は//です。

プレビュー枠内でコメントは反転表示されます。

この例で値「//」を入力すると、以下の結果が得られます。

	名前	メーカー	ID	列4
1	Inspiron	Dell	Comp111	
2	"Inspiron"	Dell	Comp112	
3	'Inspiron'	Dell	Comp113	
4	"Inspiron"	Dell	Comp114	
5	//Inspiro	n",Dell;Co	mp115	
6	Inspiron\		Dell	

エスケープ文字

エスケープ文字を入力します。


この例で値「/」を入力すると、以下の結果が得られます。

	名前	メーカー	ID
1	Inspiron	Dell	Comp111
2	"Inspiron"	Dell	Comp112
3	'Inspiron'	Dell	Comp113
4	"Inspiron"	Dell	Comp114
5	//Inspiro	n",Dell;Co	mp115
6	Inspiron;	Dell	Comp;116

列

テキストファイル内の各列は、データベースのテーブル内のフィールドに当たります。つまり、テキストファイル内の各列は、各書式設定ファイル用に作成されるドキュメントタイプのフィールドに当たります。

このページでは、

- テキストファイルの各列を作成し、列に名前を付けます。
列を作成するには、
 -  をクリックします。
 - **[名前]** 列内のテキストゾーンをクリックし、作成する列の名前を入力します。
- 作成する各列用のデータ値のタイプを指定します。

データ値タイプを入力するには、[タイプ]列をクリックし、ドロップダウンリストから1項目を選択します。使用可能なタイプは、[テキスト]、[数値]、[日付]、[日付と時刻]と[通貨]です。

- [区切り文字または固定幅]ページで[固定幅]オプションが選択されている場合は、各列の固定幅を指定します。

固定幅値を入力するには、[幅]列をクリックし必要な文字数を入力します。

列を削除するには、

- 列を選択します。
- **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。

テキストコネクタ（書き込み）の取り込み用ルール

テキストコネクタの取り込み用ドキュメントタイプに適用されるルール（ディレクティブ）を指定できます。

- [汎用名]
このフィールドのデフォルト値は、テキストコネクタの取り込み用ドキュメントタイプの名前です。

- [ファイルの拡張子]
拡張子はピリオドなしで入力します。例えば「txt」が正しい入力方法で、「.txt」は間違いです。デフォルトでは「txt」が入力されています。

テキストコネクタの設定時に、[各ドキュメントごと別々のファイルに書き込む]を選択した場合のみ、これらの2フィールドを使用します。

詳細については、「テキストコネクタの設定（書き込み）」の節の「ファイルまたはフォルダを選択する」を参照してください。

シナリオの起動時に書き込まれるファイルの名前は、汎用名、ファイル作成の順番に一致する番号（_01、_02、_03など）と、指定されたフィルの拡張子から構成されます。

例：「employee」と「txt」を入力すると、作成されるファイルの名前は、employee_01.txt、employee_02.txt、employee_03.txtなどとなります。

 **警告:**

テキストコネクタに複数の取り込み用ドキュメントタイプがある場合、同じ汎用名を使用してはなりません。同じ汎用名が使用されていると、先に作成されたファイルが、後に作成されたファイルに削除され上書きされます。

図 16.5. テキストコネクタ - 取り込み用ルール

汎用名
amAssetDst
ファイルの拡張子
txt

用例シナリオ - テキストコネクタ

このコネクタ用の用例シナリオはありません。

XMLコネクタ

オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

XMLコネクタでは、以下の操作が可能です。

- Connect-Itがインストールされているコンピュータやネットワーク上にあるXMLファイル进行处理する。
- 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:**

圧縮ファイル（gzip）も処理可能です。

XMLコネクタの既知の制約点

XMLコネクタに処理されるドキュメントタイプでは、以下のフィールドタイプはサポートされていません。

- Any
- PCDATA (Blob)
- CDATA
- Namespaces

XMLコネクタの設定（読み取り）

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 16.12. XMLコネクタの設定（読み取り）

コネクタに名前を付け、役割を説明する		
処理モードを選択する		
接続のプロトコルを選択する		
[ローカルファイル / ネットワークファイル]	HTTP Webサイトへの接続	FTPサーバ接続
処理後のアクションを定義する	処理後のアクションを定義する	
DTDを選択する		
ピボットドキュメントタイプの使用		

警告:**ファイルまたはフォルダのパス**

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

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、「[Connect-Itサービスの作成](#)」の章、「[WindowsでConnect-Itサービスを作成する](#)」 / 「[シナリオのパス](#)」を参照してください。

コネクタに名前を付け、役割を説明する

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

処理モードを選択する

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

接続のプロトコルを選択する

以下の3つのオプションがあります。

- [HTTP Webサイト]
- [FTPサーバ]
- [ローカルファイル / ネットワークファイル]

[ローカルファイル / ネットワークファイル]

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

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

ファイルまたはフォルダを選択する

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

- [ファイルを読み取る]

- **[フォルダを読み取る]**



警告:

ファイルまたはフォルダのパス

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

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、「[Connect-Itサービスの作成](#)」の章、「[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。

[ファイルを読み取る]

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

[フォルダを読み取る]

このオプションを選択する場合は、以下の2フィールドに入力します。

- **[フォルダ名]**

ファイルを含むフォルダのパスを入力します。

- **[拡張子]**

コネクタが読み取るファイルの拡張子を入力します。例：「xml」

[サブフォルダを読み取る]

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

処理後のアクションを定義する

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

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

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

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

HTTP Webサイトへの接続

WebサイトにあるXMLファイルの読み取りを選択する場合は、HTTP接続パラメータを入力します。

- [アドレス]
[プロトコル]:/[アドレス]:[ポート]/[パス]のようにアドレスを入力します。
[アドレス] の部分のみが必須の場合もあります。
HTTPサーバ用の通常のポート番号は80です。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

[セキュリティで保護された接続 (HTTPSプロトコル)]

このオプションでは、セキュリティで保護された接続 (HTTPS) でサイトに接続するかどうかを指定できます。

重要項目:

Webサイトアドレスの [プロトコル] の部分がHTTPSプロトコルでない場合でも、[セキュリティで保護された接続 (HTTPSプロトコル)] オプションを使用すると、HTTPSプロトコルが強制的に使用されます。

HTTPSサーバ用の通常のポート番号は443です。

HTTPアドレスの例

アドレス	セキュリティで保護された 接続 (HTTPS) のオプション	接続モード
https://a.b.c.d/	選択	https port 443
https://a.b.c.d:9000/	選択	https port 9000
http://a.b.c.d:9000/	非選択	https port 9000
http://a.b.c.d:9000/	選択	https port 9000
https://a.b.c.d:9000/	非選択	https port 9000
http://a.b.c.d:443/	非選択	https port 443
https://a.b.c.d:80/	選択	https port 80

[クライアント証明書]

このフィールドでは、コンピュータに既存のHTTPS証明書の中から1つを選択します。

重要：コネクタがドキュメントを読み込んだ後またはドキュメントに書き込んだ後に、このフィールドに指定されているクライアント証明書を変更したり削除したりする場合は、シナリオビルダを一度終了してから再起動して、この変更事項を有効にします。

クライアント証明書のリストを管理する

設定ウィザードにあるクライアント証明書のリストは、Microsoft Managementアプリケーションの「コンソールルート / 証明書 - 現在のユーザー / 個人 / 証明書」フォルダにあるリストと一致します。

Windows 2000でこのフォルダに証明書を追加するには、

- 1 Windowsの [スタート] メニューで [ファイル名を指定して実行] を選択します。
- 2 [名前] フィールドに「MMC」と入力します。
- 3 [コンソール / スナップインの追加と削除] を選択します。
- 4 表示されるウィンドウで [追加] をクリックします。
- 5 表示されるウィンドウで [証明書] を選択します。
- 6 [追加] をクリックします。
- 7 表示されるダイアログボックスで [ユーザーアカウント] オプションを選択します。
- 8 [完了] をクリックします。
- 9 [OK] をクリックします。
- 10 「証明書 - 現在のユーザー / 個人 / 証明書」フォルダ内でファイルを追加または削除します。

FTPプロトコル

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

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

FTPサーバ接続

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

- [サーバ]

- FTPサーバ名を指定します。
 - [ログイン]
サイトへのアクセスを許可するログインを入力します。
 - [パスワード]
ログイン用のパスワードを入力します。
- [**ファイルを読み取る**] または [**フォルダを読み取る**] オプションを選択します。
[**フォルダを読み取る**] を選択する場合以下の内容も指定します。
- FTPサイト上のフォルダのパス
 - ファイルの拡張子
フィールドのデフォルト値は、「xml」です。
- [**サブフォルダを読み取る**] オプションを選択することもできます。

処理後のアクションを定義する

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

- [フォルダに保存する]
 - [フォルダから削除する]
 - [次のフォルダに移す]
- 3番目のオプションを選択する場合は、処理されたファイルの移動先フォルダのパスを入力します。

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

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

[**ファイルを読み取る**]

このオプションを選択する場合は、FTPサイト上にあるXMLファイルのパスを指定します。

[**フォルダを読み取る**]

このオプションを選択する場合は、以下の2フィールドに入力します。

- [**フォルダ名**]
ファイルを含むフォルダのパスを入力します。
- [**拡張子**]

コネクタが読み取るファイルの拡張子を入力します。

[サブフォルダを読み取る]

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

DTDを選択する

XMLファイル进行处理するために、コネクタはDTD（文書型定義）を使用する必要があります。このページでは、使用するDTDを [DTD] フィールドで指定できます。

注意:

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

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

ファイルの場所

以下の3つのオプションがあります。

- [ローカル/ネットワーク]

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

- [FTP]

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

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

注意：FTPサイトに学習モードでDTDを書き込むことはできません。

 **注意:**

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

- **[HTTP]**
[サーバ名]、**[ログイン]**、**[パスワード]** と **[パス]** フィールドにHTTPパラメータを入力します。
[パス] フィールドでは、DTDのサーバ上の場所を指定します。例えば、DTDがサーバ「http://mycompany.com」上の「myfolder/dtd」フォルダ内にある場合、**[サーバ名]** フィールドに「http://mycompany.com」を入力し、**[パス]** フィールドに「/myfolder/DTD/mydtd.dtd」を入力します。
注意 : HTTPサイトに学習モードでDTDを書き込むことはできません。
-

 **注意:**

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

[第1レベルの各コレクションに使用可能なドキュメントタイプを発行する]

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

例 :

例

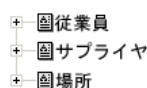
あるDTDのルート要素が「会社」で、第1レベルに3つのコレクション「従業員」、「サプライヤ」と「場所」があるとします。上記のオプションが選択されていないと、コネクタが発行する使用可能なドキュメントタイプのルート要素は「会社」になります。

注意:

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



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



ルート要素を指定する

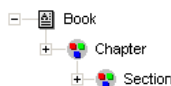
DTDは複数の要素から構成されており、これらの要素は更に別の要素を含んでいます。例えば、文書編集用のDTDでは、複数の「Section」要素が複数の「Chapter」要素に含まれており、更に「Chapter」要素は1つの要素「Book」に含まれています。他の要素に含まれることがない要素は、ルート要素と呼ばれます。DTDは複数のルート要素から構成されることもあれば、DTDに全くルート要素が存在しないこともあります。

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

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

このオプションを選択すると、XMLコネクタはDTD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。



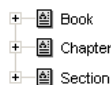
注意:

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

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

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

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



注意:

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

学習モード

学習モードを使用すると、XMLコネクタは **[ファイルまたはフォルダを選択する]** ページで指定したXMLファイルに基づいてDTDを作成します。

注意:

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

学習モードを使用するには、


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

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

XMLドキュメントとDTDドキュメントの矛盾

原則として、XMLコネクタに読み取られるファイルのXML要素は、DTD内に定義された要素に一致するはずですが、新規の要素が現れることもあります。要素の少ないXMLファイルを学習モードで処理した時に得られたDTDを使用する際に、このようなXMLとDTDの矛盾が頻繁に起こります。

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

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

ピボットドキュメントタイプの使用

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

XMLコネクタの設定（書き込み）

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 16.13. XMLコネクタの設定（書き込み）

コネクタに名前を付け、役割を説明する	
処理モードを選択する	
接続のプロトコルを選択する	
ファイルまたはフォルダを選択する	HTTP Webサイトへの接続 FTPサーバ接続
2セッション間の設定	2セッション間の設定
DTDを選択する	
ピボットドキュメントタイプの使用	

警告:

ファイルまたはフォルダのパス

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

パスにネットワークドライブを指定すると、サービスに関連付けられたシナリオは正常に機能しなくなります。詳細については、「Connect-Itサービスの作成」の章、「WindowsでConnect-Itサービスを作成する」/「シナリオのパス」を参照してください。

コネクタに名前を付け、役割を説明する

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

処理モードを選択する

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

接続のプロトコルを選択する

以下のオプションがあります。

- [FTPサーバ]
- [ローカルファイル/ネットワークファイル]
- [HTTP Webサイト]

HTTP Webサイトへの接続

このページでは、ファイルの書き込み先HTTPサーバへ接続するために、以下の3つのフィールドに入力します。

- [アドレス]
これは必須フィールドです。[プロトコル]:/[アドレス]:[ポート]/[パス]のようにアドレスを入力します。
- [ログイン]
サイトへのアクセスを許可するログインを入力します。
- [パスワード]
ログイン用のパスワードを入力します。

セキュリティで保護された接続 (HTTPSプロトコル)

このオプションの使用方法については、本節の「 [セキュリティで保護された接続 (HTTPSプロトコル)] 」を参照してください。

書き込み用のコマンド

HTTPサーバへの書き込みには、以下の2タイプのコマンドが使用されます。

- POST
指定されたアドレスに位置するプログラムヘータを送信します。PUTメソッドと異なり、送信されたデータは処理されなければなりません。例：CGI書式のデータの送信
- PUT
データを送信し、指定されたアドレスにデータが保存されるようにします。例：Webサイトのページの更新

HTTPサーバ用のコマンドを選択します。

ローカルファイル/ネットワークファイル

ローカルまたはネットワーク上にファイルを書き込む場合は、ファイルまたはフォルダを選択します。

ファイルまたはフォルダを選択する

このページでは、コネクタが書き込むXMLファイルを選択します。以下の2つのオプションがあります。

- [1つのファイルにドキュメントを書き込む]
ローカルコンピュータまたはネットワーク上のXMLファイルのパス名を指定します。このファイルが既に存在すると、コネクタの起動時に既存ファイルの内容は上書きされます。
- [各ドキュメントごと別々のファイルに書き込む]
ファイルの書き込み先フォルダ名を指定します。ファイル名は、XMLコネクタの取り込み用ルール内に指定した汎用名に一致します（本節の「XMLコネクタ（書き込み）の取り込み用ルール-FTPとファイル」を参照してください）。

FTPプロトコル

FTPサイトにテキストファイルを書き込む場合は、以下の操作を実行します。

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

FTPサーバ接続

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

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

[1つのファイルにドキュメントを書き込む]

XMLコネクタで作成するファイルのパス名を指定します。このファイルが既に存在する場合は、既存ファイルの内容が上書きされます。

[各ドキュメントごと別々のファイルに書き込む]

ファイルの書き込み先フォルダ名を指定します。ファイル名は、XMLコネクタの取り込み用ルール内に指定した汎用名に一致します（本節の「XMLコネクタ（書き込み）」の取り込み用ルール-FTPとファイル」を参照してください）。

2セッション間の設定

このページでは、データ書き込みの2セッション間のドキュメント処理を選択します。1つのファイルにデータを書き込むか、または1つのフォルダの複数のファイルにデータを書き込むかにより、動作オプションは変化します。

[1つのファイルにドキュメントを書き込む]

1つのファイルに書き込む場合は、以下のオプションがあります。

- [同じファイルの後に書き込む]
このオプションを選択すると、コネクタは、前回のセッションが中断した時に開いていたファイル内に、データの書き込みを再開します。
- [前回の全ファイルに上書きする]
コネクタは、前回のセッション時にデータを書き込んでいたファイルを削除します。
- [異なるファイルに番号を付ける]
コネクタはファイルを作成し、前回のファイル名を基に名前を付けます。
例：ファイル.xml、ファイル1.xml、ファイル2.xmlなど。

[各ドキュメントごと別々のファイルに書き込む]

各ドキュメントごと別々のファイルに書き込む場合は、以下のオプションがあります。

- [前回の全ファイルを削除し、番号付けを再開する]
フォルダ内の全ファイルが削除されます（前回のセッションのみでなく、前回までの全セッションのファイルが削除されます）。
コネクタは、書き込む全ファイルへの番号付けを開始します。
- [ファイルの番号付けを続行する]
コネクタはファイルへの書き込みを再開し、既存ファイルを削除せずに番号を付けます。

DTDを選択する

XMLファイルを作成するために、コネクタはDTD（文書型定義）を使用する必要があります。このページでは、使用するDTDを【DTD】フィールドで指定で

きます。コネクタXMLの使用可能なドキュメントタイプは、このDTDに一致します。

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

ファイルの場所

以下の3つのオプションがあります。

- [ローカル/ネットワーク]

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

- [FTP]

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

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

 **注意:**

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

- [HTTP]

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

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

 **注意:**

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

[XMLファイルで取得したDTDを参照しない]

このオプションはデフォルトのオプションです。

[取得したDTDをXMLファイル内に書き込む]

このオプションを選択すると、XMLコネクタに作成されたXMLファイルは、ヘッダーにDTDを含みます。

[取得したDTDを外部ファイルに書き込み、XMLファイル内で参照する]

このオプションを選択する場合は、作成されるXMLドキュメントのDTD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は、ローカルまたはネットワークファイル上に保存されなければなりません。

! 警告:

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

DTD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が、使用されることとなります。

新規のDTDが外部ファイルに書き込まれるようにするには、シナリオを停止し、再起動しなければなりません。

ルート要素を指定する

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

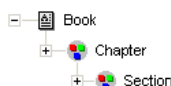
ルート要素（つまり、XMLコネクタに発行される使用可能なドキュメントタイプに当たります）を指定するためには、以下の2つのオプションがあります。

- [DTD内に見つかった各ルート要素に使用可能なドキュメントタイプを発行する(推奨オプション)]
- [ユーザの選択する各ルート要素に使用可能なドキュメントタイプを発行する(コンマで値を区切る)]

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

このオプションを選択すると、XMLコネクタはDTD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。



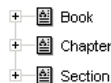
注意:

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

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

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

例：文書編集用のDTDで、ユーザが、3つのルート要素「Book」、「Chapter」と「Section」を選択するとします。この場合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。

ピボットドキュメントタイプの使用

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

XMLコネクタ（書き込み）の取り込み用ルール - FTPとファイル

XMLコネクタの取り込み用ドキュメントタイプに適用されるルール（ディレクトィブ）を指定できます。

- [汎用名]
このフィールドのデフォルト値は、XMLコネクタの取り込み用ドキュメントタイプの名前です。
- [ファイルの拡張子]
拡張子はピリオドなしで入力します。例えば「xml」が正しい入力方法で、「.xml」は間違いです。デフォルトでは「xml」が入力されています。

XMLコネクタの設定時に、[各ドキュメントごと別々のファイルに書き込む] を選択した場合のみ、これらの2フィールドを使用します。

（本節の「XMLコネクタの設定（書き込み）」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 16.6. XMLコネクタ - の取り込み用ルール

