

Peregrine

Connect-It



Connectors

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services.

Nothing herein should be construed as constituting an additional warranty.

HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software.

Valid license from HP required for possession, use or copying.

Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyrights

© Copyright 1994-2006 Hewlett-Packard Development Company, L.P.

Brands

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

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



Table of Contents

Introduction	11
What do the connectors do?	11
Who are the connectors intended for?	12
How to use this guide	12
Compatibility of the connectors	13
Chapter 1. Configuring connectors	15
Configuring a connector	16
Name and describe the connector	17
Select a connection type	18
Configuring the connection	19
Select files or folders	27
Advanced configuration (advanced mode)	29
Configure the reconnection parameters (advanced mode)	31
Determine server delay (advanced mode)	33
Configure schedule pointers (advanced mode)	36
Manage transactions (advanced mode)	38
Configure the cache (advanced mode)	39
Advanced options (advanced mode)	40
Define document types (advanced mode)	42
Configure the JVM	43
Parallelize consumption (advanced mode)	44
Chapter 2. Connector directives	47

Production directives	48
Consumption directives	51
Reconciliation	52
 Chapter 3. Peregrine Systems connectors	 63
Asset Management connector	63
Desktop Administration 6.x connector	76
Enterprise Discovery connector	82
Peregrine Desktop Inventory 7.x (PDI) connector	85
Peregrine Desktop Inventory 8.x connector (PDI)	90
Peregrine Network Discovery connector	95
BizDoc connector	98
SCAuto listening connector	103
RDS connector	108
ServiceCenter connector	112
ServiceCenter Web Service connector	130
 Chapter 4. Application connectors	 133
Action Request System connector	133
Lotus Notes connector	140
MQSeries connector	150
NT Security connector	173
Mercury Connectors	177
Tivoli Enterprise Console connector (sending)	177
Tivoli Enterprise Console connector (receiving)	182
 Chapter 5. Protocol connectors	 193
Database connector	193
XML listening connector	204
E-mail connectors	212
LDAP connector	228
Command-line connector	242
Delimited-text connector	246
Web Service connector	269
XML connector	275
 Chapter 6. Inventory connectors	 301
Altiris connector	301
Asset Insight connector	304
LANDesk for Inventory 8 Connectors	309
LANDesk Software Distribution connector	312
System Management System (SMS) connectors	315
iInventory connector	319

Tivoli Inventory connector (version 4.0)	323
Tivoli CM for Software Distribution 4.2 connector	326
Tivoli CM for Inventory 4.2 connector	327
Tivoli CM for Software Distribution Status 4.2 connector	332
TS.Census 2 connector	333
TS.Census 3 connector	337
CA Unicenter AMO 3 connector	341
CA Unicenter AMO 4 connector	344
Winpark Actima connector	347
Chapter 7. ERP connectors	351
SAP connectors	351
SAP BAPI connector	353
SAP IDoc connector	357
SAP ALE connector	360
Chapter 8. Hewlett-Packard connectors	363
OpenView Service Desk (Outbound)	363
OpenView Service Desk (Inbound)	366
OpenView Inventory Manager Connector	371
Chapter 9. Out-of-the-box scenarios	375
Altiris scenarios	376
Action Request System scenarios	376
Asset Insight scenarios	378
Asset Management scenarios	379
E-mail scenarios	379
Enterprise Discovery scenarios	381
iInventory scenarios	383
OpenView Inventory Manager Scenarios	385
LANDesk scenarios	386
Mercury Scenarios	388
LDAP scenarios	389
MQSeries scenarios	390
NT Security scenarios	392
Peregrine Network Discovery scenarios	394
Peregrine Desktop Inventory scenarios	396
RDS scenarios	399
SAP IDOC scenarios	399
SAP BAPI scenarios	402
SMS scenarios	405
ServiceCenter scenarios	406
ServiceCenter Web Service scenarios	416

TS Census scenarios	418
Tivoli scenarios	420
Tivoli Enterprise Data Warehouse scenarios	424
Unicenter AMO scenarios	424
Web Service scenarios	425
Winpark Actima scenarios	425
A. AQL queries	427
Presentation	427
Recommendations for writing AQL queries	430
Sorts and index	439
The query editor	442
AQL syntax	447
AQL function references	460
Index	465



List of Figures

PEREGRINE

2.1. Connectors - Production and consumption directives	48
2.2. Consumption directives - Reconciliation tab	52
2.3. Consumption directives - Advanced reconciliation tab	58
2.4. Reconciliation - Filtering collections	59
2.5. Reconciliation - Deleting members of the destination collections	60
2.6. Consumption directives - Reconciliation scripts tab	60
3.1. PDI 8 - AM mapping	73
3.2. PDI connector - produced document-type	89
3.3. PDI connector - produced document-type	95
3.4. ServiceCenter Web Service connectors - Type of document consumed - produced	131
4.1. MQSeries connector - Identifying messages using Correl ID	170
4.2. NT Security connector - Published document types	175
5.1. E-mail connector - Non-extended InMailMessage document type	224
5.2. E-mail connector - InMailMessage document type extended using a DTD	224
5.3. Delimited-text connector - Representation of data in a text file	246
5.4. Delimited-text connector - Processing data	262
5.5. Delimited-text connector - Consumption directives	269
5.6. Web Service connector - general principles	271
5.7. XML connector - Consumption directives	296
6.1. Asset Insight connector - Production directive example	307
6.2. Asset Insight connector - AI_CONFIG_T document type	308
6.3. LANDesk for Inventory 8 connector - COMPUTER_SYSTEM document type	312

6.4. SMS connectors - Production directive example	318
6.5. Tivoli CM - Inventory 4.2 connector - COMPUTER_SYSTEM document type	330
6.6. Tivoli CM Inventory 4.2 connector - production directives	331
6.7. Production directives of the TS.Census 2 connector	336
6.8. Production directives of the TS.Census 3 connector	340
7.1. SAP connectors - Consumed-produced document-type	354
A.1. Query editor - composition modes	442



List of Tables

PEREGRINE

1.1. Inventory and database connectors - Advanced configuration options . . .	41
3.1. Configure the Asset Management connector	67
3.2. Configure the Desktop Administration 6.x connector	78
3.3. Configure the Enterprise Discovery connector	82
3.4. Configure the PDI connector	85
3.5. Configure the PDI connector	90
3.6. Configure the Peregrine Network Discovery connector	96
3.7. Configure the BizDoc (read) connector	99
3.8. Configure the BizDoc (write) connector	101
3.9. Configure the SCAuto listening connector	105
3.10. Configure the RDS connector	109
3.11. Configure the ServiceCenter connector	114
3.12. Mapping between the processing report produced by the Asset Management connector and the ConnectItDel document type from the ServiceCenter connector.	120
3.13. Configure the ServiceCenter Web Service connector	131
4.1. Configure the Action Request System connector	135
4.2. Configure the Lotus Notes connector	141
4.3. Configure the MQSeries connector (read)	151
4.4. Configure the MQSeries connector (write)	160
4.5. Configure the NT Security connector	174
4.6. NT Security connector - Examples of production directives	176
4.7. Configure the Tivoli Enterprise Console adapter (sending)	178
4.8. Configure the TEC connector	183
5.1. Configure the Database connector	200