汎用名	amAssetDst
ファイルの拡張子	xml

追加情報 - XMLコネクタ

本節ではコネクタに関する追加情報を説明します。

1つまたは複数のファイルに書き込む

書き込みモードのXMLコネクタは、1つまたは複数のファイルにデータを書き込むことができます。

1つのファイルに書き込む

このモードでは、コネクタに取り込まれた各ドキュメントを1つのファイル内に記録します。

例えば、1つのXMLファイルのみが、AssetCenter内の資産の棚卸情報を含むようにする場合に、このモードを使用できます。以下のDTDを使用できます。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT AssetInventory (amAsset*)>
<!ELEMENT amAsset (AssetTag?)>
<!ELEMENT AssetTag (#PCDATA)>
```

このDTDではルート要素（AssetInventory）には子要素amAssetコレクションがあります。このコレクションの構成要素は、AssetCenterのデータベース内の資産のレコードに対応します。各資産は資産タグ（AssetTag）で識別されます。

この場合、XMLコネクタは以下のような内容を含むファイルを書き込みます。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<AssetInventory>
  <amAsset>
    <AssetTag>DEMO-NTR1</AssetTag>
  </amAsset>
  <amAsset>
    <AssetTag>DEMO-OFF1</AssetTag>
  </amAsset>
  <amAsset>
```



```
<AssetTag>DEMO-SFT2</AssetTag>
</amAsset>
</AssetInventory>
```



ソースとターゲットのマッピング

amAssetドキュメントタイプとXMLコネクタが使用するDTD間のマッピングには、以下のような特徴があります。

- 1 資産タグのフィールド (AssetTag) は、DTDのAssetTagフィールドに直接マップされます。
- 2 ルート要素 (AssetInventory) の子ノード (AssetTag) は、ターゲットドキュメントタイプ内に表示されます。

以下のDTDを使用します。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT AssetInventory (amAsset*, amSoftware)>
<!ELEMENT amAsset (AssetTag?)>
<!ELEMENT AssetTag (#PCDATA)>
<!ELEMENT amSoftware (Version?)>
<!ELEMENT Version (#PCDATA)>
<!ATTLIST AssetInventory ID CDATA #REQUIRED>
```

amSoftware要素とIDフィールドは、XMLコネクタの取り込み用ドキュメントタイプ内では無視され、表示されません。これはこの2つがDTD内で宣言されていないためです。

- 3 **[第1レベルの各コレクションに使用可能なドキュメントタイプを発行する]** オプションを選択してはなりません。

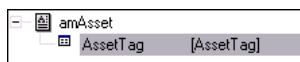
複数のファイルに書き込む

このモードでは、XMLコネクタが取り込む各ドキュメントを別々のファイル内に保存できます。

上記の例で説明すると、XMLコネクタは、資産の各レコードを別々のファイルに書き込むこととなります。

以下のDTDを使用できます。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT amAsset (AssetTag?)>
<!ELEMENT AssetTag (#PCDATA)>
```



この場合、XMLコネクタは複数のファイルに書き込みます。ファイルの内容は以下の例の通りです。

ファイル1

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<amAsset>
  <AssetTag>DEMO-OFF1</AssetTag>
</amAsset>
```

ファイル2

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<amAsset>
  <AssetTag>DEMO-NTR1</AssetTag>
</amAsset>
```

ソースとターゲットのマッピング

amAssetドキュメントタイプとXMLコネクタが使用するDTD間のマッピングには、以下のような特徴があります。

- 1 DTDで宣言されている全要素（コレクション、構造体、フィールド）がターゲット内に表示されます。
- 2 **[第1レベルの各コレクションに使用可能なドキュメントタイプを発行する]** オプションを選択してはなりません。

用例シナリオ - XMLコネクタ

このコネクタを使用する用例シナリオの一部は以下の通りです。

- acscups /dataprop /01-SCcompany2ACamCompany.scn
- acscups /dataprop /02-SClocation2ACamLocation.scn
- acscups /dataprop /03-SCdept2ACamEmplDept.scn
- acscups /dataprop /04-SCcontacts2ACamEmplDept.scn
- acscups /dataprop /05-SCdevtype2ACamNature.scn
- acscups /dataprop /06-SCocmlcat2ACamNature.scn

- `acscups /dataprop /07-SCmodel2ACamModel.scn`
- `acscups /dataprop /08-SCdevtype2ACamModel.scn`
- `acscups /dataprop /09-SCdevice2AACamPortfolioAndamComputer.scn`
- `acscups /dataprop /10-SClaptop2ACamPortfolioAndamComputer.scn`
- `acscups /dataprop /11-SCworkstation2ACamPortfolioAndamComputer.scn`
- `acscups /dataprop /12-SCpc2ACamAndamComputer.scn`
- `acscups /dataprop /13-SCserver2ACamAndamComputer.scn`
- `acscups /dataprop /14-SCphone2ACamPortfolioAndamPhone.scn`

17 | インベントリコネクタ

インベントリコネクタは、IT資産のインベントリを実行するアプリケーションのデータを処理するコネクタです。

Intel LANDeskコネクタ

オプションコネクタ

- 生成 (ソース)

Intel LANDeskインベントリコネクタでは、Intel LANDeskのインベントリツールで取得されたデータベースを処理できます。

2種類のIntel LANDeskコネクタがあります。

- Intel LANDesk 6.3と6.4コネクタ
- Intel LANDesk 6.5コネクタ

使用しているIntel LANDeskのバージョンに応じたコネクタを選択してください。このインベントリツールはIT資産の全情報を取得します。

Intel LANDeskコネクタの互換性

Intel LANDeskコネクタは、以下のドライバとODBC Administratorバージョン3.520.4403.2とのテストに成功しています。

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739
- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301
- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500

Intel LANDeskコネクタは、バージョン3.xxの全ODBCドライバと互換性があるものと推察されます。

既知の制約点

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

例：





ODBCAdministratorバージョン3.0と、Access 4.00ODBCドライバを使用していると、Intel LANDeskコネクタは機能しません。

Intel LANDeskコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 17.1. Intel LANDeskコネクタの設定

	コネクタに名前を付け、役割を説明する
	接続タイプを選択する
	接続を選択する
	高度な設定

- 再接続のパラメータを設定する
- サーバとの時間差を指定する
- スケジュールのポイントを設定する
- キャッシュを設定する
- ピボットドキュメントタイプの使用

前提条件

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

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

コネクタに名前を付け、役割を説明する

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

接続タイプを選択する

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

接続を選択する

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

高度な設定

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

スケジュールのポイントを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

Intel LANDeskコネクタの生成用ルール

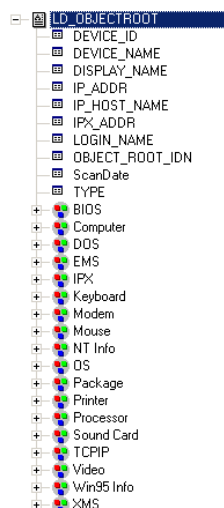
コネクタの生成用ルールの入力方法については、「コネクタのルール（ディレクティブ）」章の「生成用ルール」の節を参照してください。

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

使用可能なドキュメントタイプ

Intel LANDeskコネクタの使用可能なドキュメントタイプは、LD_OBJECTROOTのみです。このドキュメントタイプは、Intel LANDeskがコンピュータに実行するスキャンに相当します。

図 17.1. Intel LANDeskコネクタ - COMPUTER_SYSTEMドキュメントタイプ



用例シナリオ - PDA Intel LANDeskコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- ild/ild2am.scn
- ild/ild65am.scn

Tivoli Inventoryコネクタ (バージョン3.1と3.6)

オプションコネクタ

- 生成 (ソース)

Tivoli Inventory Managementコネクタでは、Tivoli Inventory アプリケーション (バージョン3.1と3.6) で取得されたデータベースを処理できます。このアプリケーションはTivoli Inventory Managementスイートに含まれています。

Tivoli InventoryはIT資産の全情報を取得します。Tivoli Inventory Management - Asset Management既製シナリオ (tim2am.scn) では、これらの情報を基に、Asset Managementアプリケーションの資産のテーブルにレコードを作成します。

Tivoli Inventory Managementコネクタの互換性

Tivoli Inventory Managementコネクタは、以下のドライバとODBC Administratorバージョン3.520.4403.2とのテストに成功しています。

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739
- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301
- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500

Tivoli Inventory Managementは、バージョン3.xxの全てのODBCドライバと互換性があるものと推察されます。

Tivoli Inventory Managementコネクタの既知の制約点

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

例：

ODBCAdministratorバージョン3.0と、Access 4.00ODBCドライバを使用していると、Tivoli Inventory Managementコネクタは機能しません。

ポインタのステータスの必須の初期化

Tivoli Inventory Managementコネクタを含むシナリオでは、スケジュール編集用のウィンドウで、ポインタに値「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 Inventory Management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 Inventory Managementはドキュメントを生成できません。

この値を割り当てるには、

- 1 **[シナリオ/スケジュール]** を選択します。

- 2 Tivoli Inventory Managementコネクタに生成されるドキュメントを、予め作成したスケジューラに関連付けます。
- 3 生成されるドキュメントごとに、[ポインタのステータス]の列をダブルクリックします。
- 4 表示されるテキスト編集用枠に「0」を入力します。



Tivoli Inventory Managementコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 17.2. Tivoli Inventory Managementコネクタの設定

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

前提条件

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

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

Connect-Itのコネクタの設定は、ウィザードで実行できます。

コネクタに名前を付け、役割を説明する

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

接続タイプを選択する

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

接続を選択する

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

高度な設定

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

スケジュールのポイントを設定する

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

キャッシュを設定する

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

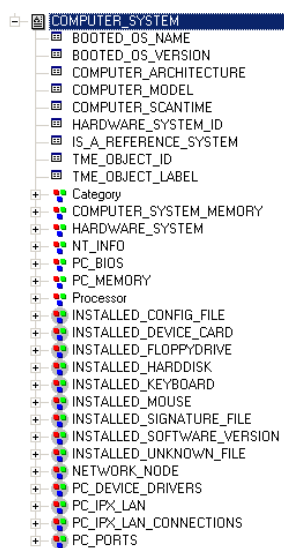
ピボットドキュメントタイプの使用

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

使用可能なドキュメントタイプ

Tivoli Inventory Managementコネクタの使用可能なドキュメントタイプは、COMPUTER_SYSTEMのみです。このドキュメントタイプは、Tivoli Inventoryがコンピュータ上に実行するスキャンに相当します。このドキュメントタイプの要素は、スキャンの設定により変化します。例えばデータベースのテーブルを表すコレクションは、スキャンに応じて表示されたり消滅したりします。

図 17.2. Tivoli Inventory Managementコネクタ - COMPUTER_SYSTEMドキュメントタイプ



警告:

このドキュメントタイプの要素は、Tivoli Inventory Managementデータベースの構造に応じて変化します。このため、COMPUTER_SYSTEMドキュメントタイプでコレクションとして表されるテーブルが、データベースに存在しないこともあります。シナリオのマッピングが存在しないテーブルのフィールドを含んでいる場合、このシナリオは適切に実行されない可能性があります。この問題を解決するには、不在テーブルのフィールドを以下の場所で削除します。

- 1 シナリオのマッピング内
- 2 Tivoli Inventory Managementコネクタの生成用ドキュメントタイプ内

Tivoli Inventory Managementコネクタの生成用ルール

Tivoli Inventory Managementコネクタの生成用ルールにより、コネクタは、ODBCデータソース内のデータにフィルタと並べ替えを直接実行できます。

コネクタが生成するドキュメント用のデータにフィルタと並べ替えを実行するには、SQLクエリ用の句に類似する以下の2種類の句を作成する必要があります。

- WHERE句
- ORDER BY句

図 17.3. Tivoli Inventory Managementコネクタ - 生成用ルール

WHERE句

WHERE句ではODBCデータベースのレコードにフィルタを適用します。例えば、2001年1月23日かそれ以降に実行されたスキャンを取得するとします。この場合、[COMPUTER_SCANTIME]>='2001-01-23'と入力します。

ORDER BY句

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

[配列の取り出しを使用しない]

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

SQLシンタックス用の規則

ドキュメントの生成用ルールに使用するSQLシンタックスは、Tivoli Inventory データベース作成に使用したデータベースエンジンに使われているシンタックスと同一でなければなりません。

SQLシンタックスに加えて、Connect-Itには以下の規則があります。

- フィールド名は角括弧で囲む
例：[フィールド1]
- 文字列はシングルクォーテーションマークで囲む
例：'コンピュータ'
- 数値にはクォーテーションマークをつけない
- 日付は、コンピュータの地域の設定オプションで指定したフォーマット従って入力し、シングルクォーテーションマークで囲む

用例シナリオ - Tivoli Inventory Managementコネクタ (バージョン3.1と3.6)

このコネクタを使用する用例シナリオは以下の通りです。

- tim/tim2am

Tivoli Inventoryコネクタ (バージョン4.0)

オプションコネクタ

- 生成 (ソース)

Tivoli Inventory Management 4.0コネクタでは、Tivoli Inventory 4.0アプリケーションで取得されたデータベースを処理できます。このアプリケーションはTivoli Inventory Managementスイートに含まれています。

Tivoli InventoryはIT資産の全情報を取得します。Tivoli Inventory Management 4.0 - Asset Management既製シナリオ (tim4am.scn) では、これらの情報を基に、Asset Managementアプリケーションの資産のテーブルにレコードを作成します。

Tivoli Inventory Management 4.0コネクタの互換性

Tivoli Inventory Management 4.0コネクタは、以下のドライバとODBC Administratorバージョン3.520.4403.2とのテストに成功しています。

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739
- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301
- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500
- IBM DB2 ODBCバージョン7.01.00.00

Tivoli Inventory Management 4.0コネクタは、バージョン3.xxのODBCドライバ全てと互換性があるものと推察されます。

DB2

コネクタは、IBM DB2 ODBC 7.01.00.00ドライバともテストされており、正常に作動することが確認されています。

既知の制約点

Tivoli Inventory Management 4.0コネクタを適切に使用するためには、ODBCドライバとODBCアドミニストレータのDLLバージョンが、同一である必要があります。

例：

ODBC Administratorバージョン3.0と、Microsoft Access 4.00 ODBCドライバを使用していると、コネクタは機能しません。

Tivoli Inventory Management 4.0コネクタの設定

設定を行うと、コネクタはTivoli Inventory Management 4.0データソースに接続できるようになります。

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 17.3. Tivoli Inventory Management 4.0コネクタの設定

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

前提条件

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

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

コネクタに名前を付け、役割を説明する

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

接続タイプを選択する

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

接続を選択する

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

高度な設定

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

スケジュールのポイントを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

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

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

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

用例シナリオ - Tivoli Inventory コネクタ (バージョン 4.0)

このコネクタを使用する用例シナリオは以下の通りです。

- tim/tim4am
- tim/tim4sc4

SMS 1.x コネクタと SMS 2.x コネクタ

オプションコネクタ

- 生成 (ソース)

SMS 1.x と SMS 2.x (Microsoft Systems Management Service) ゲートウェイコネクタでは、SMS データソースのデータを処理できます。

SMS コネクタでは、SMS データベースのデータの読み取りのみが可能です。SMS データベースへのデータの書き込みは不可能です。

SMS コネクタの互換性

SMS 1.x コネクタは、Microsoft Systems Management Service のバージョン 1.x 全てと互換性があります。

SMS 2.x コネクタは、Microsoft Systems Management Service のバージョン 2.0、ビルド 1239 とテストされています。このコネクタは Systems Management Service のバージョン 2.x 全てと互換性があると推察されます。

SMS コネクタは Microsoft SQL Server ドライバ、バージョン 3.70.08.20 とのテストの結果、正常に作動することが確認されています。このドライバの他のバージョンとも正常に機能するものと推察されます。

コネクタ使用上の重要な情報

ゲートウェイコネクタに関する注意事項

ゲートウェイコネクタを含むシナリオの稼動前に、データのインポート先データベースのコピー（例えばAssetCenterデータベースのコピー）を使ってこのシナリオを予めテストしておくことを強くお勧めします。

このテストでは以下の操作を実行できます。

- マッピングの識別キーの有効性を確認する
無効な識別キーを使用すると、ターゲットデータベースに重複する値が作成される可能性があります。重複する値を手動で削除する操作は非常に困難です。
- マップテーブル、文字列テーブル、グローバル関数を適応させる
Connect-Itまたは用例シナリオ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、用例シナリオが適切に機能しない、または用例シナリオを開くことができない可能性があります。

- SMSコネクタの接続先データベースがカスタマイズされている場合（テーブルの追加や削除、またはフィールド名の変更など）。この場合シナリオでマップされている要素が、コネクタの生成用ドキュメントタイプに含まれていない可能性があります。
- SMSコネクタの接続先データベースがSMSデータベースではない場合。シナリオでマップされている要素は、コネクタの生成用ドキュメントタイプには含まれません。

使用するSMS 2.xデータベースの構造がSMS 1.xデータベースの構造と一致する場合

Microsoftのツールを用いてSMS 1.xデータベースからSMS 2.xデータベースへ移行した場合、データベースの内容はSMS 1.xデータベースと同じです。この場合、以下のことに注意する必要があります。

- SMS 2.xコネクタの使用可能なドキュメントタイプはデータベース構造には一致しません。

- SMS 2.x - Asset Management用例シナリオのマッピングは使用不可能になります。旧SMS 1.xデータベースから継承されたテーブルを考慮しつつ、データベースをカスタマイズする必要があります。

SMSコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 17.4. SMS 1.xコネクタとSMS 2.xコネクタの設定

1	コネクタに名前を付け、役割を説明する
2	接続タイプを選択する
3	接続を選択する
4	高度な設定
5	再接続のパラメータを設定する
6	サーバとの時間差を指定する
7	スケジュールのポイントを設定する
8	キャッシュを設定する
9	ピボットドキュメントタイプの使用

前提条件

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

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

コネクタに名前を付け、役割を説明する

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

接続タイプを選択する

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

接続を選択する

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

高度な設定

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

スケジュールのポインタを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

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

これらの句はAQL言語のシンタックスに従う必要があります、詳細については「AQL言語」を参照してください。

SMSコネクタ用のWHERE句の例

ID (dwMachineID) が4575であるコンピュータに当たるレコードを取得するには、WHERE句のフィールドに以下の関数を入力します。

```
[dwMachineID]=4575
```

図 17.4. SMSコネクタ - 生成用ルールの例



用例シナリオ - SMS 1.xとSMS 2.xコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- sms \sms1am.scn
- sms \sms1amlt.scn
- sms \sms1am.scn
- sms \sms2amlt.scn

Unicenter AMOコネクタ

オプションコネクタ

• 生成 (ソース)

Unicenter AMOゲートウェイコネクタでは、Unicenter AMOスイートのインベントリツールで取得されたデータベースを処理できます。

このインベントリツールはIT資産の全情報を取得します。Unicenter AMO - Asset Management用例シナリオ (amo2am.scn) では、これらの情報を基に、Asset Managementアプリケーションの資産のテーブルにレコードを作成できます。

Unicenter AMOコネクタの互換性

Unicenter AMOコネクタは、以下のドライバとODBC Administratorバージョン 3.520.4403.2とのテストに成功しています。

- Accessバージョン4.00.4403.02
- Excelバージョン4.00.4403.02
- Sybase 11.9バージョン3.11.0001
- Sybase SQL Anywhereバージョン5.05.052739
- Textバージョン4.00.4403.02
- Microsoft SQL Serverバージョン3.70.08.20
- Oracle 7バージョン2.05.0301
- Oracle 8.0バージョン8.00.0500
- Oracle 8iバージョン8.01.0500

Unicenter AMOコネクタは、バージョン3.xxのODBCドライバ全てと互換性があるものと推察されます。

Unicenter AMOコネクタの既知の制約点

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

例：

ODBCAdministratorバージョン3.0と、MicrosoftAccess 4.00ODBCドライバを使用していると、Unicenter AMOコネクタは機能しません。

Unicenter AMOコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 17.5. Unicenter AMOコネクタの設定