5.2. Configure the XML listening connector	206
5.3. Configure the E-mail connector (fetching)	214
5.4. Configure the E-mail connector (sending)	219
5.5. Document type produced by the E-mail (fetching) connector	222
5.6. Document type consumed by the E-mail (sending) connector	225
5.7. Configure the LDAP connector	229
5.8. LDAP filters	237
5.9. Mapping table (character - ASCII value)	238
5.10. Configure the Command line connector	243
5.11. Configure the Delimited-text connector (read)	247
5.12. Configure the Delimited-text connector (write)	255
5.13. Supported protocols - Web Service connector	272
5.14. Configure the Web Service connector	272
5.15. Web Service connector - Consumed document types	274
5.16. Configure the XML connector (read)	276
5.17. Configure the XML connector (write)	287
6.1. Configure the Altiris connector	302
6.2. Configure the Asset Insight connector	305
6.3. Configure the LANDesk for Inventory 8 Connector	309
6.4.	313
6.5. Configure the SMS 2.x and SMS 2003 connectors	316
6.6. Configure the Inventory connector	320
6.7. Configure the Tivoli Inventory Management 4.0 connector	324
6.8. Configure the Tivoli CM Inventory 4.2 connector	328
6.9. Configure the TS.Census 2 connector	333
6.10. Configure the TS.Census 3 connector	337
6.11. Configure the Unicenter AMO 3 connector	341
6.12. Configure the CA Unicenter AMO 4 connector	344
6.13. Configure the Winpark Actima connector	348
7.1. Correspondence between ABAP data types and Connect-It data types	354
7.2. Configure the SAP BAPI connector	355
7.3. Configure the SAP IDoc connector	358
8.1. Configure the OpenView Service Desk (Outbound) connector	364
8.2. Configure the OpenView Service Desk (Inbound) connector	366
8.3. Configure the OpenView Inventory Manager connector	371
A.1. AQL - syntax conventions	447
A.2. AQL - logical operators	451
A.3. AQL - comparison operators	452
A.4. AQL - Aggregate-type functions	460
A.5. AQL - String-type functions	461
A.6. AQL - Date-type functions	461
A.7. AQL - Examples of Date-type functions	462
A.8. AQL - Numeric-type functions	463
A.9. AQL - Test-type functions	464



Introduction

What do the connectors do?

Connectors communicate with the external applications and enable these applications to exchange data.

The connectors can be divided into three categories:

- Base connectors
These connectors are available to all users of the Scenario Builder.
- Optional connectors
These connectors are provided with the Scenario Builder. Depending on your license, however, you may or may not have the right to use them.
- Additional connectors
These connectors are not provided with the Scenario Builder. You must obtain them from Peregrine Systems.

Who are the connectors intended for?

The connectors are for people who integrate the different applications used by their company.

How to use this guide

Connector chapters

The following chapters:

- [Peregrine Systems connectors](#) [page 63]
- [Application connectors](#) [page 133]
- [Protocol connectors](#) [page 193]
- [Inventory connectors](#) [page 301]
- [ERP connectors](#) [page 351]

describe the connectors available in Connect-It. The following information is available for each connector:

- Compatibility
- Limitations
- Configuration
- Directives
- Additional information

Chapter Out-of-the-box scenarios

This chapter presents the scenarios provided with the Scenario Builder.

These scenarios are located in the **scenario** folder of the Connect-It installation folder. The out-of-the-box scenarios (SCN file) are all referenced according to their position in the sub-folder. For example, the file `fc\fcac41\fcac.scn` is located here: **[Connect-It installation folder]\scenario\fc\fcac41\fcac.scn**.

Compatibility of the connectors

This guide does not include version numbers for the software supported by the connectors.

For compatibility-related information, refer to the Compatibility Matrix available on the Peregrine Systems customer support site: <http://support.peregrine.com>

You will require a valid login and password to access this site.



1 | Configuring connectors

CHAPTER

Configuring a connector in a scenario enables you to:


- Name and describe it.
- Specify the connection parameters that enable it to communicate with an external application (database, messaging system, queue manager, etc.).
- Specify a multitude of options enabling it to process the document types it publishes, produces or consumes.

After explaining how to access the connector configuration wizards, this chapter's sections regroup the pages common to all connector wizards. In chapters [Peregrine Systems connectors](#) [page 63], [Application connectors](#) [page 133], [Protocol connectors](#) [page 193], [Inventory connectors](#) [page 301], [ERP connectors](#) [page 351] the sections detailing connector configuration reference the sections of this chapter. Example: Chapter [Peregrine Systems connectors](#) [page 63], section [Asset Management connector](#) [page 63], section [Configure the Asset Management connector](#) [page 67].

Some of these common pages contain options specific to the configuration of certain connectors. These options are documented in the sections covering the different configurations.

Configuring a connector

Configuring a connector in Connect-It is performed with the help of a wizard. To launch a connector's configuration wizard, you can use any of the following methods:

- **The connector is not in the Scenario diagram pane.**
 - 1 Double-click the connector in the Toolbox.
 - 2 Drag and drop the connector from the Toolbox into the Scenario diagram.
- **The connector is already in the Scenario diagram pane.**
 - 1 Select the connector and choose the **Tools/Configure** menu.
 - 2 Select the connector and press **F2** on the keyboard.
 - 3 Select the connector, right-click and choose **Configure connector** from the shortcut menu.
 - 4 Click .

Advanced mode

The number of pages displayed in the configuration wizard depends on this option.

To enable this option, click the  icon.

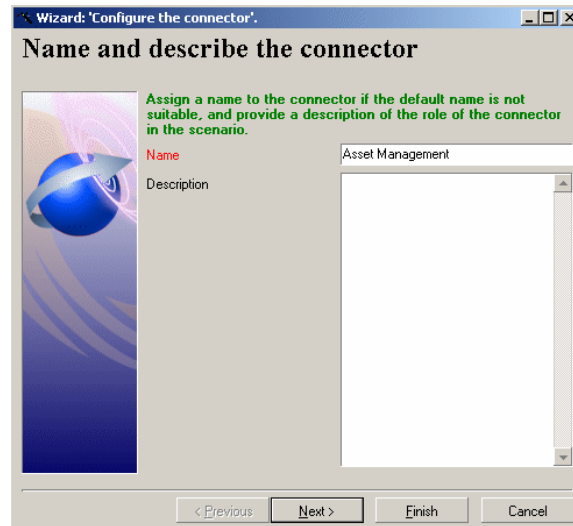
The following pages are displayed in advanced mode:

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

Name and describe the connector

This page enables you to specify:

- The name of the connector.
- The description of the connector's role.



Name

This field enables you to name your connector. By default, the value of the field is the name of the connector as it appears in the Toolbox. If you have another connector of the same kind in the same scenario, Connect-It will add a number to the connector name: **NameOfConnector**, **NameOfConnector1**, **NameOfConnector2**, etc.

This name identifies the connector in the scenario, and each connector displays its name in the Scenario diagram.

 **Warning:**

If you use the same connector more than one time in a scenario, you need to assign them each a different name. Example: a different name for each Asset Management connector in a scenario to transfer data between two AssetCenter databases.

Description

Enter text to describe the connector's role in the scenario. (Example: Reading or write such and such a data source.)