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

前提条件

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

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

コネクタに名前を付け、役割を説明する

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

接続タイプを選択する

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

接続を選択する

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

高度な設定

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

再接続のパラメータを設定する

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

サーバとの時間差を指定する

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

スケジュールのポイントを設定する

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

キャッシュを設定する

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

ピボットドキュメントタイプの使用

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

Unicenter AMOコネクタの生成用ルール

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

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

使用可能なドキュメントタイプ

Unicenter AMOコネクタには以下の使用可能なドキュメントタイプがあります。

- Application
- Computer
- Engine
- Groupe
- User
- DirectAccessTables

このドキュメントタイプは、コネクタの他の使用可能なドキュメントタイプ全てのコレクションを含みます。このコレクションを、ターゲットコネクタの取り込み用ドキュメントタイプに直接マップすることも可能です。

コレクション - ドキュメントのマッピングについては、「ドキュメントタイプのマッピング」の章の「ターゲット要素をソース要素にマップする」の節、「コレクションからドキュメントへのマッピング」を参照してください。

用例シナリオ - Unicenter AMOコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- `amo/amo2am.scn`

18 | Javaコネクタ

Javaコネクタにより、Java言語を使用するアプリケーションと通信できるようになります。

Connect-ItにはJava開発キット（JDK）が付属しており、独自のJavaコネクタをユーザが作成できるようになっています。

この開発キットの詳細については、「[Connect-ItのJava開発キット（JDK）](#)」を参照してください。

SCAutoリスニングコネクタ

オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

SCAuto リスニングコネクタは、ServiceCenterのSCAutomateサーバのエミュレータ（SCAuto）として機能します。コネクタは、イベントを受信しクライアントの要求を処理することが出来ます。

イベント受信モードのSCAutoリスニングコネクタ

イベント受信モードのSCAutoリスニングコネクタはイベントを受信します。このイベントによりコネクタは、手動またはスケジュールモードでコネクタが起動する度にドキュメントを生成できるようになります。このコネクタは通信路（ソケット）を使用します。



クライアント要求の処理モードのSCAutoリスニングコネクタ

クライアント要求の処理モードでのコネクタの使用は、Connect-Itサーバに送信する要求を処理するリソースコンポーネントの使用に依存しています。



SCAutoリスニングコネクタの制約点

コネクタは以下のアダプタをサポートしません。

- 日本語版のSCAuto OpenView Network Node Manager adapterバージョン3.x
- SCAuto Email adapter
- 以下のコマンドを使用する全アダプタ
 - QUERY2
 - INSERTBLOB
 - CREATEBLOB

- DESCRIBE OBJECT
- SELECT OBJECTS
- END SELECT OBJECTS
- GETNEXT OBJECT
- STORE OBJECT




SCAutoリスニングコネクタがSCAutoのSDKの「DELETE」コマンドをサポートする場合でも、このコマンドはイベントを削除することは出来ません。イベントを削除するには、処理レポートを使用する必要があります。処理レポートの詳細については、「[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](#)」章の「[コネクタを設定する](#)」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 18.1. SCAutoリスニングコネクタの設定

	コネクタに名前を付け、役割を説明する
	コネクタの導入
	ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

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

コネクタの導入

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

リッスンポート

クライアントの要求を受信するためにコネクタが使用するリッスンポートを指定します。

これは必須フィールドです。


設定ファイル

SCAutoリスニングコネクタの使用可能なドキュメントタイプを記述する、XML設定ファイルのパスを指定します。

これは必須フィールドです。

これらのドキュメントタイプはSCAutoのイベントのタイプに対応します。Connect-Itのシナリオでは、ServiceCenterのイベントタイプから設定ファイルを取得することが可能です。

設定ファイルを取得するには、

- 1 シナリオビルダを起動します。
- 2 「scacfg.scn」シナリオを開きます。
このファイルは、[Connect-Itインストール先フォルダ]scenario\scauto\scacfgにあります。
- 3 シナリオのServiceCenterコネクタを設定します。
- 4  をクリックしてシナリオのコネクタを開きます。
- 5 全スケジューラを起動します（「Ctrl + F5」キー）。
シナリオは、[Connect-Itインストール先フォルダ]scenario\scauto\scacfgにXMLファイルを作成します。

要求用のリソース

Connect-Itサーバに送信する要求を処理するコンポーネントの名前を入力します。

ピボットドキュメントタイプの使用

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

追加情報

SCAutoリスニングコネクタ設定ファイルを変更する

SCAutoリスニングコネクタのXML設定ファイルを変更すると、サポートできるイベントのタイプを増減できます。また、文字列のコレクションをMemoタイプのフィールドとして処理することも可能です。

設定ファイルには2つのセクションがあります。

- InputEventTypes
SCAutoリスニングコネクタが取り込むイベント

取り込まれるイベントは、SCAutoアダプタが生成するイベントに相当しません。

- OutputEventTypes

SCAutoリスニングコネクタが生成するイベント

生成されるイベントは、SCAutoアダプタが要求するイベントに相当します。

イベントタイプは構造体として表現され、構造体の名前はイベントタイプの名前になります。この構造体は、イベントのフィールドを表すサブノードを含みます。

4つのタイプの単一フィールドがサポートされています。

- String (Character)
- Byte (Logical)
- Long (Number)
- TimeStamp (Date)

括弧内のタイプは、ServiceCenterのタイプに相当します。配列 (Array) タイプのフィールドは、単一フィールドのコレクションの形で表されます。これは scacfg.scnシナリオでのデフォルトの表示方法です。Memoタイプの単一フィールドとして表示することも可能ですが、この場合フィールド値は1段落 (パラグラフ) となり、配列内の要素の各値は1つの行で表されます。**注意**：文字列の配列のみにこの表示方法を適用すべきです。この表示方法を文字列以外の配列に適用すると、配列の要素は文字列であると見なされてしまいます。

単一フィールドはインデックス、名前と、タイプで特徴付けられます。

例

```
ATTRIBUTE index="11" name="orig.operator" type="String"/
```

配列フィールドは、インデックス、名前、区切り文字、要素名と、要素タイプで特徴付けられます。要素名は恣意的に付けられるため、フィールド名とは異なる場合があります。

例

```
< COLLECTION index="2" name="comments.2" separator="|">
  < ATTRIBUTE name="comments" type="String"/>
</COLLECTION>
```

この例では、配列フィールドを以下の行で表現することも可能です。

```
< ATTRIBUTE index="2" name="comments.2" type="Memo" separator="|"/>
```

以下の例では3つの要素 (「abc」「def」と「ghi」) の配列は、属性値を付与します。

```
"abc
```

```
def
ghi"
```

属性「separator」がない場合または空の場合は、デフォルトの区切り文字「|」が使用されます。

「lf」と「cr」の2つは特別な区切り文字です。この2つは改行（line feed）と復帰（carriage return）を意味します。この2つの区切り文字以外の場合、区切り文字は^以外の1文字でなければなりません。

用例シナリオ - SCAutoリスニングコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- scauto /scacfg /scacfg.scn
- scauto /sca-sc /sca-sc.scn

このシナリオでは、同じフォルダ内にある設定ファイル「BasicScaCfg.xml」を、シナリオの設定中に使用する必要があります。

XMLリスニングコネクタ

オプションコネクタ

- 生成（ソース）

XMLリスニングコネクタは、要求をコネクタに転送するサーブレットから以下の操作を実行できます。

- コネクタが受信する要求からドキュメントを生成します。
- 受信する要求を処理する別のコネクタと相互作用します。

XMLデータは、サーブレットへ送信されるHTTP要求の内容から抽出されます。

発行モードでのXMLリスニングコネクタ（ドキュメントの生成）

使用可能なドキュメントタイプ発行モードのXMLリスニングコネクタは、イベントを受信します。このイベントによりドキュメントの生成が可能になります。



相互作用モードのXMLリスニングコネクタ

相互作用モードで、コネクタはクライアントが送信するHTTP要求をリスンします。コネクタはこの要求をリソースコネクタへ転送し、このリソースコネクタがドキュメントを生成します。



XMLリスニングコネクタの互換性

コネクタは、Javaサーブレットをサポートする全Webサーバと互換性があります。

XMLリスニングコネクタの制約点

XMLリスニングコネクタの制約点は、使用するサーブレットにより変化します。

XMLリスニングコネクタの設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 18.2. XMLリスニングコネクタの設定

■	コネクタに名前を付け、役割を説明する
■	コネクタの導入
■	ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

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

コネクタの導入

接続

この枠では、XMLリスニングコネクタへの接続パラメータを設定できます。

- ポート**
 XMLリスニングコネクタが使用するリスンポートを入力します。
 このポートを使用すると、Webサーバは、WebサーバとXMLリスニングコネクタ間の通信路（ソケット）を作成できます。
- URL**
 XMLイベントを生成するHTTP要求（POSTまたはGET）を処理するサーブレットのURLを指定します。
- 同時接続の最大数**
 コネクタが処理する同時接続の最大数を指定します。フィールドのデフォルト値は20です。
[この最大数を超過する接続を拒否する] オプションはデフォルトでオフになっています。この場合、接続が最大数に達すると、新規接続はコネクタに処理されるまでキュー内に保持されます。

要求の処理

この枠では要求の処理用のパラメータを入力できます。

- HTTP要求の処理ポリシー
コネクタがHTTP要求を処理する際のモードを指定します。[発行]と[相互作用]の選択肢があります。
- 発行用の設定ファイル
コネクタが発行する使用可能なドキュメントタイプを指定するXMLファイルのパスを指定します。
- リソース
相互作用の要求を処理するコネクタを指定します。このコネクタはドキュメントの生成モードで使用できなければなりません。

ピボットドキュメントタイプの使用

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

XMLリスニングコネクタの生成用ルール

これらのルールは、コネクタがXMLドキュメントの形で処理するHTTP要求内にあります。

XMLリスニングコネクタの使用可能なドキュメントタイプ

発行モードのXMLリスニングコネクタは、設定ファイルで指定されている使用可能なドキュメントタイプを発行します。

用例シナリオ - XMLリスニングコネクタ

このコネクタ用の用例シナリオはありません。

Tivoli Enterprise Consoleコネクタ（送信）

オプションコネクタ

- 取り込み（ターゲット）

送信モードのTivoli Enterprise Consoleコネクタでは、Tivoli Enterprise Console（TEC）ハイイベントを送信できます。

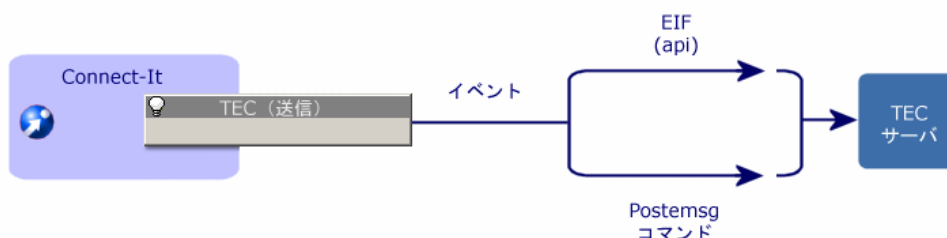
注意:

TECコネクタ（送信）はTECアダプタに相当します。

このコネクタはJCA規格に従って開発されています。

TECコネクタはソースコネクタからくるドキュメントを取り込み、以下のいずれかを使用してドキュメントを送信します。

- TECのEIF API
- postmsgコマンド



前提条件

TECコネクタ（送信）を使用するには、Connect-Itのクラスパス内に「eif.jar」ライブラリを追加する必要があります。

TEC 3.7.1の「eif.jar」ライブラリは、フォルダ[TECのインストール先フォルダ]/bin/generic-unix/TME/TEC内にあります。それより前のバージョンのTECを使用している場合は、Tivoliの顧客サポートまでお問合せください。

TEC adapter（送信）コネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定」章の「ピボットドキュメントタイプの使用」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 18.3. TEC adapterコネクタの設定

コネクタに名前を付け、役割を説明する

- コネクタの導入
- ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

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

コネクタの導入

このページでは、TECサーバへの接続パラメータを設定できます。

サーバアドレス

TECサーバのアドレスを入力します。
これは必須フィールドです。

サーバポート

サーバポートを入力します。
これは必須フィールドです。

設定ファイル

TECイベントクラスを含むXMLファイルを指定します。このファイルのイベントクラスは、BAROCファイルで定義されているTECサーバのクラスに一致しなければなりません。

これは必須フィールドです。

TECデータベースのイベントクラスを含むデフォルトの記述ファイルが、Connect-Itに付属しています。ファイルは、[Connect-Itインストール先フォルダ]/config/tec/teccfg.xmlにあります。

イベントバッファを有効にする

このオプションを選択すると、TECサーバへの接続が切断された場合、サーバに送信されるイベントはバッファファイルに格納されます。

注意:

このオプションを使用しないと、TECサーバへの接続が切断した場合コネクタはドキュメントを拒否します。

バッファファイル

バッファファイルの完全パスを指定します。

[イベントバッファを有効にする] オプションが選択されている場合、このフィールドは必須です。

バッファファイルの最大サイズ (KB)

バッファファイルの最大サイズを指定します。ファイルの最大サイズに達し、サーバへの接続が切断されると、TECコネクタは新規ドキュメントの取り込みを拒否します。

バッファ削除の速度 (イベント/分)

サーバへの接続が回復した際に、バッファ内に格納されるドキュメントをConnect-Itが処理する速度を指定できます。1分ごとのイベントの数はTECサーバの性能に応じて調節されます。

値「0」は、速度制限なしにイベントが削除されることを意味します。

イベントの最大サイズ

このフィールドでは、TECコネクタが送信するイベントの最大サイズを指定します。

サイズは256KBと4096KBの間でなければなりません。

サーバはUTF-8をサポートしない

デフォルトでは、TECサーバへ送信される全イベントは、UTF-8にコード化されます。

サーバがUTF-8をサポートしないは、このオプションを選択します (TECバージョン3.7より前)。

このオプションを選択する場合は、[サーバのコードページ] フィールドにサーバが使用するコードページを入力します。

サーバのコードページ

上記のオプションを選択する場合、サーバが使用するコードページをこのフィールドに入力します。

コマンドラインを使用する

postemsgコマンドを使ってコネクタがTECサーバへイベントを送信する場合、このオプションを選択します。

'postemsg' コマンドのパス

postemsgコマンドの完全パスを指定します。

【コマンドラインを使用する】 オプションが選択されている場合、このフィールドは必須です。

このコマンドのパスは、[TECインストール先フォルダ]/bin/[コンピュータの依存フォルダ]/bin/です。

例：Windows 32ビットのTECサーバでは、コンピュータの依存フォルダは「W32ix86」です。

ピボットドキュメントタイプの使用

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

追加情報

TECコネクタ（送信）にデフォルトで付属しているイベントクラスのファイルを、新規のイベントクラスを使用するために拡張することも可能です。イベントクラスはBAROCファイルで定義されています。本章の「Tivoli Enterprise Console コネクタ（受信）」章の「SCAutoリスニングコネクタ設定ファイルを変更する」の節を参照してください。

付属の用例シナリオ

付属の用例シナリオはありません。

Tivoli Enterprise Consoleコネクタ（受信）

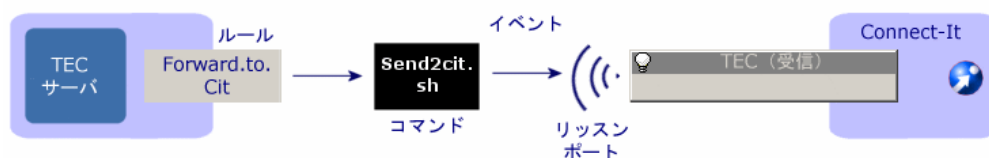
オプションコネクタ

- 生成（ソース）

受信モードのTivoli Enterprise Console（TEC）コネクタは、Tivoli Enterprise Consoleのイベントの受信を可能にするイベントコネクタです。

コネクタの設定内に指定されるリスンポートにより、TECが送信するイベントをリスンできるようになります。

TEC内で定義されるルールにより、TECコネクタへイベントを送信できるようになります。詳細については、本節の「SCA Autoリスニングコネクタ設定ファイルを変更する」を参照してください。



TECコネクタ（受信）の互換性

コネクタは、以下のTECとのテストに成功しています。

- TEC 3.6x
- TEC 3.7x

イベント送信用のTECの設定

TECコネクタがイベントを受信できるようにするには、TEC内でルールを作成する必要があります。

TECのルール作成についてはTECの『ルールビルダ用ガイド』を参照してください。

TECコネクタ（受信）の設定

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

コネクタの設定ウィザードの各ページは以下の通りです。

表 18.4. TECコネクタの設定

📄	コネクタに名前を付け、役割を説明する
📄	コネクタの導入
📄	ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

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

コネクタの導入

このページでは、TEC（受信）コネクタ導入用のパラメータを設定できます。

リッスンポート

TECが送信するイベントを受信するためにコネクタが使用するリッスンポートを指定します。

このリッスンポートは、スクリプト `send2cit.sh` のパラメータ内に指定されているポートと一致します。本節の「イベントクラス「TEST_CLASS」のイベントを Connect-Itへ送信する」を参照してください。

これは必須フィールドです。

設定ファイル

TECイベントクラスを含むXMLファイルを指定します。このファイルのイベントクラスは、BAROCファイルで定義されているTECサーバのクラスに一致しなければなりません。

これは必須フィールドです。

TECデータベースのイベントクラスを含むデフォルトの記述ファイルが、Connect-Itに付属しています。ファイルは、[Connect-Itインストール先フォルダ] `config/tec/teccfg.xml` にあります。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定」章の「ピボットドキュメントタイプの使用」の節を参照してください。

追加情報

teccfg.xml イベントクラスファイルを拡張する

イベントクラスのデフォルトの付属ファイルを、新規イベントクラスを追加するために拡張することも可能です。イベントクラスはBAROCファイル内で定義されています。

BAROCシンタックスを使用するイベントクラスの定義の例は以下の通りです。

```

ENUMERATION EXAMPLE_ENUM
  0 GA
  1 BU
  2 ZO
  3 ME
END

TEC_CLASS:
Example_class ISA Parent_class
DEFINES {
  slot_1 : STRING;
  slot_2 : INTEGER;
  slot_3 : INT32;
  slot_4 : EXAMPLE_ENUM;
  slot_5 : STRING, parse = no;
  slot_6 : INTEGER, default = 0;
  slot_7 : EXAMPLE_ENUM, default = GA;
  slot_9 : LIST_OF STRING, parse = no;
  slot_10 : LIST_OF INTEGER, default = [0,1];
  slot_11 : LIST_OF EXAMPLE_ENUM, default = [GA,BU,ZO,ME], parse = no;

  slot_12 : default = "dummy", parse = no;
};
END

```

BAROCファイルの拡張の例に関する注意

タイプは以下のように対応します。

- STRINGとリストデータはStringになります。
- INT32とINTEGERはIntegerになります。
- REALはDoubleになります。

LIST_OFタイプのフィールドはコレクションとして処理されます。例：LIST_OF STRINGフィールドは、Stringフィールドのコレクションになります。

継承は特別なXMLタグPARENTで表記されます。

クラスが親クラスのフィールドを再定義する場合（デフォルト値、タイプなど）、このフィールドはXML設定ファイル内で完全に再定義されなければなりません。変更されていない親クラスのフィールドは、自動的に新規クラスに追加されます。

TECコネクタのXMLシンタックスに翻訳されると、Example_classの定義は以下のようになります。

```

<STRUCTURE name="Example_class">
  <PARENT name="Parent_class"/>
  <ATTRIBUTE name="slot_1" type="String"/>
  <ATTRIBUTE name="slot_2" type="Integer"/>
  <ATTRIBUTE name="slot_3" type="Integer"/>
  <ATTRIBUTE name="slot_4" type="String"/>
  <ATTRIBUTE name="slot_5" type="String" parse="no"/>
  <ATTRIBUTE name="slot_6" type="Integer" default="0"/>
  <ATTRIBUTE name="slot_7" type="String" default="GA"/>
  <COLLECTION name="slot_9" type="String" parse="no"/>
  <COLLECTION name="slot_10" type="Integer" default="[0,1]"/>
  <COLLECTION name="slot_11" type="String" parse="no" default="[GA,BU,ZO,
ME]"/>
  <ATTRIBUTE name="slot_12" type="String" parse="no" default="dummy"/>
</STRUCTURE>

```

 注意:

「slot_12」タイプおよび「parse」と「default」属性は、再定義されていない場合、親クラスから削除されなければなりません。

TECでイベント送信用ルールを作成する

TECがイベントクラスのイベントをConnect-Itへ送信出来るようにするには、コマンド「send2cit」を発信するルールをTEC内に作成する必要があります。

このコマンドは、TEC内で作成されたイベントをTECコネクタ（受信）へ送信します。

コマンドsend2citは、[Connect-Itのインストール先フォルダ]/datakit/teevent_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「teevent_package.exe」には、以下のファイルとフォルダが含まれています。

- send2cit.sh
- teevent.jar
- teevent_lang.jar
- 「getvalues.exe」ファイルを含むWindowsフォルダ
- 「getvalues」ファイルを含むSolaris、AIXとHP-UXフォルダ

これらの全ファイルと、使用するオペレーティングシステムに対応する「getvalues.exe」または「getvalues」ファイルは、TECサーバで同一のフォルダ内にコピーされなければなりません。

例 18.1. イベントクラス「TEST_CLASS」のイベントをConnect-Itへ送信する

イベントクラス「TEST_CLASS」のイベントを、Connect-Itへ送信するためのルールを作成するには、

- 1 新規ルールベースを作成します。
- 2 このルールベース内に、例えば「Forward」という名前の新規ルールセットを作成します。
- 3 このルールセット内に、例えば「fwd_test_class」という名前のルールを作成します。
- 4 このルールがイベントクラスTEST_CLASSに適用されることを指定します。
- 5 ルールのアクションを編集します。

ルールのアクションの編集

ルールのアクションを編集するには、

- 1 [イベント受信時] にアクションを作成します。
- 2 [コマンドの起動] を選択します。
- 3 コマンド「send2cit.sh」を選択します。
- 4 コマンドの引数を編集します（ [引き数の編集] ）。
 - -p XXX
「XXX」はTECコネクタ（受信）のリッスンポートに相当します。
この引数は**必須**です。
 - -s citserver
「citserver」は、Connect-Itがインストールされているサーバの名前です。
この引数はオプションです。この引数が指定されていないと、TECがインストールされているコンピュータがサーバになります。
 - -l logfile
「logfile」はログファイルの完全パスです。
この引数はオプションです。この引数が指定されていないと、ログファイルは作成されません。
- 6 ルールベース（Forward）を保存します。
- 7 ルールベースをコンパイルします。
- 8 ルールベースを読み込みます。

一度ルールベースを読み込むと、各TEST_CLASSイベントはTECに受信されません。イベントはConnect-Itへ送信されてからTECコネクタ（受信）により処理されます。

TECのルール作成についてはTECの『ルールビルダ用ガイド』を参照してください。

付属の用例シナリオ

付属の用例シナリオはありません。

Webサービスコネクタ

オプションコネクタ

- 生成（ソース）
- 自発的な取り込み（ターゲット）

Webサービスコネクタを使用すると、Webサービスと相互作用することができます。Webサービスは、1つのサービスまたは複数の演算からなるサービスの集合を提供します。Webサービスが使用する主要プロトコルは、SOAP、HTTP GETとPOST、MIME、SMTPです。

SOAPメッセージのプロトコルは、XML言語に基づいています。これはDCOM規格とCORBA規格に類似しています。SOAPは、COM、DCOM、Internet ExplorerとMicrosoftのJava実装によってサポートされています。

Webサービスコネクタは、WSDL（Web Services Description Language、Webサービス記述言語）プロトコルを用いて、Webサービスの自動記述を取得します。この自動記述には、サービスが使用する通信プロトコルと、演算が処理できるフォーマットのリストが含まれています。

使用可能なWebサービスのリストは、<http://www.xmethods.com>などの他多数のサイトに記載されています。

統合シナリオでは、Webサービスコネクタはまずドキュメントを取り込み、そのドキュメントをWebサービスへ要求の形で送信します。コネクタは応答を受信すると、自動的に応答をドキュメントに変換して生成します。このコネクタはマッピングボックスのように機能し、送信するデータ型（要求）と受信するデータ型（要求への応答）の間のマッピングを実行します。

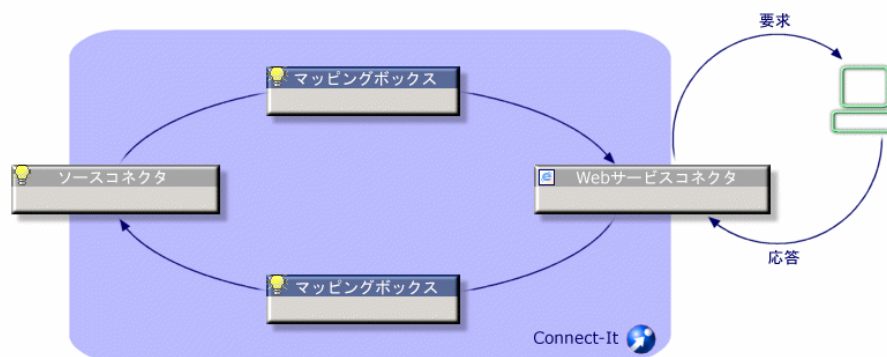
Webサービスコネクタの使用例

- 1 外部アプリケーションは郵便番号を含むドキュメントを生成し、マッピングはWebサービスコネクタが取り込めるように、このドキュメントを変換します。
- 2 コネクタは気温を提供するWebサービスへ要求を送信し、郵便番号をパラメータとして渡します。
- 3 Webサービスは、郵便番号に対応する気温を含む応答を返信します。

これを受信するWebサービスコネクタは、自動的にConnect-It内にドキュメントを生成します。

- 4 生成されるドキュメントは、郵便番号を提供した外部アプリケーションへ、フィードバックとして送信されます。

図 18.1. Webサービスコネクタ - 機能の概要



 注意:

一部のWebサービスは要求 / 応答のモデルで機能しないため、この場合応答はありません。

Webサービスコネクタの制約点

コネクタは以下の操作をサポートしません。

- 同名の複数の演算

Webサービスコネクタは、1つのWebサービス内で固有名の演算のみをサポートします。Webサービスが同一名の2つの演算を参照すると、第1の演算のみがコネクタの使用可能なドキュメントタイプとして発行されます。

- 複数の応答がある要求

Webサービスコネクタは、各要求（取り込み用ドキュメントタイプ）に、1つの応答（生成用ドキュメントタイプ）を関連付けます。Webサービスが1つの要求に複数の応答を関連付けると、最初の演算のみがコネクタの使用可能なドキュメントタイプとして発行されます。

- 複数の要求がある応答
Webサービスが1つの応答に複数の要求を関連付けると、最初の演算のみがコネクタの使用可能なドキュメントタイプとして発行されます。
- 再帰的な構造のドキュメントタイプ
Webサービスコネクタは、同ドキュメントタイプ内で1要素が別の要素を参照する構造のドキュメントタイプを発行しません（例：リレーショナルデータベースのモデル）。
- Webサービスに処理されるフォーマットを記述する部分のWSDLへ、別のフォーマットをインポートする

表 18.5. サポートされるプロトコル - Webサービスコネクタ




プロトコル	サポート可/不可
SOAP RPC	可
SOAP Messaging (HTTPプロトコルによる SOAP)	可
HTTP get	可
HTTP post	可
SOAP FTP	不可
SOAP SMTP	不可
MIME	不可

Webサービスコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定」章の「コネクタを設定する」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 18.6. Webサービスコネクタの設定

 _____	コネクタに名前を付け、役割を説明する
 _____	コネクタの導入
 _____	ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定」章の「コネクタに名前を付け、役割を説明する」の節を参照してください。

コネクタの導入

このページでは、Webサービスへのコネクタの接続パラメータを設定できます。

WSDLのアドレス

WSDL URLアドレスによって、各Webサービス用に以下の内容を取得できるようになります。

- 使用される通信プロトコル
- サポートされる演算操作
- 演算のフォーマット (XMLスキーマ、DTD、Javaオブジェクトなど)

例 : <http://www.xmethods.net/sd/2001/TemperatureService.wsdl>

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定」章の「ピボットドキュメントタイプの使用」の節を参照してください。

Webサービスコネクタの取り込み用ルール

コネクタの取り込み用ルールの入力方法については、「コネクタのルール (ディレクティブ)」章の「取り込み用ルール」の節を参照してください。

適用する演算

コネクタがデータをWebサービスサーバへ送信できるようにするには、演算「query」を選択します。

【適用する演算】フィールドで「なし」を選択しても、「query」演算が使用されます。この場合、この演算が実行されたことを示す警告メッセージが、ドキュメントログ内に表示されます。

使用可能なドキュメントタイプ

Webサービスコネクタは、拡張型処理レポートに対応する使用可能なドキュメントタイプを、生成用に発行します。

処理レポートの詳細については、「処理レポート」の章を参照してください。この拡張内容は、処理レポートのルートノード下に、Webサービスの演算に対応する要素として表示されます。

例：XMethods Temperature Serviceサービスは2つのメソッドGetTempとGetWeatherを発行します。サービスに送信される要求内では、この2つのメソッド用に、郵便番号に対応する文字列型のパラメータが予期されています。

取り込み用に使用可能なドキュメントタイプは、Webサービスの演算に当たります。

各演算ごとに、1つの使用可能なドキュメントタイプが取り込み用（要求の送信）と生成用（要求への応答）にそれぞれ発行されます。

使用可能なドキュメントタイプの名前

表 18.7. 拡張型処理レポートの名前 - Webサービスコネクタ

プロトコル	生成用に使用可能なドキュメントタイプ (拡張型処理レポート)の接頭文字
SOAP RPC	ProcessReportSoapRpc-[演算名]
SOAP Messaging (HTTPプロトコルによる Soap)	ProcessReportSoapHttp-[演算名]
HTTP Get	ProcessReportHttpGet-[演算名]
HTTP Post	ProcessReportHttpPost-[演算名]

注意:

Webサービスコネクタは、サポートされていないプロトコルを使用する演算を、使用可能なドキュメントタイプとしては発行しません（例：MIMEプロトコル）。

Webサービスコネクタの使用可能なドキュメントタイプの名前には、通信プロトコルを表す接頭文字と、演算の名前が含まれます。生成用ドキュメントタイプでは、名前に接頭文字ProcessReportが付きます。

用例シナリオ - Webサービスコネクタ

このコネクタを使用する用例シナリオは以下の通りです。

- ws/ac41/currency.scn

JDBCコネクタ

オプションコネクタ

- 生成（ソース）
- 取り込み（ターゲット）

JDBCコネクタは、SQLクエリ言語を使用するデータベースへのアクセスを可能にするJava DataBase Connectivityインタフェースを使用します。

JDBCコネクタの互換性

JDBCコネクタは、以下のJDBCドライバとのテストに成功しています。






- Sun Systems sun.jdbc.odbc.JdbcOdbcDriver
- Microsoft SQL Server 2000 for JDBC
- Sybase jConnect 4.5/5.5
- Oracle JDBC 9.01

JDBCコネクタの設定

ウィザードでコネクタを設定できます。「コネクタの設定」章の「コネクタを設定する」の節を参照してください。

コネクタの設定ウィザードの各ページは以下の通りです。

表 18.8. JDBCコネクタの設定

	コネクタに名前を付け、役割を説明する
	コネクタの導入
	スケジュールのポイントを設定する
	トランザクションを設定する
	ピボットドキュメントタイプの使用

コネクタに名前を付け、役割を説明する

このページの使用方法については、「コネクタの設定」章の「コネクタに名前を付け、役割を説明する」の節を参照してください。

コネクタの導入

このページでは、データベースへの接続パラメータを設定できます。

URL

[JDBCドライバ]フィールドにデフォルトで指定されるドライバと共に、このフィールドでは、シンタックス「jdbc:odbc:[データソース名]」を使用しなければなりません。

例 : jdbc:odbc:AC420FR

デフォルトユーザ

データベースへ接続するユーザに対応する名前を入力します。

デフォルトのパスワード

デフォルトのユーザに関連付けられたパスワードを入力します。

JDBCドライバ

このパラメータのデフォルト値は、「sun.jdbc.odbc.JdbcOdbcDriver」です。このドライバにはJREまたはJDKが付属しています。この汎用ドライバに問題が発生する場合、ユーザはデータベースサーバに適したドライバを使う必要があります。

メタ記述ファイル

このフィールドには、JDBCコネクタのカスタマイズ情報に対応する「customizedSchema.xml」ファイルのパスを入力します。

サーバとの時間差を指定する

このページの使用方法については、「コネクタの設定」章の「サーバとの時間差を指定する」の節を参照してください。

トランザクションを設定する

このページの使用方法については、「コネクタの設定」章の「トランザクションを設定する」の節を参照してください。

ピボットドキュメントタイプの使用

このページの使用方法については、「コネクタの設定」章の「ピボットドキュメントタイプの使用」の節を参照してください。

JDBCコネクタの生成用ルール

コネクタの生成用ルールの入力方法については、「コネクタのルール（ディレクティブ）」章の「生成用ルール」の節を参照してください。

このコネクタでは、これらのルールはWHERE句やORDERBY句で作成されており、ソースデータベースのレコードにフィルタをかけることができます。これらの句の詳細については「コネクタのルール（ディレクティブ）」章の「生成用ルール」節、「WHERE句とORDER BY句」を参照してください。

JDBCコネクタの取り込み用ルール

コネクタの取り込み用ルールの入力方法については、「コネクタのルール（ディレクティブ）」章の「取り込み用ルール」の節を参照してください。

取り込み用ルールを作成するには、**[整合性]**と**[高度な整合性チェック]**タブ内で整合性用のパラメータを入力します。

これらの整合性用タブの詳細については、「コネクタのルール（ディレクティブ）」章の「取り込み用ルール」節内「整合性」を参照してください。

用例シナリオ - JDBCコネクタ

このコネクタ用の用例シナリオはありません。

19 | 用例シナリオ

この章では、Connect-It付属の用例シナリオについて説明します。
シナリオはソースコネクタによって分類されています。

Unicenter AMOシナリオ

本節では、Unicenter AMOコネクタを使用する既製シナリオについて簡潔に説明します。

amo / amo2am.scn

このシナリオは、Unicenter AMOデータベースを、ピボットドキュメントタイプを經由してAsset Managementアプリケーションへ移行します。
移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア
- 関連付けられた周辺機器

InfraTools Desktop Discoveryシナリオ

本節では、InfraTools Desktop Discoveryコネクタを使用する既製シナリオについて簡潔に説明します。

idd /iddac36 /iddac.scn

このシナリオは、Asset Management 3.6データベースの [資産] テーブルのコンピュータに関する情報を更新します。

idd /iddac40 /iddac.scn

このシナリオは、Asset Management 4.0データベースの [資産] テーブルのコンピュータに関する情報を更新します。

idd /iddac41 /iddac.scn

このシナリオは、Asset Management 4.1データベースの [コンピュータ] テーブルのレコードに関する情報を更新します。

idd /idditsm /iddam.scn

このシナリオは、Remedy Action Request Systemsデータベースを、以下の内容に関するデータを用いて更新します。

- 資産
- コンピュータ
- プリンタ

idd /idditsm /iddhd.scn

このシナリオは、Remedy Action Request Systems (HelpDesk) データベースを、コンピュータに関するデータを用いて更新します。

idd /dds /ddsc.scn

このシナリオは、ServiceCenterデータベースを以下の内容に関するデータを用いて更新します。

- コンピュータ
- 連絡先
- ソフトウェア
- プリンタ

rsm /dd2rsm.scn

このシナリオは、Remedy Action Request Systemsデータベースを、コンピュータに関するデータを用いて更新します。

Intel LANDesk シナリオ

本節では、Intel LANDeskコネクタを使用する既製シナリオについて簡潔に説明します。

ild /ld2am.scn

このシナリオは、Intel LANDesk 6.3と6.4のデータベースを、Asset Management 3.6アプリケーションへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア
- 周辺機器

ild /ld65am.scn

このシナリオは、Intel LANDesk 6.5のデータベースを、Asset Management 3.6アプリケーションへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア
- 周辺機器

InfraTools Managementシナリオ

本節では、InfraTools Managementコネクタを使用する用例シナリオについて簡潔に説明します。

im /htsec2im.scn

このシナリオは、NTドメイン特有の情報（ユーザとコンピュータ）をInfraToolsデータベースへ転送します。これらの情報は、InfraTools Administratorモジュールで使用されます。InfraTools Administratorモジュールは、リモートコントロール操作をセキュリティで保護するためにこれらのデータを使用します。マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
MachineSrc	amComputerDst
UserSrc	amEmplDeptDst

InfraTools Network Discoveryシナリオ

本節では、InfraTools Network Discoveryコネクタを使用する用例シナリオについて簡潔に説明します。

ind /indac36 /indac.scn

このシナリオは、InfraTools Network Discoveryが監視するIT資産に関するデータを、Asset Management 3.6アプリケーションの資産のテーブルへ移行します。マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
FULL_DEVICE	amAsset
FULL_DEVICE_CNX	amAssetCnx
EVENT	amAsset1
EVENT	amAssetDst
EVENT	amTicketDst

ind /indac40 /indac.scn

このシナリオは、Asset Management 4.0アプリケーションで、ind /indac36 /indac.scnシナリオと同じ機能を果たします。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
FULL_DEVICE	amAsset
FULL_DEVICE_CNX	amAssetCnx

ind /indac41 /indac.scn

このシナリオは、Asset Management 4.1アプリケーションで、ind /indac36 /indac.scnシナリオと同じ機能を果たします。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
FULL_DEVICE	amAsset
FULL_DEVICE_CNX	amAssetCnx

rsm /ind2rsm.scn

このシナリオは、InfraTools Network Discoveryが監視するIT資産に関するデータを、Action Request Systemデータベースへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
FULL_DEVICE_Src	IND:InventoryDst

Action Request Systemシナリオ

本節では、Action Request Systemコネクタを使用する用例シナリオについて簡潔に説明します。

itsm-am /cmipo.scn

このシナリオは、Action Request Systemデータベースのデータを、Asset Managementアプリケーションへ移行します。

このシナリオによって移行されるデータは以下の通りです。

- 購入依頼
- カテゴリ
- モデル

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
CHG:ChangeSrc (Request creation)	amRequestDst1
CHG:ChangeSrc (Category creation)	amCategoryDst1
CHG:ChangeSrc (Model creation)	amProductDst

itsm-am /sharedat.scn

このシナリオは、Action Request Systemデータベースのデータを、Asset Managementアプリケーションへ移行します。

このシナリオによって移行されるデータは以下の通りです。

- 従業員
- 場所
- 資産

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
SHR:PeopleSrc1	amEmplDeptDst1
SHR:LocationSrc	amLocationDst
AST:AssetSrc	amAssetDst

LDAPシナリオ

本節では、LDAPコネクタを使用する用例シナリオについて簡潔に説明します。

ldap /ac /complete.scn

このシナリオは、LDAPディレクトリのデータを、Asset Managementアプリケーションの部署と従業員のテーブルへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
inetOrgPersonSrc	amEmplDeptDst

ldap /ac /simple.scn

このシナリオは、LDAPディレクトリのデータを、Asset Managementアプリケーションの部署と従業員のテーブルへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
organizationalPersonSrc	amEmplDeptDst

ldap /notes /hames.scn

このシナリオは、LDAPディレクトリのデータを、Lotus Notesデータベースへ移行します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
organizationalPersonSrc1	PersonDst