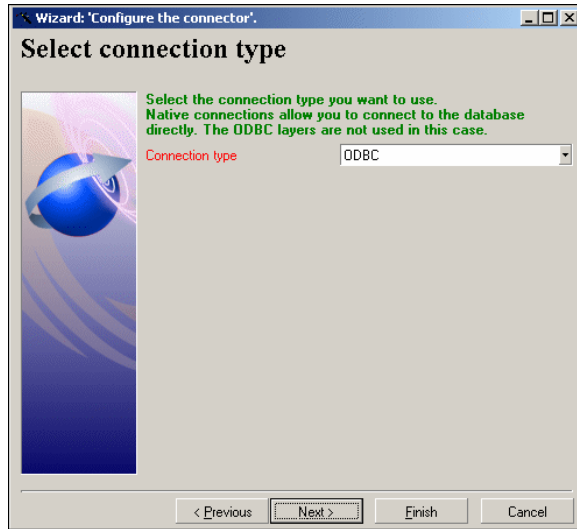
This field is not mandatory.

Select a connection type

This page enables you to select the type of connection for a database application. You can choose from:

- An ODBC data source.
In this case, the ODBC layers are used.
- A native Oracle connection.
- A native Sybase connection.
- A native MySQL connection.

- A native DB2 connection.



 **Important:**

We recommend that you use a native connection rather than an ODBC connection. A native connection performs better, notably during the auto-description of a database by the connector.

 **Note:**

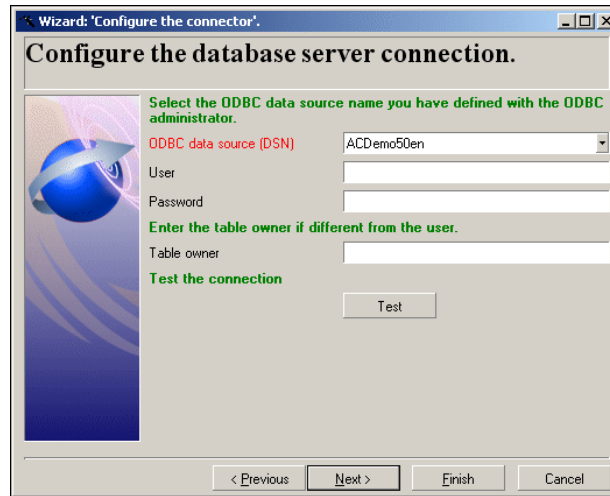
To use a MySQL native connection, you must copy the MySQL dynamic library (DLL) client to the **bin** folder of the Connect-It installation folder (or in the **system32** folder of the operating system).

Configuring the connection

This page enables you to configure a connection.

This page changes according to the type of connection you choose on the [Select a connection type](#) [page 18] page.

Select a connection - ODBC connection



This page enables you to specify:

- An ODBC data source
- A user
- A password
- An owner of the tables

ODBC data source

Indicate the name of the ODBC connection by selecting a value from the drop-down list. The drop-down list contains the ODBC connections available on your computer. Because this list cannot be modified, the ODBC data source must already be configured in the ODBC Administrator before creating your connector.

User

Enter the user name to access the ODBC data source.

Password

Enter the password associated with the login. This password is not visible when being entered and is stored in an encrypted format.

Table owner

Enter the name of the table owner if different from login.

Test

This button enables you to test your connection.

To test your connection:

- Enter your connection parameters.
- Click **Test**.

The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.

- Click **Close** to return to the configuration wizard.

Select a connection - Oracle connection

This page enables you to specify:

- A database server
- A user
- A password
- An owner of the tables

Data server

Enter the name of your Oracle server.

User

Enter the Oracle login that enables you to access your database.

Password

Enter the password associated with the login. This password is not visible when being entered and is stored in an encrypted format.

Table owner

Enter the name of the table owner if different from login.

Test

This button enables you to test your connection.

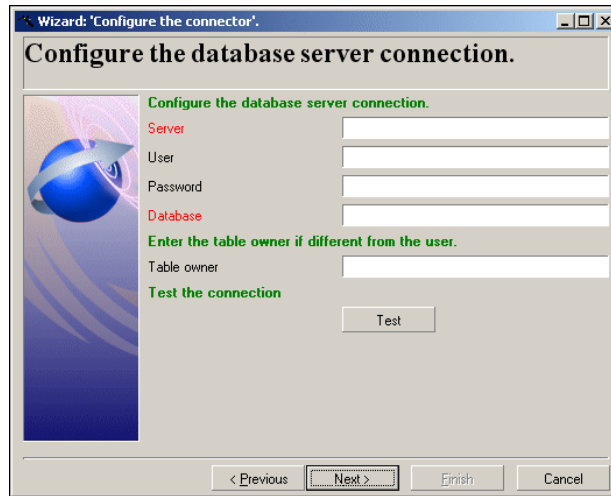
To test your connection:

- Enter your connection parameters.
- Click **Test**.

The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.

- Click **Close** to return to the configuration wizard.

Select a connection - Sybase connection



Wizard: "Configure the connector".

Configure the database server connection.

Configure the database server connection.

Server

User

Password

Database

Enter the table owner if different from the user.

Table owner

Test the connection

Test

< Previous Next > Finish Cancel

This page enables you to specify:

- A database server
- A user
- A password
- A database
- An owner of the tables

The database server

Indicate the name of your Sybase database server.

User

Enter the Sybase login that enables you to access your database.

Password

Enter the password associated with the login. This password is not visible when being entered and is stored in an encrypted format.

Server database

Indicate the name of the database to which you want to connect.

Table owner

Enter the name of the table owner if different from login.

Test

This button enables you to test your connection.

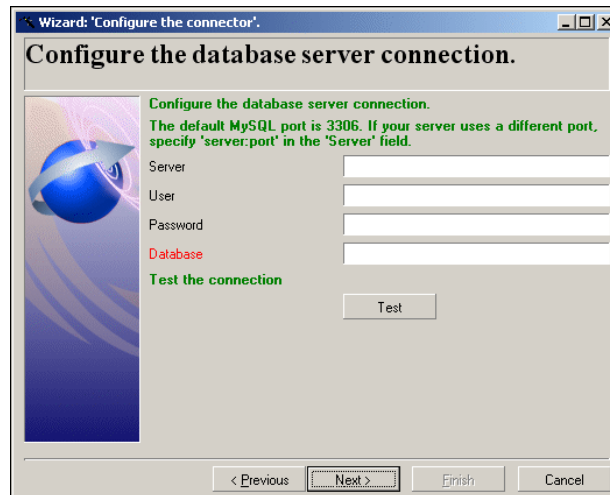
To test your connection:

- Enter your connection parameters.
- Click **Test**.

The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.

- Click **Close** to return to the configuration wizard.

Select a connection - MySQL connection



The screenshot shows a window titled "Wizard: 'Configure the connector'." with the main heading "Configure the database server connection." On the left is a decorative graphic of a blue globe with a white arrow. The main area contains the following text: "Configure the database server connection. The default MySQL port is 3306. If your server uses a different port, specify 'server.port' in the 'Server' field." Below this are four input fields labeled "Server", "User", "Password", and "Database". A "Test the connection" button is located below the "Database" field. At the bottom of the window are navigation buttons: "< Previous", "Next >", "Finish", and "Cancel".

This page enables you to specify:

- A database server
- A user
- A password
- A database

Database server

Enter the name of your MySQL server.

User

Enter the login that enables you to access your database.

Password

Enter the password associated with the login. This password is not visible when being entered and is stored in an encrypted format.

Server database

Indicate the name of the database to which you want to connect.

Test

This button enables you to test your connection.