Eメールシナリオ

本節では、Eメールコネクタを使用する用例シナリオについて簡潔に説明します。

ac36 /finrequе.scn

このシナリオは、依頼のテーブルのレコードに対応する購入依頼を、Eメールアドレス（購入責任者のアドレス）へ送信します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
amRequestSrc	OutMailMessageDst

ac36 /finconfi.scn

このシナリオは、購入依頼の承認のメッセージを送信します。Asset Managementアプリケーションの購入依頼のテーブル内で、依頼のステータスが変更されません。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
InMailMessageSrc	amRequestDst

ac36 /hewemplo.scn

このシナリオは、Eメールメッセージに基づいて、Asset Managementアプリケーションの従業員のテーブル内にレコードを作成します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
InMailMessageSrc	amEmplDeptDst

ac40 /finconfi.scn

このシナリオは、購入依頼の承認のメッセージを送信します。Asset Managementアプリケーションの購入依頼のテーブル内で、依頼のステータスが変更されません。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
InMailMessageSrc	amRequestDst

ac40 /finrequе.scn

このシナリオは、依頼のテーブルのレコードに対応する購入依頼を、Eメールアドレス（購入責任者のアドレス）へ送信します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
amRequestSrc	OutMailMessageDst

ac40 /hewemplo.scn

このシナリオは、Eメールメッセージに基づいて、Asset Managementアプリケーションの従業員のテーブル内にレコードを作成します。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
InMailMessageSrc	amEmplDeptDst

MQSeriesシナリオ

本節では、MQSeriesコネクタを使用する用例シナリオについて簡潔に説明します。

mqseries /acmq /mqtoprgn.scn

このシナリオは、mqseriesメッセージに基づいて、Asset Managementアプリケーションの以下のテーブル内にレコードを作成します。

- コストセンタのテーブル (amCostCenter)
- サプライヤのテーブル (amCompany)
- 発注のテーブル (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 /acmq/prgntomq.scn

このシナリオは、

- Asset Managementアプリケーションの以下のテーブルのレコードに基づいて、mqseriesメッセージを作成します。
 - 発注のテーブル (amPOOrder)
 - 受領のテーブル (amDeliv)
- 上記のテーブルを以下のキューに基づいて更新します。
- PEREGRINE.OUTキュー
- PEREGRINE.INキュー

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
amPOOrderSrc	PO
amDeliv	Receipt

mqseries /scmq/mqsc.scn

このシナリオは、

- PEREGRINE.INキューにあるMQSeriesメッセージに基づいて、ServiceCenterのcontactファイル内にレコードを作成します。

- ServiceCenterのcontactファイルのレコードに基づいて、PEREGRINE.OUT.PERSONキュー内にMQSeriesメッセージを作成します。マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
External Contacts	Contacts from Exterior
Contacts of ServiceCenter	Contacts from ServiceCenter

NTセキュリティシナリオ

本節では、NTセキュリティコネクタを使用する既製シナリオについて簡潔に説明します。

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、Asset Management 3.6へインポートします。

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、Asset Management 3.6へインポートします。

ntac40 /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、Asset Management 4.0へインポートします。

ntac40 /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、Asset Management 4.0へインポートします。

ntac41 /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、Asset Management 4.1へインポートします。

ntac41 /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、Asset Management 4.1へインポートします。

ServiceCenterシナリオ

本節では、ServiceCenterコネクタを使用する用例シナリオについて簡潔に説明します。

「accscps/dataprop」フォルダのシナリオは、シナリオの接頭番号の順番で実行されなければなりません。例えば、01-SCcompany2ACamCompany.scnシナリオを1番目に実行してから、02-SCLocation2ACamLocation.scnシナリオを実行します。

scac /sc3ac36 /scac.scn

この複製（レプリケーション）シナリオは、Asset 3.6のデータベースからServiceCenter3.6へデータを作成または更新します。

Asset Managementアプリケーションの設定

本節では、用例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たコンピュータ上に作成されているデータベース接続のリストを含んでいます。このファイルは、Asset Managementアプリケーションのインストール時に自動的に作成され、Windowsのシステムフォルダに含まれます。

例

C:\WINNT\amdb.ini.

Asset Managementコネクタは、このファイルを読み込み、コネクタの設定時に使用可能な接続のリストを表示します。使用中のコンピュータが、Asset Management

アプリケーションへアクセスできない場合、Asset Managementコネクタは使用不可能です。

任意管理項目のテーブル

Asset Managementアプリケーションのデータベースでは、任意管理項目を用いてデータモデルを拡張できます。

Connect-Itにおいて、任意管理項目はAsset Managementコネクタの使用可能なドキュメントタイプ内で、フィールドまたはコレクションの形で表示されます。

例：

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、『データベース管理』の「インポート」の章を参照してください。

Asset Managementデータベースの必須フィールド

Asset Managementアプリケーションでは、テーブルのフィールドやリンクが必須である場合があります。これはデフォルトで必須に設定されているか、またはAsset Managementアプリケーションの管理者により必須に設定されています。

整合性チェック時に、Asset Managementコネクタの使用可能なドキュメントタイプ内の各構造体は、1つのレコードに当たります。この構造体内の1要素が必須フィールドであるにも関わらず入力されていない場合は、要素の所属する構造体は拒否されます。

例：

資産のテーブル内では[AssetTag]は必須フィールドです。資産のテーブルに対応するドキュメントタイプ内で、このフィールド用の要素が入力されていないと、要素の所属先の構造体は拒否されます。

API

Asset Managementコネクタを適切に使用するには、AssetCenter、AssetCenter Cable and Circuit、InfraCenterの完全インストールを実行するか、またはカスタムインストールで、コンポーネント「API」を選択する必要があります。

ServiceCenterの設定

本節では、用例シナリオを使用できるようにServiceCenterを設定する方法を説明します。

入力イベントの作成

シナリオを適切に機能させるためには、以下の2UNLファイルのいずれかをServiceCenter内にインポートする必要があります。

- scac.unl
- scacsp2.unl

これらのファイルは、「device」に関連付けられ、ICM接頭文字を含むイベントを更新します。また、「model」と「modelvendor」ファイルに関連付けられた妥当性の規則も変更します。これらのファイルはServiceCenterの標準データキットも整理します。

警告:

これらのファイルを、プロダクションモードで使用するServiceCenterデータベース内へはインポートしないでください。これらのファイルは、ペレグリンシステムズのサポートから入手できます。

入力イベントの同期をとる

ServiceCenter内の入力イベントは、同期式または非同期式に処理されています。非同期モードでは、ServiceCenter用スケジュールが、イベントキューに一時的に保存されているイベントを処理する時期を決定します。同期モードでは、イベントはServiceCenterイベントキューに挿入されるとすぐに処理されます。

シナリオのテスト中に、シナリオが機能するかどうかをすぐに確認する場合は、同期モードを選択します。

Connect-Itは同期モードで機能するため、非同期モードで処理すると、ServiceCenterでデータが適切に処理されたかどうかを即時に確認することができなくなります。この場合、ドキュメント処理が待機中であることを知らせる警告メッセージが表示されます。

同期モードでは、ドキュメント処理の成功または失敗のメッセージが表示されます。

適用コンテキスト

用例シナリオは、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）を実行するデモ用シナリオです。

 **警告:**

このシナリオは、プロダクション用にインストールされたServiceCenterアプリケーションには当てはまりません。このシナリオは単なる用例に過ぎないことを念頭に置いて下さい。

このシナリオは、資産のリポジトリ、製品、サプライヤ、場所、従業員と部署に関するデータのマッピングのみに制限されています。

これらのデータキットは、ServiceCenterとAsset Managementアプリケーション間のデータの複製用に作成されてはいないため、データは非常に相違しています。

実際の企業では、ServiceCenterデータベース内に保存される従業員のリストは、Asset Managementデータベースの従業員のリストに一致するはずで

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。

用例シナリオでは、まず第1にServiceCenterのデータをAsset Managementアプリケーションへインポートし、またその反対も実行します。

この作業は問題の一部を解決し、またServiceCenterとAsset Managementアプリケーション間で実際に複製（レプリケーション）を実行する際に適用する一部の規則を明らかにします。

以下の3つの規則の適用が可能になり次第、すぐに適用するようにします。

- 各リポジトリは1つのアプリケーションのみに管理されなければならない。
矛盾を回避するために、あるリポジトリのレコードは、常に同じアプリケーション（ServiceCenterまたはAsset Managementアプリケーション）で作成、変更、削除されなければなりません。これによって複製のプロセスは、あるリポジトリのレコードを別のアプリケーション内で更新するだけになります。例えば、資産のリポジトリは、AssetCenterで管理し、ヘルプデスクのチケットはServiceCenterで管理します。リポジトリは、データの内容がより正確で、データの処理方法がより自然であるアプリケーション内で管理されるべきです。リポジトリを1つのアプリケーションのみで管理すると重要な問題が生じる場合は、別のアプリケーションを小規模な更新時に使用します。しかし、新規レコードの作成やテーブルの識別子の変更などには、このアプリケーションを使用してはなりません。
- レコードを作成したアプリケーションの情報は保存されなければならない。
トラッキング情報を保存することにより、複製が適切に実行されたかどうかを確認できるのに加えて、レコードを作成したアプリケーションに応じて特定のマッピングを作成できるようになります。
- Asset Management アプリケーションとServiceCenter間のデータは調整されなければならない。
マッピングを簡略にするには、2アプリケーション内でデータが同じ方法で処理されなければなりません。例えば、Asset Managementアプリケーション

で製品にデフォルトで関連付けられるカテゴリの一部は、ServiceCenterの「model」に関連付けられるカテゴリとは全く違います。マッピングが適切に作成されるには、同じカテゴリが2アプリケーション内で定義されていなければなりません。

データの整合性チェックとデータの複製の相違点

データの整合性チェックとデータの複製との間には、大きな相違点があります。

整合性チェック

整合性チェックとは、別アプリケーションから受信するデータの統合であり、ソースアプリケーションから来るデータの方が、ターゲットアプリケーション内の既存のデータよりも新しいと見なされています。

このプロセスは、以下の質問とその答えに応じて実行されます。

「受信するデータは、ターゲットアプリケーション内に既存するか?」

- 答えが「いいえ」であると、データは挿入されます。
- 答えが「はい」であると、受信するデータの新規情報に応じて、既存のデータが更新されます。

ターゲットアプリケーションが空である場合は、このプロセスは整合性チェックではなく「インポート」になります。

複製（レプリケーション）

複製のプロセスでは、受信するデータは最新のデータであるとは見なされません。

受信するデータと既存データの最終変更日は、最終複製日と比較されます。どのデータが最も新しい情報であるかを決定するために、この比較が実行されます。

受信するデータと既存のデータが両方とも最終複製日以降に変更されていると、複製の矛盾が発生します。

この場合、どちらのデータがより新しいかを知るための確実な方法はありません。複製シナリオの作成者が、複製エンジンの操作方法を規定しなければなりません。

以下の規則が適用されます。

- A：受信データが拒否される
- B：受信データが最新データであると見なされ、選択される
- C：既存データが最新データであると見なされ、選択される

Connect-It内でのデータの複製の管理

ServiceCenterコネクタは、ドキュメントを取り込む際に、ServiceCenterデータベース内に直接レコードを作成しません。取り込まれたドキュメントから、入力イベントが作成され、キュー（event queue entry）内に追加されます。

この結果整合性チェックのプロセスは、複製のプロセスと同様に、入力イベントがServiceCenterサーバに処理される時に内部で管理されます。

ServiceCenterサーバ内の複製の問題を解決するのは、シナリオの実行責任者の役目になります。

Asset Managementコネクタを使用すると、Asset Managementアプリケーションのデータベースと直接相互作用することが可能になり、整合性のプロセスが（複製のプロセスと同様に）Connect-It内で完全に管理されるようになります。

以下の節では次の点について説明します。

- Asset Managementアプリケーションが複製の矛盾を検出する方法
- 矛盾が発生した場合に実行すべき操作
- データの複製を可能にする日付（日時）が比較される方法

複製の矛盾の検出

Asset Managementアプリケーションでは、取り込まれるドキュメントに【**変更日**】フィールド（Connect-It SQL名：dtLastModif）が含まれていると、複製のプロセスが起動します。

このプロセスは、このフィールドを含む全構造体と全コレクションに適用されます。

シナリオ内で、このフィールドはServiceCenterの【sysmodtime】フィールドにマップされなければなりません。

このシナリオでは、ドキュメントがServiceCenterコネクタに生成される場合は、ドキュメントが最終複製日以降に変更されたこと（またはドキュメントが最終複製日に存在しなかったこと）を意味します。この結果次の質問に対する答えが重要になります。

「ドキュメントに含まれるフィールドは、最終複製日以降にAsset Managementアプリケーション内で変更されたか？」

- 変更されていた場合、複製の矛盾が発生します。
- 変更されていなかった場合（Asset Managementアプリケーションのデータベース内に、データがまだ存在していない場合などがこれに当たります）、データは更新または挿入されます。

矛盾が発生した場合に実行すべき操作

複製の矛盾が検出された場合、矛盾は生成されるドキュメント内の構造体またはコレクションのフィールドに関係します。

矛盾に関連するレコードでは、以下の3つの操作が可能です。

- 拒否
この場合、[dtLastModif] フィールドに関するフィールドのエラーが発生します。Asset Managementアプリケーションのデータは変更されません。
- フィールドに対する警告
この場合、[dtLastModif] フィールドに関するフィールドの警告が発生します。Asset Managementアプリケーションのデータは更新されます。
- 上書き
この場合、Asset Managementアプリケーション内のデータは、ServiceCenterから来るデータにより、警告なしで上書きされます。

Asset Managementコネクタの取り込み用ルールタブで、これらのオプションのうち1つを選択できます。

日付の比較

複製プロセスで日付（日時）の比較を実行するには、日付が正確である必要があります。日付の正確さは以下の要因に左右されます。

- タイムゾーン
- サーバとの時間差

タイムゾーン

GMT +1タイムゾーンに基づいた日付と、GMT -7タイムゾーンに基づいた日付を、直接比較することは不可能です。まず変換作業が必要になります。

Connect-Itでは、GMTタイムゾーンに基づいた日付が、常に格納され使用されません。GMT日付をConnect-Itへ提供するのは、各コネクタの役割です。

また日付は、Connect-Itを使用しているコンピュータでの設定に応じて表示されます。

サーバとの時間差

全サーバの時間が完全に一致するのが理想的です。しかし実際には、一部のサーバの時間は他のサーバより進んだり遅れたりしています。

Connect-Itでは、コネクタが提供する日時のデータは、ある一定の応答時間に応じて調整されます。応答時間は負の時間になり得ます。

この調節以外では、サーバの応答時間はConnect-It内で隠されています。

scac /sc3ac40 /scac.scn

このシナリオは、scac /sc3ac36 /scac.scnシナリオと同じ機能を果たします。
この場合複製は、ServiceCenterバージョン3.0と、Asset Managementアプリケーションのバージョン4.0間で実行されます。

scac /sc3ac41 /scac.scn

このシナリオは、scac /sc3ac36 /scac.scnシナリオと同じ機能を果たします。
この場合複製は、ServiceCenterバージョン3.0と、Asset Managementアプリケーションのバージョン4.1間で実行されます。

scac /sc4ac40 /scac.scn

このシナリオは、scac /sc3ac36 /scac.scnシナリオと同じ機能を果たします。
この場合複製は、ServiceCenterバージョン4.0と、Asset Managementアプリケーションのバージョン4.0間で実行されます。

scac /sc4ac41 /scac.scn

このシナリオは、scac /sc3ac36 /scac.scnシナリオと同じ機能を果たします。
この場合複製は、ServiceCenterバージョン4.0と、Asset Managementアプリケーションのバージョン4.1間で実行されます。

scauto /scacfg /scacfg.scn

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
Output Event Type	InputEventTypes
Input Event Type	OutputEventTypes
EventTypes	SCAutoConfiguration

acscups /dataprop /01-SCcompany2ACamCompany.scn

このシナリオは、ServiceCenterの「company」ファイルのデータと、Asset Managementアプリケーションの [サプライヤ] テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups /dataprop」フォルダ内のシナリオの実行順序を表しています。

acscups /dataprop /02-SClocation2ACamLocation.scn

このシナリオは、ServiceCenterの「location」ファイルのデータと、Asset Managementアプリケーションの [場所] テーブルのレコード間の統合を実行します。

このシナリオを実行した後、Asset Managementアプリケーションの [場所] のテーブルの [関連所在地] フィールドを確認する必要があります。親所在地と関連所在地のリンクが破損している場合は、リンクを手動で作成する必要があります。

シナリオの接頭番号は、「acscups /dataprop」フォルダ内のシナリオの実行順序を表しています。

acscups /dataprop /03-SCdept2ACamEmpDept.scn

このシナリオは、ServiceCenterの「department」ファイルのデータと、Asset Managementアプリケーションの [部署と従業員] テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups /dataprop」フォルダ内のシナリオの実行順序を表しています。

acscups /dataprop /04-SCcontacts2ACamEmpDept.scn

このシナリオは、ServiceCenterの「contacts」ファイルのデータと、Asset Managementアプリケーションの [部署と従業員] テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups /dataprop」フォルダ内のシナリオの実行順序を表しています。

acscups /dataprop /05-SCdevtype2ACamNature.scn

このシナリオは、ServiceCenterの「device types」ファイルのデータと、Asset Managementアプリケーションの [属性] テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups /dataprop」フォルダ内のシナリオの実行順番を表しています。

acscups /dataprop /06-SCocmlcat2ACamNature.scn

このシナリオは、ServiceCenterの「categories」ファイルのデータと、Asset Managementアプリケーションの [属性] テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups /dataprop」フォルダ内のシナリオの実行順番を表しています。

acscups /dataprop /07-SCmodel2ACamModel.scn

このシナリオは、ServiceCenterの「model」ファイルのデータと、Asset Managementアプリケーションの [モデル] テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups /dataprop」フォルダ内のシナリオの実行順番を表しています。

acscups /dataprop /08-SCdevtype2ACamModel.scn

このシナリオは、ServiceCenterの「device type」ファイルのデータと、Asset Managementアプリケーションの [モデル] テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups /dataprop」フォルダ内のシナリオの実行順番を表しています。

acscups /dataprop /09-SCdevice2AACamPortfolioAndamComputer.scn

このシナリオは、ServiceCenterの「device data」ファイルのデータと、Asset Managementアプリケーションの [ポートフォリオ品目] と [コンピュータ] テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups /dataprop」フォルダ内のシナリオの実行順番を表しています。

acscups/dataprop/10-SClaptop2ACamPortfolioAndamComputer.scn

このシナリオは、ServiceCenterの「laptop」ファイルのデータと、Asset Managementアプリケーションの[ポートフォリオ品目]と[コンピュータ]テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups/dataprop」フォルダ内のシナリオの実行順番を表しています。

acscups/dataprop/11-SCworkstation2ACamPortfolioAndamComputer.scn

このシナリオは、ServiceCenterの「workstation」ファイルのデータと、Asset Managementアプリケーションの[ポートフォリオ品目]と[コンピュータ]テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups/dataprop」フォルダ内のシナリオの実行順番を表しています。

acscups/dataprop/12-SCpc2ACamAndamComputer.scn

このシナリオは、ServiceCenterの「pc」ファイルのデータと、Asset Managementアプリケーションの[コンピュータ]テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups/dataprop」フォルダ内のシナリオの実行順番を表しています。

acscups/dataprop/13-SCserver2ACamAndamComputer.scn

このシナリオは、ServiceCenterの「server」ファイルのデータと、Asset Managementアプリケーションの[コンピュータ]テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups/dataprop」フォルダ内のシナリオの実行順番を表しています。

acscups/dataprop/14-SCphone2ACamPortfolioAndamPhone.scn

このシナリオは、ServiceCenterの「server」ファイルのデータと、Asset Managementアプリケーションの[電話機]テーブルのレコード間の統合を実行します。

シナリオの接頭番号は、「acscups/dataprop」フォルダ内のシナリオの実行順番を表しています。

SMSシナリオ

本節では、SMS 1.xとSMS 2.xコネクタを使用する用例シナリオについて簡潔に説明します。

sms/sms1am.scn

このシナリオは、SMS 1.xデータベースの情報（IT資産）を、バージョン4.1より前のAsset Managementアプリケーションの [資産] のテーブルへ移行します。マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
MachinesSrc	amAssetDst

sms/sms1amlt.scn

このシナリオは、SMS 1.xデータベースの情報（IT資産）を、バージョン4.1より前のAsset Managementアプリケーションの [資産] のテーブルへ移行します。このシナリオは、`sms/sms1am.scn`シナリオの簡易バージョンです。マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
MachinesSrc	amAssetDst

sms/sms2am.scn

このシナリオは、SMS 2.xデータベースの情報（IT資産）を、バージョン4.1より前のAsset Managementアプリケーションの [資産] のテーブルへ移行します。マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
System_DATASrc	amAssetDst

sms /sms2amlt.scn

このシナリオは、SMS2.xデータベースの情報（IT資産）を、バージョン4.1より前のAsset Managementアプリケーションの [資産] のテーブルへ移行します。

このシナリオは、[sms /sms2am.scn](#)シナリオの簡易バージョンです。

マップされるソースドキュメントタイプとターゲットドキュメントタイプは、以下の表の通りです。

ソース	ターゲット
System_DATASrc	amAssetDst

sms /sms2ac41.scn

このシナリオは、SMSデータベースの情報（IT資産）を、Asset Management 4.1アプリケーションの [コンピュータ]（amComputer）のテーブルへ移行します。

ソース	ターゲット
System_DATASrc	amComputerDst

Tivoli Inventory Managementシナリオ

本節では、Tivoli Inventory Managementコネクタを使用する既製シナリオについて簡潔に説明します。

tim /tim2am

このシナリオは、Tivoli Inventory Management 3.1と3.6のデータベースを、Asset Management 3.6アプリケーションへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア
- 周辺機器

tim /tim4am

このシナリオは、Tivoli Inventory Management 4.0のデータベースを、Asset Management 4.0アプリケーションへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア

tim /tim4sc4

このシナリオは、Tivoli Inventory Management 4.0のデータベースを、ServiceCenter 4.0アプリケーションへ移行します。

移行されるデータは以下の通りです。

- コンピュータ
- ソフトウェア

ゲートウェイ3.xシナリオ

本節では、ゲートウェイ3.xコネクタを使用する用例シナリオについて簡潔に説明します。

uagw /uagw.scn

このシナリオは、ゲートウェイ3.xxから来るデータを、ピボットドキュメントタイプを経由してAsset Managementアプリケーションへ移行します。

移行されるデータは以下の通りです。

- IT資産内の全コンピュータ
- コンピュータの技術的な情報（任意管理項目）
- コンピュータの場所
- コンピュータが対応する製品
- インストールされたソフトウェア
- コンピュータのユーザ

Webサービスシナリオ

本節では、Webサービスコネクタを使用する用例シナリオについて簡潔に説明します。

ws/ac41/currency.scn

このシナリオは、Asset ManagementアプリケーションのamCurRateテーブルで既に定義された通貨間の為替レートを更新します。

このシナリオで使用されているWebサービスのWSDLアドレスは、<http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl>です。

Peregrine Desktop Inventoryシナリオ

本節では、Peregrine Desktop Inventoryコネクタを使用する用例シナリオについて簡潔に説明します。

pdi/pdiac41/pdiac.scn

このシナリオは、Peregrine Desktop Inventoryで取得したデータを使って、Asset Management 4.1データベースの[コンピュータ]テーブル(amComputer)のコンピュータに関する情報を更新します。

A | AQL言語

付録

本章ではAQLクエリ言語について説明します。

クエリの例

本節では、各例につきクエリ作成の1側面が説明されています。以下の例を変更したり組み合わせたりして、独自のクエリを作成してください。

以下の用例はクエリの完全なシンタックスを紹介しています。これらの例をそのままテストする場合は、InfraCenter Exportの使用をお勧めします。クエリによるフィルタ内などで用例クエリを使用する場合は、シンタックスを変更する必要があります。

例えば、次のような明示的に記述したクエリがあるとします。

```
SELECT self FROM amAsset WHERE Brand='Compaq'
```

これと同じ内容のクエリが、資産のテーブル内でクエリフィルタとして使用される場合は、次のように記述されます。

```
Brand='Compaq'
```

AssetCenterプログラムに付属しているデモ用データベースに格納されているクエリも、参照してください。



使用中のDBMSのSQLコードに書き換えられたクエリは、クエリの詳細画面の [プレビュー] タブページに表示されます。

メインテーブルのフィールドを特定の値と比較する

例：メーカーが「Compaq」であるすべての資産

```
SELECT Self FROM amAsset WHERE Brand = 'Compaq'
```

メインテーブルのリンクを別のリンクと比較する

例：親資産と場所が同じであるすべての資産

```
SELECT Self FROM amAsset WHERE Location = Parent.Location
```

メインテーブルのリンクを特定の値と比較する

例：「大阪支社」に直接リンクしているすべての部門と従業員

```
SELECT Self FROM amEmpIDept WHERE Parent.Name = '大阪支社'
```

メインテーブルにリンクしたテーブルのフィールドの値に応じて比較する

例：親資産と場所名が同じであるすべての資産

```
SELECT Self FROM amAsset WHERE Location.Name = Parent.Location.Name
```

階層構造のテーブル

[FullName] フィールドを使う

例：名前が「大阪支社」である場所の、すべての関連（従属）場所

```
SELECT Self FROM amLocation WHERE FullName LIKE '/大阪支社/%'
```

[FullName] フィールドと [sLvl] フィールドを使う

階層構造のテーブルのクエリでは、[FullName] フィールドと [sLvl] フィールドが頻繁に使用されます。

例：「大阪支社」場所の従属場所で、階層のレベルが3よりも小さいレコード（3は含まれない）

ツリー構造のルートの階層レベルは「0」です。

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/大阪支社/%') AND (sLvl < 3)
```

フルネームの先頭と末尾にスラッシュ (/) 記号が付くことに注意してください。

2つの条件を組み合わせたクエリ

例：役職が「会計部長」で、場所「府中支社」に位置するすべての従業員

```
SELECT Self FROM amEmpIDept WHERE (Title = '会計部長') AND (Location.Name = '府中支社')
```

数値、日付、テキスト型フィールドの比較

例：1995年1月1日から1995年12月31日までに実行したすべての作業指示

```
SELECT Self FROM amServiceCall WHERE (dtFirstCall >= #95/01/01 00:00:00#) AND (dtFirstCall <= #95/12/31 00:00:00#)
```

任意管理項目に関するクエリ

例：[Size]（SQL名）任意管理項目の値が150 cm以上である全ての資産

```
SELECT Self FROM amAsset WHERE fv_Size >= 150.00
```

式でレコードを検索する

例：購入価格がデータベース内で最高の購入価格に等しいすべての資産。メインクエリ内にサブクエリを使って最高価格を指定しています。

```
SELECT Self FROM amAsset WHERE mPrice = (SELECT max(mPrice) FROM amAsset)
```

未入力のフィールドを検索する

例：電話番号のないすべての従業員。空の文字列を一重引用符で囲みます。

```
SELECT Self FROM amEmpDept WHERE Phone=""
```

リンクなしのレコードを検索する

1リンクの場合

例：ユーザに割り当てられていないすべての資産。リンクがないことを「0」で表します。

```
SELECT Self FROM amAsset WHERE User = 0
```

nリンクの場合

例：資産にリンクしていないすべてのカテゴリ

```
SELECT Self FROM amCategory WHERE 0 = (SELECT COUNT(a2.IAstId) FROM amAsset a2 WHERE a2.ICatgId = ICatgId)
```

上記のクエリは、カテゴリのテーブルを走査して各カテゴリに属する資産の数を0と比較します。

1リンクとnリンクのテストを組み合わせた例

例：親資産も従属資産もないすべての資産

```
SELECT Self FROM amAsset WHERE (0 = (SELECT COUNT(a.IAstId) FROM amAsset a WHERE a.IParentId = IAstId)) AND (Parent = 0)
```

上記のクエリでは次の処理を行います。

- 1リンク (Parent = 0) のテスト。親資産のない資産を選択します。
- nリンク (0 = (SELECT COUNT(a.IAstId) FROM amAsset a WHERE a.IParentId = IAstId)) のテスト。従属資産のない資産を選択します。nリンクのテストは、各資産の識別子 [IAstId] を選択し、[IParentId] 識別子が [IAstId] に等しいすべての資産をカウントします。

別の組み合わせの例

「ハードドライブ」カテゴリの付属品がないすべての資産

```
SELECT self FROM amAsset p WHERE NOT ( EXISTS (SELECT IAstId FROM amAsset WHERE (FullName LIKE (p.FullName + '%/')) AND (Category.Name = 'ハードドライブ'))))
```

エイリアスを使ったクエリ

例：「Peregrine」研修プログラムと「データベース」研修プログラムを受講したすべての従業員

開始テーブル：部署と従業員のテーブル

クエリは次の通りです。

```
SELECT Self FROM amEmpIDept WHERE (Training_1.Title = 'Peregrine') AND (Training_2.Title = 'データベース')
```

Training_1とTraining_2と表記したエイリアスによって、Trainingリンクで関連付けられている2つの異なるレコードに2つの条件を定義できます。

次の例

```
SELECT Self FROM amEmpIDept WHERE (Training.Title = 'Peregrine') AND (Training.Title = 'データベース')
```

では、2つの名前がついた1つの研修コースを受講したすべての従業員を選択します。

別の例

```
SELECT Self FROM amEmpIDept WHERE (Training.Title = 'Peregrine') OR (Training.Title = 'データベース')
```

では、2つの名前の内いずれかの研修コースを1つ受講したすべての従業員を選択します。

AQLの関数

次のAQL関数は、クエリと計算式内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:

お使いのDBMS専用のSQL関数を使うことも可能です。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.1. 集合型のAQL関数

機能	説明
Avg(<列>)	数値型の<列>項目の平均を返します。<列>にレコードがない場合は「0」を返します。
Count(<列>)	列内の非NULL値をカウントします。
Countdistinct(<列>)	列内の異なる非NULL値をカウントします。
Max(<列>)	数値、文字列、または日付型の列の最大値を返します。 列にレコードがない場合は、0 (数値型の列)、空の文字列 (文字列型の列)、または空の日付 (日付型の列) を返します。
Min(<列>)	数値、文字列、または日付型の列の最小値を返します。 列にレコードがない場合は、0 (数値型の列)、空の文字列 (文字列型の列)、または空の日付 (日付型の列) を返します。
Sum(<列>)	数値型の列値の合計を返します。列にレコードがない場合は、「0」を返します。

上記の関数には、GROUP BY句とHAVING句を組み合わせて使います。

文字列型のAQL関数

表 A.2. 文字列型のAQL関数

機能	説明
Ascii(<文字列>)	<文字列>の先頭の文字のASCIIの値を返します。
Char(<n>)	ASCIIコード「n」の文字を返します。
Left(<文字列>, <n>)	<文字列>の先頭の「n」文字を返します。
Lower(<文字列>)	<文字列>を小文字で返します。
Ltrim(<文字列>)	<文字列>の左側のスペースを削除します。
Right(<文字列>, <n>)	<文字列>の末尾の「n」文字を返します。
Rtrim(<文字列>)	<文字列>の右側のスペースを削除します。
Substring(<文字列>, <n1>, <n2>)	<文字列>の第<n1>文字から始まる<n2>文字の部分文字列を抽出します(<文字列>の先頭文字を第1文字とします)。
Upper(<文字列>)	<文字列>を大文字で返します。

日付型のAQL関数

表 A.3. 日付型のAQL関数

機能	説明
Year(<日付>)	日付または日付+時刻型のフィールドの年を表す数字を返します(1997など)。
Month(<日付>)	日付または日付+時刻型のフィールドの月を表す数字を返します(1-12)。
Day(<日付>)	日付または日付+時刻型のフィールドで、その月の何日目かを表す数字を返します(1-31)。
DayOfYear(<日付>)	日付または日付+時刻型のフィールドで、その年の何日目かを表す数字を返します(1-366)。

機能	説明
WeekDay(<日付>)	日付または日付+時刻型のフィールドの曜日を表す数字を返します。 この数字は、サーバの設定によって変わります。例えば、SybaseまたはMicrosoft SQL Serverのデフォルトの設定は (1=日, 2=月, ..., 7=土)、Oracleのデフォルトの設定は (1=月, ..., 7=日) です。
Hour(<時間>)	時刻または日付+時刻型のフィールドの1日の時間を表す数字を返します (0-23)。
Minute(<時間>)	時刻または日付+時刻型のフィールドの分を表す数字を返します (0-59)。
Second(<時間>)	時刻または日付+時刻型のフィールドの秒を表す数字を返します (0-59)。
Getdate()	サーバの現在のシステム日付を返します。
AddDays(<日付>, <数値>)	日付または日付+時刻型のフィールドに特定の日数を加えます。
AddHours(<日付>, <数値>)	日付または日付+時刻型のフィールドに特定の時間数を加えます。
AddMinutes(<日付>, <数値>)	日付または日付+時刻型のフィールドに特定の分数を加えます。
AddSeconds(<日付>, <数値>)	日付または日付+時刻型のフィールドに特定の秒数を加えます。
DaysDiff(<日付1>, <日付2>)	日付1から日付2までの日数 (10進の浮動小数点数)
HoursDiff(<日付1>, <日付2>)	日付1から日付2までの時間数 (10進の浮動小数点数)
MinutesDiff(<日付1>, <日付2>)	日付1から日付2までの分数 (10進の浮動小数点数)
SecondsDiff(<日付1>, <日付2>)	日付1から日付2までの秒数 (10進の浮動小数点数)
DbToLocalDate(<日付>)	データベースサーバのタイムゾーンの日付を、クライアントマシンで定義したタイムゾーンの日付に変換します。
LocalToDbDate(<日付>)	クライアントマシンのタイムゾーンの日付を、データベースサーバのタイムゾーンの日付に変換します。

例

表 A.4. 日付型のAQL関数

説明	AssetCenterのクエリ言語
先週変更したすべてのレコード	AddDays(dtLastModif,7)>=Getdate()
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.5. 数値型のAQL関数

機能	説明
Abs(<数値>)	「数値」の絶対値を返します。
Ceil(<数値>)	「数値」以上で一番小さい整数を返します。
Floor(<数値>)	「数値」以下で一番大きい整数を返します。
Mod(<a>,)	「a」を「b」で割った余りを返します(a = qb + r、ただしqは整数で0 ≤ r < q)。
Round(<a>, <n>)	「a」を小数点以下第「n」位で四捨五入します。
Trunc(<a>, <n>)	「a」を小数点以下第「n」位で切り捨てます。

適用例

Abs (2.516) = 2.
 Ceil (2.516) = 3.
 Floor (2.516) = 2.
 Mod (6,4) = 2.
 Round (31.16, 1) = 31.20.
 Round (31.16, -1) = 30.00.
 Round (31.16, -1) = 30.00.
 Trunc (31.16, 1) = 31.1.

テスト型のAQL関数

表 A.6. テスト型のAQL関数

機能	説明
IsNull(<a>,)	「a」がNullの場合は「a」を「b」で置き換えます。「a」と「b」のデータ型に互換性がなければなりません。

AQLのシンタックス

ここでは、AQLのシンタックスについて詳しく説明します。内容は次の通りです。

- 表記法
- クエリのシンタックス
- FROM句：クエリの対象となるテーブル
- クエリの要素
- WHERE句
- GROUP BY句
- HAVING句
- ORDER BY句

 注意:

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.7. 表記法

[]	大括弧は、オプションの項目を示します。実際に項目を入力するときは、大括弧は必要ありません。
<>	山形括弧は、項目の短い説明を示します。実際に項目を入力するときは、山形括弧を使わずに、括弧内にあるテキストに該当する情報だけを入力してください。
	縦線（パイプ文字）は、複数の項目候補を区切るのに使います。
...	省略記号は直前の内容が繰り返されることを表します。
FROM	大文字の単語はそのまま記述します。

クエリのシンタックス

シンプルクエリ

```
SELECT [DISTINCT] [FIRST_ROWS] <選択リスト>
[FROM <句>]
[WHERE <句>]
[GROUP BY <句>]
[HAVING <句>]
[ORDER BY <句>]
```

サブクエリ

AQLでは、フィールドの代わりにサブクエリを使うことができます。

 **注意:**

サブクエリでは、SELECTステートメントで式を1個だけ使うことができます。

```
( SELECT [DISTINCT] <式>
  FROM <句> ]
  WHERE <句> ]
  GROUP BY <句> ]
  HAVING <句> ]
)
```

 **注意:**

注意：サブクエリは括弧で囲む必要があります。

使用例

```
SELECT Self FROM amAsset WHERE dPrice >= (SELECT Max(dPrice)/2 FROM amAsset)
```

UNION型のクエリ

UNIONを使うと、次のように複数のクエリの結果をグループ化できます。

```
SELECT <選択リスト>
  FROM <句> ]
  WHERE <句> ]
  GROUP BY <句> ]
  HAVING <句> ]
  [ UNION | UNION ALL | INTERSECTS | MINUS
  SELECT <選択リスト>
  FROM <句> ]
  WHERE <句> ]
  GROUP BY <句> ]
  HAVING <句> ]... ]
  ORDER BY <句> ]
```

FROM句：クエリの対象となるテーブル

FROM句には、SELECTステートメントの対象となるテーブルを指定します。

シンタックス

AQLでは、テーブル名のエイリアスを利用できます。

FROM句の書式は次の通りです。

FROM <テーブル> [<エイリアス>][, <テーブル> [<エイリアス>] ...]

例

```
FROM amAsset
FROM amAsset a, amLocation l
```

クエリの開始テーブル

クエリのFROM句で最初に指定したテーブルが、クエリの開始テーブルになります。

テーブルが指定されていないフィールドをクエリに使った場合は、AQLはそのフィールドがクエリの開始テーブルに属していると見なします。AQLのFROM句はSQLのFROM句とは異なります。

例えば、次のサブクエリでは、AQLは【資産】テーブル（SQL名：[amAsset]）の [AssetTag] フィールドを検索します。