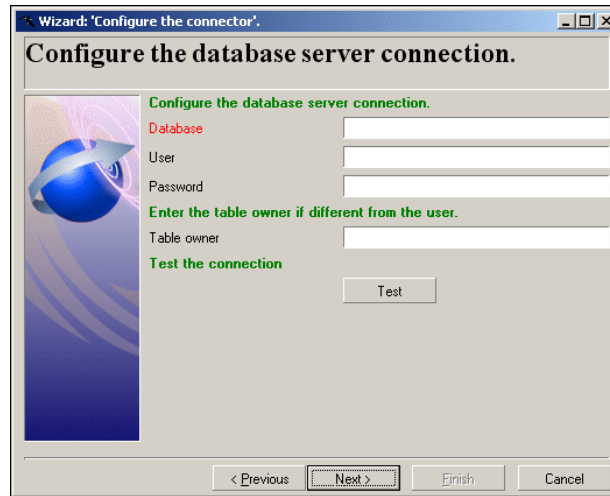
To test your connection:

- Enter your connection parameters.
- Click **Test**.

The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.

- Click **Close** to return to the configuration wizard.

Select a connection - DB2 connection



Wizard: "Configure the connector".

Configure the database server connection.

Configure the database server connection.

Database

User

Password

Enter the table owner if different from the user.

Table owner

Test the connection

Test

< Previous Next > Finish Cancel

This page enables you to specify:

- A database server
- A user
- A password
- A database
- An owner of the tables

The database server

Indicate the name of your DB2 database server.

User

Enter the DB2 login that enables you to access your database.

Password

Enter the password associated with the login. This password is not visible when being entered and is stored in an encrypted format.

Table owner

Enter the name of the table owner if different from login.

Test

This button enables you to test your connection.

To test your connection:

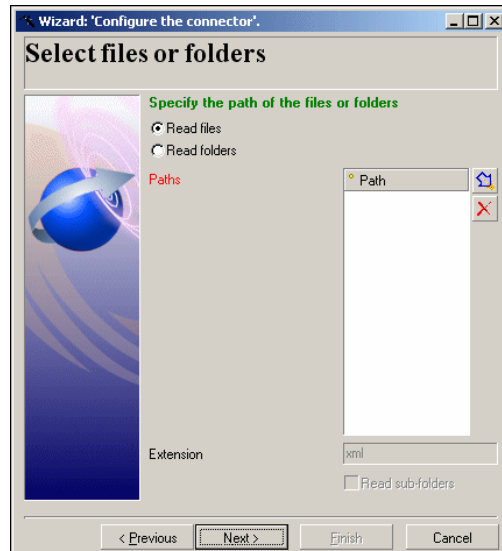
- Enter your connection parameters.
- Click **Test**.

The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.

- Click **Close** to return to the configuration wizard.

Select files or folders

This page of the wizard enables you to specify the locations of the files and folders to be read.



If you select the **Read folders** option, you must specify:

- The path of the folder
- The file extension name.

The default value of the field depends on the connector: **.xml**, **.gz**, **.xsf**.

The extension name is free. You can define multiple extensions by making a list separated by semi-colons.

You must also select the **Read sub-folders** option.

 **Warning:**



Path of a file or folder

When you indicate the path of a file or folder located in a folder associated with a network drive on your computer, you must never indicate the letter of this drive; only indicate the folder associated with this letter.

Using a network drive in a path will stop the scenario from working properly when it is associated with a service. For more information about this, refer to the **User's guide**, section **Edit the mappings**, sub-section **Path of a document's elements**.

Read files

To define multiple files to be read, proceed as follows:

- 1 Click .
- 2 In the field that appears, click  and then select the appropriate file.
- 3 Repeat this operation as many times as required.



If you select this option, the wizard will ask you to enter the path or paths of the XML files on your computer or your network.

Read folders

If you select this option, the wizard will ask you to populate the following fields:

- **Folder name**
Enter the path of the folder that contains your files.
- **Extension**
Indicate the file name extension that the connector must read. For example, **.xml**.

To define multiple folders to be read, proceed as follows:

- 1 Click .
- 2 In the field that appears, click  and then select the appropriate folder

- 3 Repeat this operation as many times as necessary

Read the sub-folders

If you select this option, the connector will also read the files in the sub-folders of the selected folder.

Advanced configuration (advanced mode)

This page enables you to specify the advanced configuration parameters

SQL92 supported

By default, this option is selected. It indicates the SQL queries sent to the database by the connector respect the SQL92 syntax. If this database does not support this syntax, you must clear this option. Example: Informix.

Archival

When a connector uses post-processing actions, you can archive the files that are moved. Archival is enabled when the **Move it to folder** option is selected. This is the case in case of both failure and success.

Archival is performed for a given period of time or for a given number of versions of a file you wish to keep.

Archival is enabled when you select the **Keep archive** option.

Each archived file is suffixed according to the defined configuration.

Archival settings

Configuring the archival means defining one of the following:

- A time limit to keep a file (time, day, week, month, year).
This duration is defined in the **Date** frame.
- A maximum number of versions of a file to keep.
Populate the **Number of versions of a file to keep** field.

- Indefinite archival.
Select the **Keep all** option.

Archival using the date

When you choose to archive using the date, each file name is suffixed with the date.

You must specify the date format used.

The following date formats are available:

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

For example, ex_2003-09-02-16-20-05.

Note:

Only the suffixed date in the file name is taken into account and not the creation date of the file itself.

Archival using a number

When you choose to archive using a number instead of the date, each file will be suffixed with a number.

You must specify the way in which the files will be incremented. This incrementation is defined in the **Numbering** frame.

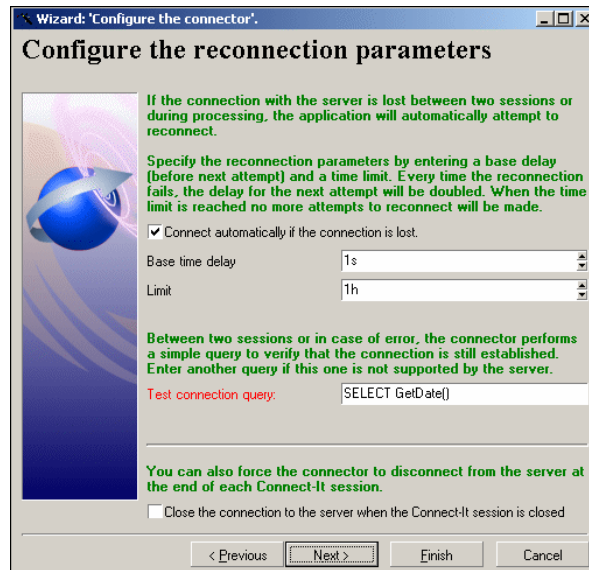
Incrementation is either by:

- Ascending sort: Each file keep is added to the end of the queue.
For example, for the files ex_1, ex_2, and ex_3, the most recent file is ex_3.
- Descending sort: Each file keep is added to the start of the queue.
For example, for the files ex_1, ex_2, and ex_3, the most recent file is ex_1.

The number of digits using as the suffix and the separator used can be configured.

Configure the reconnection parameters (advanced mode)

This page enables you to configure your reconnection parameters in case the initial connection is lost.



Connect automatically if the connection is lost

If you select this option, you must specify how Connect-It should reconnect in case the connection to the server fails or is interrupted. If it is a database, then it's the server on which this database is located.

 **Note:**

This option is selected by default. If the database server does not support reconnections, you must clear this option.