```
SELECT AssetTag FROM amAsset, amLocation
```

使用例

次のクエリはすべて同じように機能します。

```
SELECT AssetTag FROM amAsset
SELECT AssetTag FROM amAsset a
SELECT a.AssetTag FROM amAsset a
SELECT amAsset.AssetTag FROM amAsset a
SELECT amAsset.AssetTag
SELECT amAsset:AssetTag
```

クエリ内のテーブル数

クエリで指定できるテーブル数は、使用しているDBMSによって異なります。

例

- Oracle：テーブルをいくつでも使うことができます。

- Microsoft SQL ServerまたはSybase SQL Server : クエリで使えるテーブル数は16個までです。

注意:

クエリで使用するテーブルの数を数える場合、暗黙的に言及されているテーブルを数に入れるのを忘れないでください。特にクエリがリンクを使用する時には注意を要します。データベースエンジンのSQLレベルで追加結合を生成する表記法「fv_」(任意管理項目の検索用)にも気をつける必要があります。表記法「cf_」(特殊〔集計〕フィールド)も、追加結合を生成できます。

クエリの要素

フィールドとリンク

クエリには、AssetCenterデータベース内のフィールドとリンクを指定します。フィールド名は次の方法で指定できます。

- クエリの開始テーブルの場合は、そのテーブルの名前を指定する必要はありません。

[Link. ...[Link.]]<field>

資産のテーブルの例

Brand
User.Name
User.Location.Name

- 参照フィールドとして指定する場合は、次のいずれかの方法でフィールドが属するテーブルの名前を指定する必要があります。

- FROM句でテーブルを宣言してテーブルの名前を指定します(エイリアスも可)。

<table.[link...]field>

<alias.[link...]field>

- FROM句でテーブルを宣言しない代わりにコロン(:)を使います。

<table:[link...]field>

<table[_alias]:[link[_alias]...]field>

後半の2つの表記は、FROM句が使えない場合に便利です。

例えば、AssetCenterでクエリを記述する場合は、WHERE句しか使えません。クエリの開始テーブルは暗黙的に指定します(フィルタを適用するテーブルをク

エリの詳細画面の【**テーブル**】（SQL名：TableName）フィールドで指定する場合は暗黙的な指定です）。これに対し、クエリで他のテーブルを使う必要がある場合は、コロン（:）を使って明示的に指定します。

定数

次のシンタックスは、クエリで使用できる有効な定数です。

数値定数

小数点の区切り文字としてピリオド（.）を使います。

例

1.2

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時間時計を使います（A.M.またはP.M.を付けた12時間時計ではありません）。
- 日付はスラッシュ（/）またはハイフン（-）で区切ります。
- 時刻はコロン（:）で区切ります。
- 月、日、時間、分、秒は、2桁で表します。
- 日付と時刻を合わせて表示する場合は、必ず日付の後に時刻を記述し、2つをスペースで区切ります。

例

#yyyy-mm-dd hh:mm:ss#

#yyyy-mm-dd#

#hh:mm:ss#

#1999-01-01 01:00:03#

式

式は、次の要素で構成されます。

- 定数
- フィールド
- 関数
- サブクエリ

これらの要素に演算子や括弧を組み合わせて、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.8. 論理演算子

演算子	意味
AND	論理積演算
OR	論理和演算

クエリを最適化するために、比較演算子を使える場合は論理演算子を使わない方がよい場合があります。次の例は、【割当】（SQL名：seAssignment）フィールドが【納入待ち】または【メンテナンスのため返却】の資産を選択するクエリフィルタを最適化する方法です。この2つのシステムリストデータの値は、それぞれ「3」と「4」です。

```
(seAssignment=3) OR (seAssignment =4)
```

システムリストデータの値が「4」までの場合は、上のクエリを次のように記述することもできます。

```
seAssignment >=3
```


比較演算子

比較演算子は、2つの式を比較するときに使います。

表 A.9. 比較演算子

演算子	意味
=	等しい
<>	等しくない
=!	
>	より大きい
<	より小さい
>=	以上
=<	以下
=*	右外部結合。この演算子は、AQLでリンクを処理する場合のみ使います。
*=	左外部結合。この演算子は、AQLでリンクを処理する場合のみ使います。
LIKE	=演算子と同様に機能し、ワイルドカード文字を使うこともできます。
NOT LIKE	次のワイルドカード文字を使用できます。 パーセント (%) : すべての文字列に置き換わります。 アンダースコア (_) : 任意の1文字に置き換わります。 次の指定は、データベースエンジンによって変わります (SQL Anywhere、SQL Server、Sybaseはサポートしますが、Oracleはサポートしません)。 [abc...]は任意の並び文字 (間にスペースがない) を定義します。 [a-c]は、文字列値の範囲を定義します。 DB2では、LIKE X演算子のXにSQL列名が含まれる場合は、この演算子を使うことはできません。この演算子で使用できるのは定数のみです。例えば、次のようなクエリはDB2で機能しません。 SELECT COL1, COL2 FROM TABLE1 WHERE COL1 LIKE COL2

演算子

IS NULL

IS NOT NULL

意味

フィールド値がNULLかどうかを調べます。

重要：AssetCenterでは、空のテキストフィールドと、値が入力されていない日付または日付+時刻型フィールドがNULLと見なされます。

**注意:**

SQL Anywhereでは、LIKE X句のXが128文字を超える場合は処理できません。Xが128文字を超えた場合にクエリを適用すると、ODBCエラーメッセージが表示されます。このエラーは、LIKE句で [FullName] フィールドを指定して、リストをツリー構造で表示する場合などに発生する可能性があります。

サブクエリ専用の演算子

次の演算子を使って、サブクエリの結果と値を比較できます。

- = ANY (サブクエリ)
- = ALL (サブクエリ)
- = SOME (サブクエリ)

例

- 次のクエリでは、「府中支社」で使用しているメーカーの資産のリストを取得できます。

```
SELECT IAstId, Brand FROM amAsset WHERE Brand = ANY (SELECT Brand FROM amAsset 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。

```
SELECT MrMrs, (Name + FirstName) Identity FROM amEmplDept
```

選択リストは、出力クエリで出力する列名を指定する場合に特に便利です。

 **注意:**

DBMSによっては、1つのSELECTステートメントに特定の数の式しか記述できない場合があります。

WHERE句

AQLのWHERE句は、SQLのWHERE句と同様に機能します。

WHERE句は、検索条件を指定するときに使います。

WHERE **<検索条件>**

検索条件には、データベースから抽出する項目を指定します。検索条件にもWHEREまたはHAVING句を使うことができます。

多くの場合は、次の書式で条件を記述する必要があります。

```
<WHERE | HAVING> [NOT] <式> <比較演算子> <式>
<WHERE | HAVING> [NOT] <論理式>
<WHERE | HAVING> [NOT] <フィールド> [NOT] LIKE 'xxxxx'
<WHERE | HAVING> [NOT] <論理式> <AND | OR> <論理式>
<WHERE | HAVING> [NOT] <フィールド> IS [NOT] NULL
```

場合によっては、次のように複雑なクエリを記述する必要があります。

```
<WHERE | HAVING> [NOT] EXISTS (<サブクエリ>)
<WHERE | HAVING> [NOT] <式> [NOT] IN (<値のリスト> | <サブクエリ>)
<WHERE | HAVING> [NOT] <式> <比較演算子> <ANY | ALL> (<サブクエリ>)
```

GROUP BY句

AQLのGROUP BY句は、SQLのGROUP BY句と同等に機能します。

GROUP BY **<集合を使わない式>**

[, **<集合を使わない式>**]...

GROUP BY句はテーブルのサブセットを指定するときに使います。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、GROUPBY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GROUPBY句を使うと、選択リストの各式でサブセットごとに1つの値が得られます。

GROUP BY句を使った例と使わない例は以下の通りです。

次のクエリは、データベース内のメーカーの総数を算出します。このクエリの場合、AssetCenterは、1つのメーカーに関連付けられている資産ごとに、1つのメーカーインスタンスを返します。

```
SELECT Count(Brand) FROM amAsset
```

次のようにGROUP BY句を使うと、メーカーのリストと各メーカーの資産数が算出されます。

```
SELECT Brand, count(IAssetId) FROM amAsset GROUP BY Brand
```

HAVING句

AQLのHAVING句は、SQLのHAVING句と同等に機能します。

HAVING <検索条件>

HAVING句は、WHERE句と同様に検索条件を指定するときに使います。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。

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 Brand, count(IAssetId) FROM amAsset GROUP BY Brand HAVING Brand > 'B'
```

次のWHERE句を使ったクエリでも同じ結果が得られます。

```
SELECT Brand, count(IAssetId) FROM amAsset WHERE Brand > 'B' GROUP BY Brand
```

HAVING句を使ったクエリの例は以下の通りです。

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 Brand, count(IAssetId) FROM amAsset GROUP BY Brand HAVING count(Brand) > 1
```

ORDER BY句

AQLのORDER BY句は、SQLのORDER BY句と同等に機能します。

ORDER BY <式> [ASC | DESC] [,<式> [ASC | DESC]...]

項目の並べ替え順には、次のタイプがあります。

- 昇順：ASC。デフォルトの並べ替え順です。
- 降順：DESC

B | 問題点の報告方法

付録

弊社製品の品質を常に改善するためには、ユーザであるお客様に、製品の使用時に発見された問題点をペレグリンシステムズの顧客サポートに報告していただくことが非常に大切です。問題点が迅速に識別され解決されるためには、問題点に関する詳細な情報が必要になります。しかし弊社に提供される情報が不完全なために、問題を再現できないことも多々あり、よって、問題点の解決も不可能なことがあります。

本章では、製品の問題点をより適切に顧客サービスに報告するために、どの情報が不可欠であるかを説明します。

一般情報

顧客サービスに必ず報告しなければならない情報は以下の通りです。

- 機能に関する情報
- 技術的な情報

機能に関する情報

以下の情報により、問題点の分類が可能になります。

- お客様の会社名
- 問題が発生する製品名
- 問題が発生する機能・場所（例えば、Connect-Itのシナリオビルダ、AssetCenterの調達管理、など）
- 問題が発生する頻度（問題は毎回発生するか、または問題の発生はある一定の要因やデータにより左右されるのか、など）

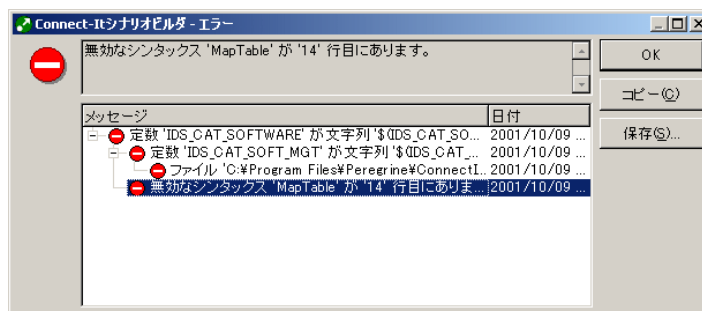
技術的な情報

重要な技術的な情報の一部は、以下の手順に従って製品内で得ることができます。

- 1 [ヘルプ/バージョン情報]メニューを選択します。
- 2 表示されるダイアログボックス内で[詳細]をクリックします。システムと製品に関する技術的な情報が表示されます。
- 3 [コピー]ボタンをクリックします。情報はWindowsのクリップボードにコピーされます。
- 4 新規ファイルにこの情報を貼り付け、ペレグリンシステムズの顧客サポートまで御送付ください。

問題発生時にアプリケーション内でエラーメッセージが表示される場合は、以下の手順に従って問題点を報告してください。

- 1 エラーメッセージが発生したら[コピー]ボタンをクリックします。これにより、情報はWindowsのクリップボード内にコピーされます。



注意:

上記の画面は例に過ぎません。エラーメッセージの内容やソフトウェア名は、発生する問題により異なります。

- 2 新規ファイルにこの情報を貼り付け、ペレグリンシステムズの顧客サポートまで御送付ください。

問題固有の情報

弊社のサポートが問題を再現するためには、まず、問題発生に至るまでの状況と過程の詳細なリストが必要になります。問題の発生は様々な要因に左右されるため、例えば以下の様に詳細に説明してください。

- 1 資産「Peregrine Prosigma 850」をAssetCenterの資産のテーブルで選択します。この資産は、製品付属のデモ用データベースに含まれているものです。
- 2 この資産の詳細画面で【取得】タブをクリックします。
- 3 【取得】タブの【調達】サブタブページをクリックします。
- 4 【市場価格】フィールド値を123.45に変更します。
- 5 【変更】をクリックします。
- 6 ...

ヒント:

原則として、問題発生の状況はできる限り詳細に記述してください。

その他の情報

問題のより適切な識別のために、スクリーンショットやその他のファイルをサポートに送付することも可能です。

注意:

Eメールで大型サイズのファイルを送るのはお控えください。添付ファイルのサイズは500 KBまでに制限してください。必要であれば、ペレグリンのサポートは、500 KB以上のファイルを受信するためにFTPサイトを開くことも可能です。

送信するファイルのサイズを縮小するには、

- 圧縮ツールを使用します。多くの場合、ファイルのサイズが大幅に縮小されます。
- 画像の場合は、色数を16に制限します。これで画像のサイズは大幅に縮小されます。

Connect-Itに関する問題を報告する場合

Connect-Itはドキュメントログ、アプリケーションログと、サービスログにデータを常時記録しています。これらのファイル内の情報は、問題の再現に非常に有用です。

これらのファイルは以下のフォルダに格納されています。

- bin32\conitgui.log : アプリケーションログ
- bin32\<サービス名>.log : Connect-Itサービスログ
- 「.idx」、「.dat」と「.msg」ファイル : シナリオに関連するログファイル。ファイル名は、Connect-Itのグラフィカルインターフェイス内のシナリオを使って設定できます。

 **注意:**

Connect-Itの問題点を報告する際には、上記のファイルとシナリオファイル（「.scn」）をサポートまで御送付ください。

C | 用語解説

付録

この用語解説では、Connect-Itで使用される主要用語の一部を説明します。また、データベース言語、Basicスクリプト、Java、外部アプリケーションで使用される主要用語も列挙されています。

Connect-It用語

コネクタ

コネクタにより、Connect-Itは、外部アプリケーションからまたは外部アプリケーションへデータを処理することができます。コネクタ間でデータを交換するために、コネクタはデータをドキュメントに転記し、ドキュメントの生成または取り込みを実行します。

コネクタは数種に分類されます。

- 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。

インベントリコネクタは、企業の資産、特にIT資産をスキャン（走査）するデータベースのデータを処理します。コネクタにはIntel LanDeskコネクタ、SMS 1.xコネクタとSMS 2.xコネクタなどがあります。

- **Peregrine Systemsコネクタ**
Peregrine Systemsコネクタは、ペレグリンシステムズアプリケーション用のコネクタです。Asset Managementコネクタ、ServiceCenterコネクタ、InfraTools Managementコネクタ、InfraTools Desktop Discoveryコネクタ、Action Request Systemコネクタなどがあります。
- **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、外部のデータアプリケーション用のコネクタです。Lotus Notesコネクタ、NTセキュリティコネクタなどがあります。
- **プロトコルコネクタ**
プロトコルコネクタは、特定のプロトコルを用いて外部アプリケーションによりフォーマットされたデータを処理します。XMLコネクタ、テキストコネクタ、データベースコネクタがあります。
- **Javaコネクタ**
JavaコネクタはJavaで開発されています。Connect-ItのJava開発キットを使用すると、ユーザは独自のコネクタを開発できます。
Javaコネクタ作成の詳細については、「[Connect-ItのJava開発キット \(JDK\)](#)」を参照してください。

コネクタの役割は以下の通りです。

- コネクタは、コネクタの接続先のアプリケーションで使用可能なデータ集合に対応する、使用可能なドキュメントタイプを発行します。

例：

Eメールコネクタ（受信）が使用可能なドキュメントタイプを発行する場合、この使用可能なドキュメントタイプの内容は、From（送信者）構造体、Carbon Copy（CC）コレクションとAttachment（添付ファイル）コレクションなど、Eメール内の全データに相当します。

- **ドキュメントの生成と取り込み**
ドキュメントを生成するために、コネクタは、ある特定の統合シナリオ用に作成された「生成用ドキュメントタイプ」テンプレートを使用します。コネクタはこのテンプレートに基づいてドキュメントを生成します。

例：

既製のInfraTools Desktop Discovery - Asset Managementシナリオで、Desktop Discoveryコネクタは、「Machine」という生成用ドキュメントタイプをテンプレートとして使用してドキュメントを生成します。この生成用ドキュメントタイプは、InfraTools Desktop DiscoveryアプリケーションがIT資産をスキャンする際に取得する全データの部分集合に当たります。

ドキュメントを取り込むために、コネクタは、マッピングボックスから送られてくるドキュメントの値を抽出し、外部アプリケーション用にデータ値を転換します。

データが、外部ソースアプリケーションから外部ターゲットアプリケーションに転送される場合、コネクタは、ソースコネクタとターゲットコネクタと呼ばれます。ソースコネクタはドキュメントを生成し、ターゲットコネクタはドキュメントを取り込みます。

ドキュメントとドキュメントタイプ

ドキュメントはConnect-It内でデータを伝達する媒体です。各ドキュメントは、外部アプリケーションの1つのデータ集合に対応します。外部アプリケーションの機能に応じて、1データ集合（データコンテナ）は以下の内容になります。

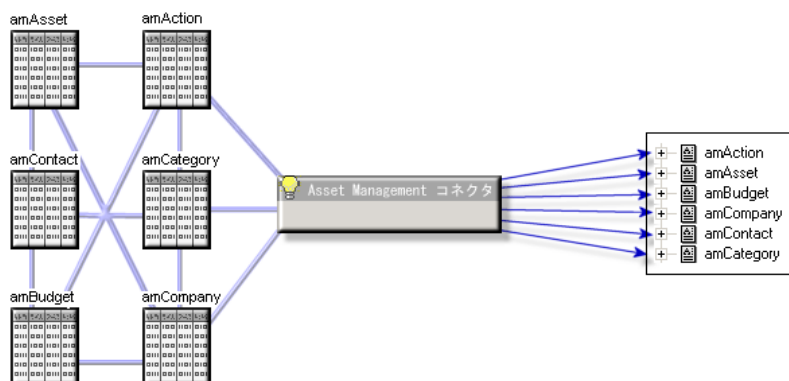
- 1つのデータベーステーブル
- 1通のEメールメッセージ
- インベントリ情報を含む1つのファイル
- 区切り文字で区切られた1つのテキストファイル
- 1つのXMLファイル
- セキュリティ情報
- その他

シナリオビルダでコネクタが開かれると、コネクタは使用可能な全てのドキュメントタイプのリストを発行します。使用可能な全ドキュメントタイプのリストは、外部アプリケーションで使用可能なデータ集合全体を指します。

例：

AssetCenterデータベースの全テーブルは、シナリオビルダ内のAsset Managementコネクタが発行する全ての使用可能なドキュメントタイプ群に一致します。

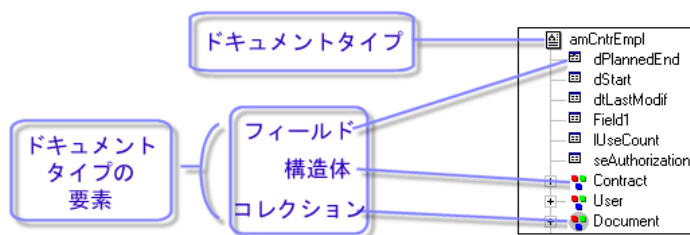
図 C.1. Asset Managementコネクタ - 使用可能なドキュメントタイプ群



ドキュメントタイプの要素

各ドキュメントタイプはツリー構造になっています。ツリー構造内には、単なる自立型要素（フィールド）と、複数レベルの構造を持つ複雑な要素があります。後者には更に、ルートノードや、単純または複雑な要素を含む構造体やコレクションが含まれています。

図 C.2. ドキュメントタイプの要素



単純要素には以下のものがあります。

- フィールド (■)
フィールドは、数、倍長整数、固定長テキスト、日付、などの特定のデータ型を含みます。
- BLOB型フィールド (■)
BLOB(binary large object)型フィールドは、画像、音声、ビデオなどの保存される必要のあるバイナリオブジェクトを含んでいます。
- メモ型フィールド (■)
メモ型フィールドは可変な長さのテキストを含んでいます。

複雑な要素には以下のものがあります。

- 単純要素または複雑要素を含む構造体 (■)
例1：データベーステーブル用のドキュメントタイプでは、各構造体は別のテーブルを指しています。構造体は、2つのテーブル間のリンクが1対1であることを示しています。つまり1番目のテーブルのレコード1つは、別のテーブルのレコード1つにのみリンクされています。
例2：受信されたEメールメッセージ用のドキュメントタイプでは、構造体はメッセージの送信者に関する情報に当たります（1つのメッセージには1人の送信人しかいません）。
- 単純要素または複雑要素を含むコレクション (■)
例1：データベーステーブル用のドキュメントタイプ内にコレクションがある場合、このコレクションはデータベース内の別のテーブルを指しています。コレクションでは、2つのテーブル間のリンクは1対N、またはN対Nです。1番目のテーブルの1つのレコードは、2番目のテーブルの1つまたは複数のレコードにリンクされており、2番目のテーブルのレコードも、1番目のテーブルの1つまたは複数のレコードにリンクされています。
例2：送信されたEメールメッセージ用のドキュメントタイプでは、コレクションはメッセージの受信者に当たります（1つのメッセージの受信者数は、何人になってもなり得ます）。