These are the following reconnection parameters:

- **Base time delay**

The base time delay enables you to specify how long Connect-It should wait before trying to reconnect to the server. If reconnection is not successful, Connect-It doubles this period before trying again. Example: With a base time delay of 2 seconds, the second attempt to reconnect will take place after 4 seconds, the third after 8 seconds, etc.

- **Time limit**

The time limit enables you to specify the maximum length of time after which Connect-It should give up trying to reconnect to the server.

 **Important:**

After successful reconnection:

- While writing data, the document whose processing was interrupted is processed again, and the session can continue normally.
 - While reading data, the interrupted session is restarted in full.
-

Test connection query

Each time you open a new session, or in case of a processing error, a query is sent to the server in order to verify that the database connection is still open. The request must be supported by the server, and must be efficient, in order for optimal document processing.

The query emitted by default is **SELECT GetDate()**. If this query is not supported by the database, enter a valid query.

 **Important:**

Each time it is open, the connector emits the test connection query. If this query is not supported by the server, a warning message will tell you. In this case, automatic reconnection is not possible.

When losing a connection is not an issue, you can deactivate this option so that you don't obtain a warning message when opening the connector. Example: An ODBC connection to an Excel file.

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

Do not select this option unless told to do so by Connect-It support technicians.

Determine server delay (advanced mode)

This page enables you to determine the delay with the server to which the connector connects when it reads or receives data.

Wizard: 'Configure the connector'.
Determine server delay

When the pointer status is a date, it is the current date (day, hour, minute, second) that becomes the last date from which to recover record modifications or creations. It is also this date that will become the first date from which to recover record modifications or creations the next time you start the scenario. If the server delay is not accounted for (whether it be minutes, hours or days), you can adjust the pointer status date so that it recovers record modifications or creations either before or after the current date set on your computer.

Time difference with server: 0s

The current date may be given by the server in a time zone that does not match the field used as schedule pointer. In particular, a UTC date (GMT 0) may be given. In this case, specify the time zones required for the conversion.

Automatically calculate the time difference with the server

Time zone of the server: [Dropdown]

Time zone of the pointer: [Dropdown]

Date and time query: SELECT GetDate()

Test server delay

Test

< Previous Next > Finish Cancel

Automatically calculate the time difference with the server selected

This option is selected by default.

In this case, Connect-It queries the server at the start of each session to automatically adjust the time different between Connect-It's clock and the server's clock.

Automatically calculate the time difference with the server cleared

If you do not select the **Automatically calculate the time difference with the server** option, you must manually enter the time difference.

You must manually enter a value when the connector cannot automatically determine this time difference.

Date and time query

This field enables you to specify a query used to return the time of the server in order to compensate the time used locally by the scheduling pointers.

A query is performed

- when the connector is opened
- each time a session is opened, in order to update the delay between the server and local time


Using time zones

You must use the **Time zone of the server** and **Time zone of the pointer** fields when the date returned by the server does not correspond to the time zone expected by the external application to perform filtering operations.

Examples

- The server returns a UTC timestamp (Universal Time Coordinated) when, in fact, the external application saves this timestamp under a different time zone.
- The external application returns the current timestamps in a given time zone when, in fact, the operation to filter these timestamps respect the time zone defined in Connect-It server's 'Date and time' parameters.

To specify a time zone

- 1 Verify that the **Automatically calculate the time difference with the server** option is selected.
- 2 Click .

- 3 Select a time zone in the window that appears.



Select or not the **Apply daylight savings time** option if your connector accounts for daylight savings time.

- 4 Click **OK**.


To display the current timestamp of the external application

To the right of the title **Test time difference with server**, click **Test**.

This action triggers the following operations, which you can view in a contextual window:

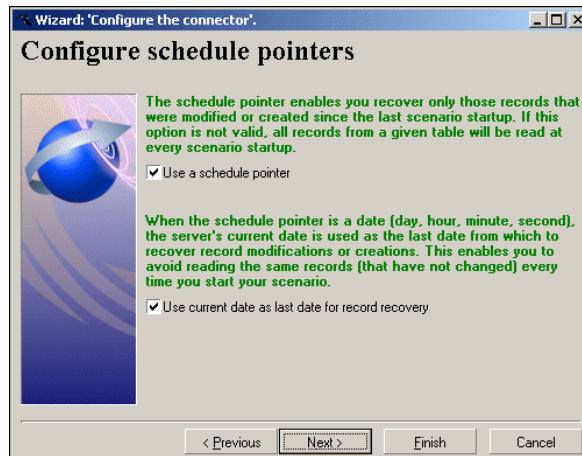
- Connection to the external application.
- Recovery and display of the external application's current timestamp.
- Display of the current timestamp after having taken into account the time server delay (if time zones have been specified).

To test the time zone parameters

- 1 Modify a record in the external application.
- 2 Create a document type that corresponds to this record.
- 3 Apply a filter on the field indicating the record's last modification date. The value of this field in the syntax of the filter must correspond to the external application's date and time, obtained using the **Test** button.
- 4 In the edit window of the document types produced by the connector, click  to verify that the modified record has been recovered.

Configure schedule pointers (advanced mode)

This page enables you to configure the schedule pointers used by the connectors.



When a scenario is launched in scheduled mode: The status of a document-type's pointer tells the connector to process only those documents that have not been processed since the last session.

Example: In its last session, an Asset Management connector took until 9h00mn00s to read all the records in the Assets table. Thus, in the next session, it will read all the records created after 9h00mn00s.

Use a schedule pointer

This option is selected by default.

When this option is selected, the connector only processes the data that was modified since the last session. Example: A database-type connector only processes the records that were created or updated since the last session.

If this option is not selected, the connector processes all the data in the source application. Example: You use a database-type connector to save a copy of the database. In this case, the record-modification date is not used.

Use current date as last date for record recovery

This option is selected by default.

To illustrate this option, we'll use the case of a scheduler that wakes up a connector whose job is to read source data every hour on the hour. The last session was at 9h00m00s, and this session is at 10h00m00s.

Use current date as last date for record recovery option selected

In this case: During the new session, the connector only processes the data whose modification date is:

- Greater than or equal to the date of the last session.

Example: all data whose modification date is greater than or equal to 9h00m00s.

Data modified at exactly 9h00m00s will thus be processed by the connector.

- Less than the current date (adjusted to the server delay if this can be automatically calculated).

Example: all data whose modification date is less than 10h00m00s.

All the data modified after 10h00m00s is not processed by the connector.

Use current date as last date for record recovery option cleared

In this case: During the new session, the connector only processes the data whose modification date is:

- ◆ Greater than or equal to the date of the last session.

Example: All data whose modification date is greater than or equal to 9h00m00s.

A record created a exactly 9h00m00s will be processed by the connector.

All the data created after 10h00m00s will be processed by the connector.

This is useful when the connector cannot automatically determine the time difference with the server.

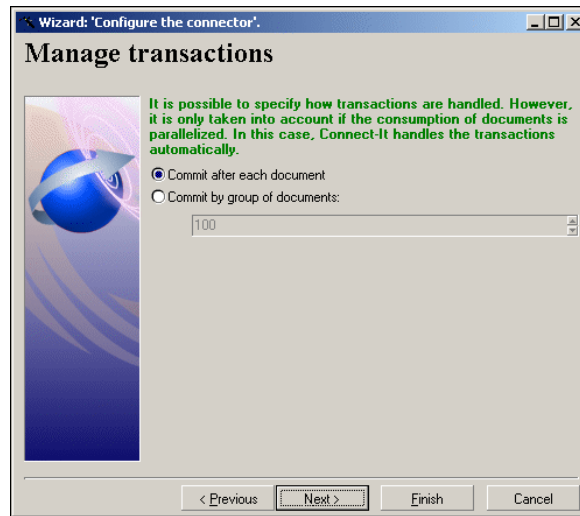
 Note:

The date of the schedule pointer uses the format defined for the local computer. This format is saved in the `[scenario name].ini` file of the folder of the current scenario.

For more information about schedule pointers, refer to the **User's Guide**, chapter **Putting a scenario into production**.

Manage transactions (advanced mode)

This page enables you to define the mode you use to transmit documents used by the connector.



The connector can:

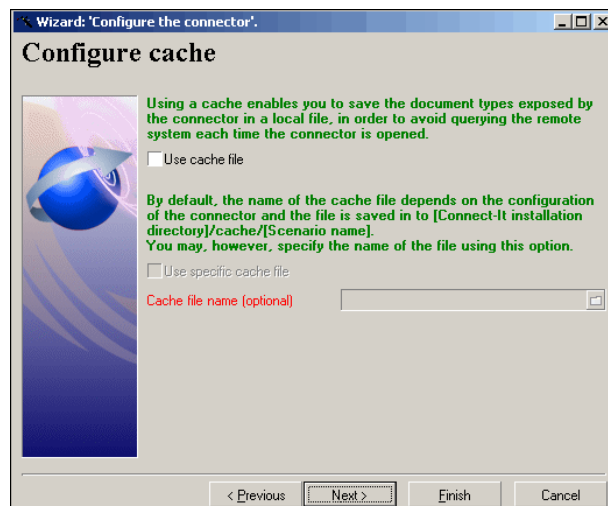
- **Commit** after each document.
After each document is processed, the **Commit** command is performed. This validates the insert, update or delete queries emitted while processing the document.
- **Commit** by group of documents.
After processing the indicated number of documents, the **Commit** command is performed. This validates the insert, update or delete queries emitted while processing the documents. The number of documents to process is set to **100** by default.

 **Note:**

If you're using this mode and an error appears while processing documents, Connect-It will insert one document at a time until it comes to the erroneous document. For optimal results, we recommend using the **Commit** by group of documents option.

Configure the cache (advanced mode)

This page enables you to use a cache file containing the description of document types published by the connector. The file extension for such a file is **.cch**.



In the case where you use an application that is not located on your local network, a cache enables you to quickly work with the connector without having to wait for the retrieval of the document types.

 **Important:**

We recommend that you use a cache file to be able to access data faster and to enable you to work offline.

For more information about using cache files, refer to the **User's guide**, chapter **Testing and debugging an integration scenario**, section **Using cache files**.

You can choose from two options:

- **Use cache file**

If you select this option, a cache file is created in a folder named after your scenario in the **cache** folder of the Connect-It installation folder. The name of each cache file **CCH** is unique. In addition, the cache files of each instance of the same connector have different names. Generally, the cache file takes its name from the name of the server.

- **Use specific cache file**

If you select this option, you need to indicate the path and file name of the cache that you want to create or reuse.

If this cache file does not exist, Connect-It will create it when you open your connector for the first time. If this file already exists Connect-It suggests that you either overwrite it or rename your new cache file.



Note:

Use this option when you want to use a cache file that was created in a previous scenario.

Important: The cache must be synchronized each time that the external application's structure is modified: creation of a new table, creation of an output event, etc.

To synchronize a connector's cache:

- 1 Select the connector in the Scenario diagram.
- 2 Right-click.
- 3 Select **Cache/Synchronize** from the shortcut menu.
(**Tool/Cache/Synchronize the cache**)

Advanced options (advanced mode)

This editable zone enables you to enter advanced options that are different depending on your connector.

To enter an advanced option:

- 1 Click

- 2 Enter the name of the option in the **Name** column.
- 3 Enter the value of the option in the **Value** column.

To modify the name or the value of an option, double-click directly on this name or value and enter a new one.

To delete an option, select the row corresponding to this option and click **x**.

Advanced options for the inventory and database connectors

The following table presents the advanced options that can be entered for the Inventory and Database connectors.

Warning:

The default values of the advanced options should not be modified unless Peregrine Systems or the accompanying documentation tell you to do so.

Table 1.1. Inventory and database connectors - Advanced configuration options

Name	Default value	Description	Data type	Examples
StmtCache	30	Cache size	<ul style="list-style-type: none"> ■ Long integer ■ Number of instances 	Enter 0 to: <ul style="list-style-type: none"> ■ DB2 ■ Microsoft Visual Studio
ConstAsText	0	Constant format using the text format.	Boolean: <ul style="list-style-type: none"> ■ 0 = False ■ 1 = True 	Enter 1 to: <ul style="list-style-type: none"> ◆ ODBC Novell driver
NoPrepare	0	Prevents use of dynamic SQL.	Boolean: <ul style="list-style-type: none"> ■ 0 = False ■ 1 = True 	Enter 1 to: <ul style="list-style-type: none"> ◆ ODBC Novell driver
AutoCommit	1	Indicates the automatic mode (1) or manual mode (0) of the commit command.	Boolean: <ul style="list-style-type: none"> ■ 0 = False ■ 1 = True 	Enter 0 to: <ul style="list-style-type: none"> ◆ ODBC Novell driver
RowsetSize	1	Enables to specify the number of rows to return.	Boolean: <ul style="list-style-type: none"> ■ 0 = False ■ 1 = True 	Enter 0 to: <ul style="list-style-type: none"> ◆ ODBC Novell driver

Name	Default value	Description	Data type	Examples
ExtendedFetch	1	Uses extended fetching.	Boolean: <ul style="list-style-type: none"> ■ 0 = False ■ 1 = True 	Enter 0 to: <ul style="list-style-type: none"> ◆ ODBC Novell driver
AddIndexDesc	1	Enables the auto-description of indexes.	Boolean: <ul style="list-style-type: none"> ■ 0 = False ■ 1 = True 	Enter 0 to: <ul style="list-style-type: none"> ◆ ODBC Novell driver
FetchingArray-Size	Depends on the database engine.	Enables to specify the number of rows to link.	Long integer	Enter 1 to: <ul style="list-style-type: none"> ◆ ODBC Novell driver
OdbcSelectCursorType	0 for DB2 and Oracle 1 for all other engines	Enables to specify the type of cursor.	Long integer: <ul style="list-style-type: none"> ■ 0 FORWARD_ONLY ■ 1 KEYSET_DRIVEN ■ 2 KEYSET_DRIVEN ■ 3 STATIC 	Enter 0 to: <ul style="list-style-type: none"> ◆ DB2 NT

Define document types (advanced mode)

This page enables you to define the extension file for the document types published by the connector.

This file contains:

- The definition of the joins between the different tables of the database.
- Fields used as schedule pointers.