注意:

属性のコレクションは、1つのフィールドのみを含むコレクションです。フィールドは属性に当たります。コレクション内の各構成要素は、1つの要素がN回繰り返された構造体です。

コネクタがドキュメントを処理する前に、ドキュメント内の各コレクションには不定数の構成要素があります。これらの構成要素の正確な数を処理前に知ることはできません。そのため、コレクションはN個の構成要素から成っていると定義します。

図 C.3. ドキュメントタイプのコレクション



例 :

例 : Desktop Discoveryコネクタが発行する使用可能なドキュメントタイプ「Machine」では、コンピュータにインストールされたソフトウェアはコレクションで示されています。これは、各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数は不定なため、この種のデータはコレクションの形を取る必要があるからです。コレクションの各構成要素はソフトウェアの内容を説明しています（名前、ソフトウェア会社、バージョンなど）。

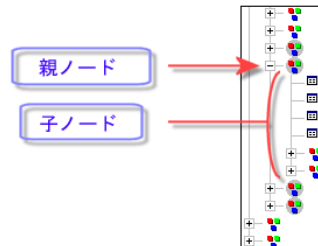
ドキュメントタイプ内の親ノードと子ノード

ツリー構造内では、階層を持つ要素下には子要素があります。ドキュメントタイプでは、これを親ノードと子ノードと呼びます。

 注意:

親ノードと子ノードの概念はドキュメントの処理中に重要です。例えば、子ノード内のフィールドが拒否された結果、親ノードも拒否されることがあります。

図 C.4. ドキュメントタイプのツリー構造



生成用ドキュメントタイプと取り込み用ドキュメントタイプ

コネクタがドキュメントを生成し取り込むために、コネクタは、生成用ドキュメントタイプと取り込み用ドキュメントタイプを使用する必要があります。生成用ドキュメントタイプ、または取り込み用ドキュメントタイプを作成する場合、Connect-Itのユーザは、コネクタが発行する使用可能なドキュメントタイプから要素を選択します。例えば、データベース型コネクタの生成用ドキュメントタイプには、外部ターゲットアプリケーションにインポートする必要のあるフィールドのみを選択します。

使用可能なドキュメントタイプ	生成用または取り込み用ドキュメントタイプ	生成されるドキュメント
----------------	----------------------	-------------

amAsset	
AcctCode	テキスト
AssetTag	テキスト
Bar Code	テキスト
bCreatedOnThe...	整数(16ビット)
bIpxSpxInstalled	整数(16ビット)
bIsCnxClient	整数(16ビット)
bNetBeuInstalled	整数(16ビット)
Brand	テキスト
bTcplpInstalled	整数(16ビット)
Comment	メモフィールド

amAsset	
AssetTag	
Brand	
mPrice	
Category	
Name	

amAsset		Hewlett Packard
AssetTag		PRNLND1006
Brand		Hewlett Packard
Category		C000007
BaiCode		C000007
binvent		1

コネクタは使用可能なドキュメントタイプを発行します。上記の図は、Asset Managementコネクタに発行される使用可能なドキュメントタイプ「amAsset」の抜粋です。

生成用、または取り込み用ドキュメントタイプを作成するには、使用可能なドキュメントタイプから必要な要素を選択します。

コネクタは、生成用または取り込み用ドキュメントタイプに沿って、ドキュメントの生成または取り込みを実行します。

XMLファイルとDTDファイル

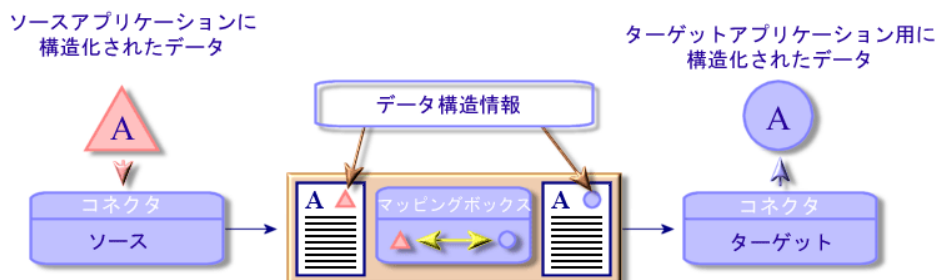
Connect-It内で使用されるドキュメントはXML (Extensive Markup Language) ファイルに相当し、ドキュメントが使用するドキュメントタイプは、DTD (Document Type Definition、文書型定義) に相当します。

Connect-Itのシナリオビルダでは、全てのドキュメントをXMLファイルに転換し、全てのドキュメントタイプをDTDに変換することができます。

マッピング

マッピングは、ソースコネクタの生成用ドキュメントタイプの要素群と、ターゲットコネクタの取り込み用ドキュメントタイプの要素群を一致させ、データを変換します。

Connect-Itのマッピングボックスにより、ターゲットコネクタは、ソースコネクタが生成したドキュメントを取り込むことができます。マッピングボックスは内部ツールで、他のコネクタのようにドキュメントの生成や取り込みを行いますが、外部アプリケーションには接続しません。



統合シナリオ

統合シナリオにより、異種アプリケーション間でデータを伝達できるようになっています。

シナリオを作成するには以下の手順に従います。

- 1 ソースコネクタとターゲットコネクタを選択します。
原則として、使用するコネクタの数に制限はありません。
- 2 コネクタを設定します。

- 3 コネクタをリンクします。
- 4 ソースコネクタでは、生成用ドキュメントタイプを作成し、ターゲットコネクタでは取り込み用ドキュメントタイプを作成します。
- 5 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間のマッピングを作成します。
- 6 スケジュールを選択します。
スケジュールは、シナリオのソースコネクタがドキュメントを生成する頻度を設定します。

Connect-Itは用例シナリオと共に提供されています。



警告:

用例シナリオは、多くの場合ユーザの特定の事例には当てはまりません。

ルール (ディレクティブ)

ルール (ディレクティブ) は、Connect-Itのコンポーネントがドキュメントを生成し、取り込むために使用する情報のことです。これには**生成用ルール**と**取り込み用ルール**があります。

ドキュメントログ

シナリオビルダ内のタブにあるドキュメントログでは、Connect-Itのコネクタやその他のコンポーネント (リンクやマッピングボックスなど) が生成した、または取り込んだ全ドキュメントの内容を確認できます。

コネクタがドキュメントを拒否すると、ドキュメントログは拒否されたドキュメントをアイコンと共に表示します。ドキュメントログは更に、コネクタまたはマッピングボックスが不完全にドキュメントを生成したり、取り込んだりした場合にも警告を發します。

スケジュール

Connect-Itのスケジュールでは、統合シナリオのソースコネクタがいつドキュメントを生成するのかを設定できます。スケジュールを作成するには、ドキュメントを1つまたは複数のスケジューラに関連付けます。

スケジューラはタイマの役割を果たしており、コネクタが特定の日時に (例: 2002年10月4日)、または定期的に (例: 一定の期間内で2時間おき) ドキュメントを生成するように設定します。

処理レポート

処理レポートは、コネクタやマッピングボックスがドキュメントを処理するたびに生成するドキュメントのことです。このレポートによりドキュメントの処理に成功したか、または問題が起こったかどうかを確認できます。更に、別のコネクタがこの処理レポートを取り込むと、Connect-Itで発生した問題について警告を出すようにも設定できます。

例：

Eメールコネクタはシナリオの処理レポートを取り込み、メッセージの形でConnect-Itの管理者へ処理レポートを送信することができます。

認証証明書

認証証明書は、ベースコネクタと、選択したConnect-Itパッケージに含まれているオプションコネクタと追加コネクタを、使用可能にするテキストファイルです。

外部アプリケーション

外部アプリケーションは、コネクタの接続先であるアプリケーションやデータソースを指します。例えば、AssetCenterアプリケーション、XMLファイル、メッセージシステムなどです。

Blob

Blob (Binary Large Object) タイプのフィールドはバイナリデータ (サウンド、ビデオ、画像ファイル) のフィールドです。

キャッシュ

コネクタが使用するキャッシュファイルは、コネクタが発行する使用可能なドキュメントタイプの記述を含みます。キャッシュファイルの拡張子は「CCH」です。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。

コレクション

ドキュメントタイプ内で、コレクションは1つまたは複数の要素と、別の複数の要素間の関係を表しています。

例：データベーステーブル用のドキュメントタイプ内にコレクションがある場合、このコレクションはデータベース内の別のテーブルを指しています。コレクションでは、2つのテーブル間のリンクは1対N、またはN対Nです。1番目のテーブルの1つのレコードは、2番目のテーブルの1つまたは複数のレコードにリンクされており、2番目のテーブルのレコードも、1番目のテーブルの1つまたは複数のレコードにリンクされています。

例：送信されたEメールメッセージ用のドキュメントタイプでは、コレクションはメッセージの受信者に当たります（1つのメッセージの受信者数は、何人にもなり得ます）。

コンポーネント

コンポーネントは、コネクタまたは内部ツール（マッピングボックス）に当たります。

サービスコンソール

グラフィカルインターフェイスであるConnect-Itコンソールでは、シナリオを管理し、シナリオに関連付けるサービスを作成することができます。

取り込み

取り込みとは、ターゲットコネクタが外部アプリケーションにドキュメントを書き込んだり送信したりするプロセスを指します。

DAT

拡張子「DAT」を含むファイルは、シナリオに処理されるドキュメントに相当します。

これらのファイルは、ドキュメントログの設定用ウィンドウ（**[ログ / ドキュメントログを設定する]**）で、**[ファイルを使用]** オプションを選択した場合にのみ生成されます。

DBK

DBKファイルは、シナリオ文書に対応するXMLファイルです。これらのファイルはDocBook DTDに準拠しています。DBKファイルの詳細については、「シナリオ文書」の章を参照してください。

ルール (ディレクティブ)

コネクタのルール (ディレクティブ) は、コネクタがドキュメントの生成または取り込み時に従う指示を指します。

例：データベース型のコネクタでは、ソースデータベースにフィルタを適用するWHERE句とORDERBY句は、取り込み用ルールに当たります。

ルールの詳細については、「コネクタのルール (ディレクティブ)」の章を参照してください。

コネクタの導入

Javaコネクタをシナリオビルダ内に導入するためには、以下の内容を指定します。

- 導入ファイル
- コネクタがドキュメントを生成するか、または取り込むかどうか
- ツールボックス内に表示されるコネクタ用のアイコン
- 数種のオプション

コネクタの導入の詳細については、「Connect-ItのJava開発キット (JDK)」の章の「イベントコネクタを作成する」の節、「Javaコネクタの導入 - イベントコネクタ」を参照してください。

シナリオビルダ

シナリオビルダは、統合シナリオ作成用のグラフィカルインタフェースです。シナリオは、ファイルの拡張子「SCN」で識別されます。

要素

ドキュメントタイプのツリー構造には以下の要素が含まれます。

- 構造体
- コレクション

- フィールド
この要素は、ドキュメントタイプの端末要素に当たります。

ユーザフォーマット

ユーザフォーマットとは、ユーザがマッピングスクリプト内で使用するために定義する日付型または数値のフォーマットを指します。フォーマットは、PifUserFmtVarToStr関数とPifUserFmtStrToVar関数と共にのみ使用されます。使用に関する詳細は、オンラインの『プログラム用参考ガイド』（Connect-Itでのスクリプト作成中に「F1」キーを押すと表示されます）を参照してください。ユーザフォーマットの詳細については、「ドキュメントタイプのマッピング」の章の「スクリプトの作成」の節、「ユーザフォーマット」を参照してください。

IDX

IDXファイルは、ドキュメントログ内のDATファイルとLOGファイルのデータを、迅速に検索するためのインデックスファイルです。

これらのファイルは、ドキュメントログの設定用ウィンドウ（**[ログ / ドキュメントログを設定する]**）で、**[ファイルを使用]** オプションを選択した場合にのみ生成されます。

MSG

MSGファイルは、ドキュメントログ内にあるメッセージを含みます。

これらのファイルは、ドキュメントログの設定用ウィンドウ（**[ログ / ドキュメントログを設定する]**）で、**[ファイルを使用]** オプションを選択した場合にのみ生成されます。

スケジュールのポインタ

スケジュールのポインタを使用すると、シナリオが処理するデータの量を少なくすることができます。

ポインタがレコードの最終変更日である場合、コネクタは最後の起動以降に作成または更新されたレコードのみを処理します。

シナリオがスケジュールモードで起動していると、生成用ドキュメントタイプのポイン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。

生成

生成とは、データをドキュメントに転換するために、ソースコネクタが外部アプリケーションからデータを読み取ったり受信したりするプロセスのことを指します。

スケジューラ

スケジューラは、シナリオのソースコネクタの1つまたは複数の生成用ドキュメントタイプに関連付けられたタイマです。

シナリオのスケジューラを編集するには、[シナリオ/スケジューラ]を選択します。

シナリオビルダには2つの既製スケジューラが付属しています。

- 「一回」スケジューラ
このスケジューラは、シナリオのすべての生成用ドキュメントタイプにデフォルトで関連付けられています。
- 「同期」スケジューラ
このスケジューラにより、コネクタは毎秒ドキュメントを生成します。

実行結果

各セッション後に、シナリオビルダは、シナリオの各コンポーネントが処理したドキュメントの数に関する実行結果を取得します。例えば、Asset Managementコネクタが1セッション中に処理したドキュメントの数などです。

各コネクタの実行結果を表示するには、シナリオ図内のコネクタの上部にカーソルを置きます。

構造体

ドキュメントタイプ内の構造体は、この構造体を含むノードと1対1の関係であることを表しています。構造体を含むノードは、ドキュメントタイプのルートノード、構造体またはコレクションです。

例：データベーステーブル用のドキュメントタイプでは、各構造体は別のテーブルを指しています。構造体は、2つのテーブル間のリンクが1対1であることを示しています。つまり1番目のテーブルのレコード1つは、別のテーブルのレコード1つにのみリンクされています。

処理後のアクション

一部のコネクタ（[InfraTools Desktop Discoveryコネクタ](#)など）は、自分が生成したドキュメントに更に操作を加えることができます。この場合、コネクタの設定ウィザードのページで、生成されるドキュメントの「処理後のアクション」を選択します。例えば、Connect-Itで処理に成功した後にソースデータを削除するなどのアクションがあります。

ピボットドキュメントタイプ

ピボットドキュメントタイプは、構成要素（資産、ソフトウェア、従業員など）の一般的な表現に当たります。ソースコネクタは、自分のドキュメントタイプとピボットドキュメントタイプを一致させます。

ピボットドキュメントタイプは、マッピングなしで、ソースコネクタとターゲットコネクタ間でデータを転送できるようにします。

ログビューア

ログビューアでは、Windowsのサービスとしてシナリオが起動するたびに作成されるLOGファイルを読むことができます。

ビュー

シナリオ図のビューは、シナリオ図を見やすくするために使用されます。例えば、多数のコンポーネントを含むシナリオでは、1つまたは一部のコンポーネントのみを表示するビューを作成できます。

ビューの詳細については、「[シナリオビルダ](#)」の章の「[シナリオ図のビュー](#)」の節を参照してください。

DSC

書式設定ファイル（ファイルの拡張子「.dsc」）は、テキストファイル内のデータがどのように構成されているかを説明するファイルです。ファイル内のデータは、データベースのテーブル内のフィールド値に当たります。

テキストコネクタの使用可能なドキュメントタイプは、書式設定ファイル内で作成されたドキュメントタイプです。

例：ある書式設定ファイルでは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）のデータを含みます。
- 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。

主要用語

AQL

AQL（Advanced Query Language）は、AssetCenterデータベースのデータにアクセスするために、AssetCenterが使用する言語です。これはSQLに匹敵します。クエリの実行時に、AQLはデータベースエンジンのSQL言語に自動的に変換されます。

DSE

DSE（Directory Entry Service）は、LDAPディレクトリのツリー構造を構成するエン트리です。これらの情報は、ブランチ形式で表現されます。

LDAPディレクトリの各エント리는、抽象オブジェクトまたは実際のオブジェクト（例えば、人、物品、パラメータなど）に対応します。

FSF

拡張子が「.fsf」（Fingerprint Save File）であるファイルは、あるコンピュータに関する全データを含みます。

「.fsf」ファイルは、InfraTools Desktop Discovery型のスキャンで生成されます。スキャン時に収集され「.fsf」ファイルに格納された情報は、コンピュータに関するデータを更新するために（整合性チェックの段階などで）分析されます。

JCA

Sun MicrosystemsのJCA (Java Connector Architecture)は、インタフェースを説明する仕様全体をまとめたものです。

JCAの主な目的は、異なる技術に基づいたアプリケーションとJavaサーバが対話できるようにする規格の層を定義することです。

JCAは、接続インタフェースや標準コネクタの調整を簡易化します。

整合性チェック

整合性チェックとは、別アプリケーションから受信するデータの統合であり、ソースアプリケーションから来るデータの方が、ターゲットアプリケーション内の既存のデータよりも新しいと見なされています。

- 既存しない場合データは挿入されます。
- 既存する場合、受信するデータの新規情報に応じて既存のデータは更新されます。

このプロセスは、受信するデータが、ターゲットアプリケーション内に既存するかどうかに応じて実行されます。

ターゲットアプリケーションが空である場合は、このプロセスは整合性チェックではなく「インポート」になります。

XMLスキーマ

XMLスキーマとは、XMLドキュメント内で使用される内容の定義です。XMLスキーマは、SGML規格のスキーマであるDTDのスーパーセット（上位集合）に当たります。

DTDとは違って、XMLスキーマはXMLシンタックスで書かれています。このシンタックスはDTDよりも拡張されています。

XMLスキーマは、全XMLツールで作成可能です。

マップテーブル

マップテーブルは、ある集合の要素を別の集合の要素に対応させる表です。

トランザクション

トランザクションとは、コンピュータとユーザ間または2台のコンピュータ間の対話における1要素です。例えば、情報の要求とそれに対する応答などを指します。

グローバル変数

グローバル変数の値は、変数を定義するモジュールだけでなく、プログラムの全命令がアクセスでき、変更できる値です。

XSL

XSL (eXtensible StyleSheet Language) は、XML専用開発された拡張可能なスタイルシート言語です。これは公式な規格とは見なされていません。

索引

目次

キャッシュ, 556, 144
グローバル変数, 564, 128
コネクタの導入, 558
サービスコンソール, 557, 168
シナリオビルダ, 558, 52
スケジューラ, 560, 158
ドキュメントタイプ, 549, 86
トランザクション, 564, 257
ピボットドキュメントタイプ, 561, 137
ビュー, 561, 50
フォーマット, 559, 128
マップテーブル, 563, 126
ルール (ディレクティブ), 558, 555, 261
ログビューア, 561, 37
外部アプリケーション, 556, 237
構造体, 560, 95
実行結果, 560, 174
取り込み, 557, 182
処理後のアクション, 561, 403, 313
整合性チェック, 563, 265
生成, 560, 176
認証証明書, 556, 42
要素, 558, 112

A

AQL, 562, 541, 521

B

Blob, 556, 114

D

DAT, 557, 36
DBK, 558, 155
DSC, 562, 413
DSE, 562, 386
DTD, 554

F

FSF, 562, 289

I

IDX, 559, 36

J

JCA, 563, 196

M

MSG, 559, 36

X

XML, 554
XMLスキーマ, 563
XSL, 564, 153



August 8, 2002