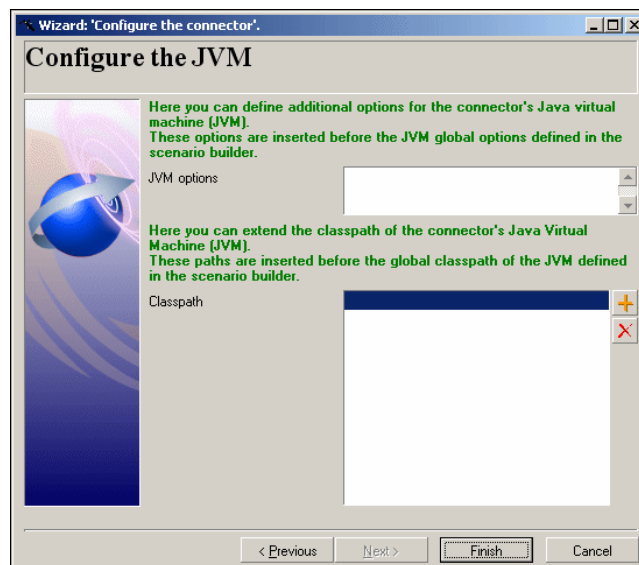
The original file must not be modified. For further information on editing an extension file, refer to the **User's Guide**, chapter **Configuration editor**.

 **Warning:**

Certain extension files cannot be modified using the configuration editor; In this case, you must use the extension file of the Database connector (`config\database\config\database.cfg` file located in the Connect-It installation folder.)

Configure the JVM

This page of the connector configuration wizard is available for all connectors using a JVM.



In Connect-It, the JVM options and the classpath are specified for the entire application using the **Java/Configure the JVM** menu. The Configure the JVM page allows you to override the parameters defined at the global level and define additional JVM parameters for a connector.

The configuration fields are the following:

- **JVM options:** use this field to specify additional options similar to those defined using the **Java/Configure the JVM** menu. The connector is redeployed automatically once the additional options have been defined.
-

 **Note:**

The values of the JVM options are exposed when exporting the scenario configuration (function **-export**).

- **ClassPath:** This field allows to add elements to the classpath. The classpath elements thus defined must come before those defined in the JVM general configuration.
-

 **Note:**

This property is exposed when exporting the scenario configuration (the **-export** parameter).

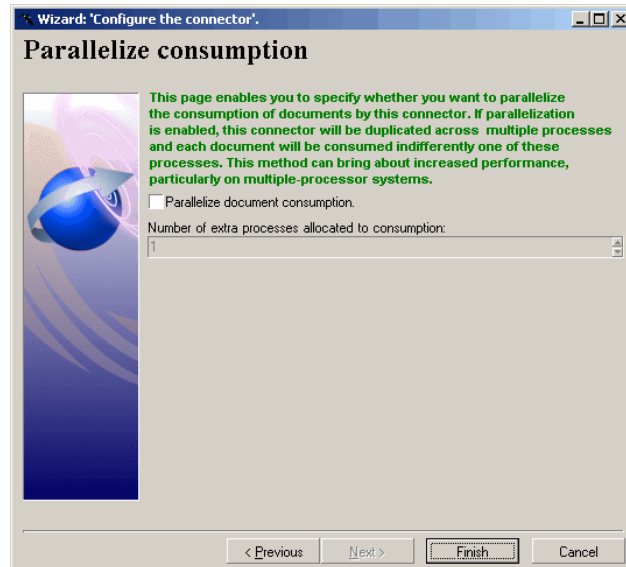
Parallelize consumption (advanced mode)

Parallelization is applied to all connectors consuming a document type. It consists of duplicating a connector, making multiple processes, in order to process the consumption of documents in parallel.

This page enables you to:

- Enable this option

- Define the number of processes running in parallel



Parallelize the consumption of the documents

Select this option to enable the parallel processing of your consumed document types.

Number of additional processes allocated to consumption

Populate this field with the number of processes running simultaneously and allocated to the consumption of documents.



2 Connector directives

CHAPTER

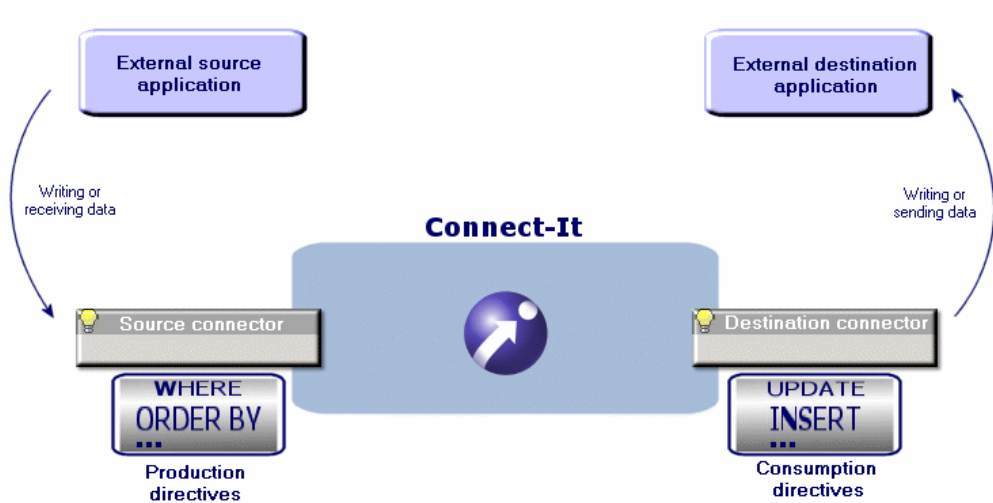
Directives are instructions for the connector that are defined by the user. The connector follows these instructions when producing or consuming documents.

These directives are entered when you create:

- Produced or consumed document types.
Or
- Mappings between document types.

Directives vary depending on the connector. Example: For database type connectors, the production directives correspond to WHERE and ORDERBY clauses that filter the values of fields recovered in the source database. On the other hand, the consumption directives correspond to the reconciliation options that let you write data to a destination database.

Figure 2.1. Connectors - Production and consumption directives



This chapter presents the production and consumption directives that are common to several connectors.


Production directives

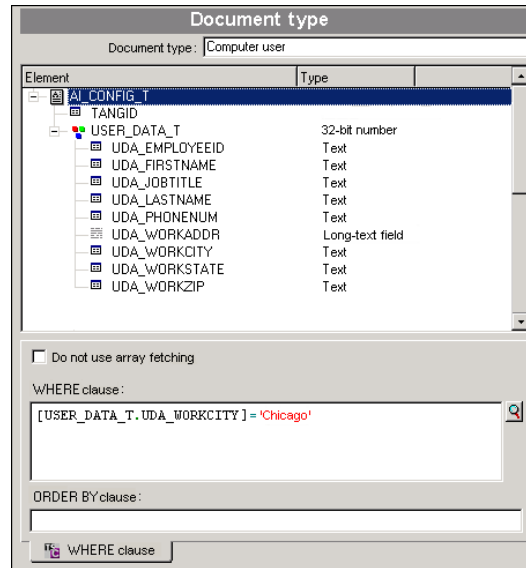
The production directives concern how data that is read or received by a source connector is processed. Example: The directives for the MQSeries connector consist of a WHERE clause and some options concerning how to recover messages.

Defining production directives

To define a connector's production directives:

- 1 Load or create a scenario that uses this connector.
- 2 Verify that the connector is correctly configured.
- 3 Open the connector (**Tools/Open** or **F4** on the keyboard).
- 4 Select the **Document types** tab.

- 5 Click  or edit a document type produced in the **Produced document types** pane.
A window appears in which you can create or edit a produced document type.
- 6 Enter the production directives in the window.



WHERE and ORDERBY clauses

Use the WHERE clause to filter the records in a source database.

For example, you can filter records extracted from the database created on or after January 1, 2002. Simply use the following clause:

```
[field containing the creation date for a record] >= '1/01/2002'
```

For example, to filter a link, use the following syntax:

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

Conventions used for the connectors supporting AQL

For the connectors using AQL, Connect-It imposes that you respect the following conventions:

- Field names must be stated in between square brackets.

Example

[Field1]

- Character strings must be surrounded by single quotes.

Example:

'computer'

- Numbers must be stated without quotes.
- Dates must be entered using the format defined in the regional settings of your computer and be stated in between single quotes.

Connectors supporting AQL

The following connectors support AQL:

[Asset Management connector](#) [page 63]

[Desktop Administration 6.x connector](#) [page 76]

[Database connector](#) [page 193]

[LANDesk for Inventory 8 Connectors](#) [page 309]

[Tivoli Inventory connector \(version 4.0\)](#) [page 323]

[System Management System \(SMS\) connectors](#) [page 315]

[TS.Census 3 connector](#) [page 337]

[CA Unicenter AMO 3 connector](#) [page 341]

[Tivoli CM for Inventory 4.2 connector](#) [page 327]

[Winpark Actima connector](#) [page 347]

Sysdate field

In the WHERE clauses written in AQL, the use of the **sysdate** field must be replaced with the function **getdate()**.

Example: The SQL query: **SELECT * from AmAsset where dinstall>sysdate** must be replaced with the AQL query **SELECT * from AmAsset where dinstall>getdate()**.

Do not use array fetching

This option must be selected, for example, when recovering Blob-type fields or **memo**-type fields when the table does not have a main field. If array fetching is used, Blob-type fields may not be processed correctly (they are clipped). This option may reduce performance in particular.

Consumption directives

The consumption directives concern how data that is written or sent by a connector to an external destination application is processed. Example: The directives for a database connector enable you to specify how the connector writes or updates records in the destination database.


Defining a connector's consumption directives

You can define the consumption directives:

- In the Consumed document-types edit window.
- In the Mappings edit window.

Consumption directives in the Document-type edit window

To define a connector's consumption directives in the Consumed document-types edit window:

- 1 Open or create a scenario that uses this connector.
- 2 Verify that the connector is correctly configured.
- 3 Open the connector (**Tools/Open** or **F4** on the keyboard).
- 4 Select the **Document types** tab.
- 5 Click  or edit a consumed document type in the **Consumed document type** pane.

You can create or edit a consumed document type in the Document type edit window that appears.


- 6 Enter the consumption directives in the window.

Consumption directives in the Mappings edit window

To define a connector's consumption directives in the Mappings edit window:

- 1 Open or create a scenario that uses this connector.

In the scenario, the mappings need to have already been created between this connector and another connector.

- 2 Open your scenario's connectors.
- 3 Select the **Mappings** tab.
- 4 Click  or edit an existing mapping

The Mapping edit window appears.

- 5 Enter the production directives in the Mappings window.

Reconciliation

For database type connectors, you specify consumption directives by indicating reconciliation parameters.

These parameters indicate:

- Which fields are used to reconcile data consumed by a connector and the records in the destination database.
- Which operation the connector performs on the records in the destination database: insert, update or delete.

Consumption directives must be entered in the **Reconciliation** and **Advanced reconciliation** tabs.

Using the Reconciliation tab

The Reconciliation tab enables you to:

- 1 Indicate a reconciliation type for each complex element (root node, collection, structure).
- 2 Indicate the sub-nodes as reconciliation keys.

Figure 2.2. Consumption directives - **Reconciliation** tab

Element	Keyset #1
Reconciliation key	Reconciliation key
Behavior	Replace

Refer also to the **User's Guide**, chapter **Putting a scenario into production**, section **Reconciliation keys**.

Selecting a reconciliation type

The reconciliation types enable you to select the action to be performed:

- Update or insert
- Update

- Insert only
- Delete

To indicate a reconciliation type

- 1 Select a root node, a structure or a collection in the Mappings window.
- 2 Select the option of your choice in the **Reconciliation type** drop-down list.

Select the reconciliation keys

After having selected a reconciliation type for a complex element, you must select the fields enabling the reconciliation of source and destination data. These fields are identified by a reconciliation key.

The fields selected as reconciliation keys must respond to the following criteria:

- Restriction of doubles

Example: an **ID** field in a database table does not allow two records to share the same ID. So, the **First name** field in the Employees table is a bad choice for a reconciliation key since several employees can share the same first name.
- Restriction of null values

Example: A **serial number** field in the Computers table of a computer hardware supplier's database. Under no circumstances can a computer not have a serial number.
- Indexing

The reconciliation process goes faster if an indexed field is used as the reconciliation key.

To select a reconciliation key

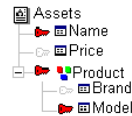
- 1 Select this element in the work area.
- 2 Then, select the **Reconciliation key** option or click directly on the transparent key in the pane showing the consumed-document type.

For example: In the **amAsset** document type of the **pd\pd\ac44\pd\ac.scn** scenario, the **AssetTag** field and the **Brand** and **Model** fields of the **Product** structure are chosen as reconciliation keys.

When a structure is used as a reconciliation key

The first level of a document type published by the Asset Management connector represents a table in Asset Management. The tables linked to this table by a 1-1 type link are represented by structures.

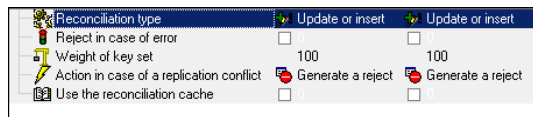
You can identify a record with the fields of a structure by choosing them as reconciliation keys. In order for these fields to be taken into account when transferring data, you must also select the structure containing these fields. In fact, each structure contains an **Identifier** field (**not shown on-screen**) that allows it to be linked to another table. Placing a reconciliation key on the structure comes down to indicating that the **Identifier** field is also used to identify the record.



The previous illustration shows that the document type representing an asset creates a record using the:

- **Name** field of the **Assets** root node.
- **Identifier** field of the **Product** structure.
- **Model** field of the **Product** structure.

To use a parent ID as the reconciliation key

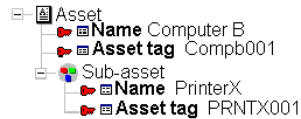


- 1 Select a collection.
- 2 Select the **Use the parent Id as the reconciliation key** option.

Example: In the database of an Asset Management application, the children tables are linked to the parent tables by a foreign key: the ID of the parent. In AssetCenter, the Portfolio items table (**amPortfolio**) is linked to the Sub-assets table (the table of the items of the portfolio) by a **lparentId** field.

When you select the **Use the parent Id as the reconciliation key** option, any creation of a record linked to a parent record is rejected if the relationship is different to that in the database. This is only valid if the link between two tables is a 1-1 or 1-N type link: A record in the child table can only be linked to a single record in the parent table.

There is a parent-child relationship between a computer A and a printer X. A document consumed by the Asset Management connector has a new parent-child relationship that links a computer B to the same printer X. The names and asset tags of the asset and sub-asset are chosen as the reconciliation keys.



Use the parent Id as the reconciliation key option - Example

This example illustrates how Connect-It behaves when the **Use the parent Id as the reconciliation key** option is selected and then how it behaves when this option is not selected.

1 Use the parent Id as the reconciliation key option selected

The parent-child relationship is not updated because the **Use the parent Id as the reconciliation key** check box indicates that the Printer X sub-asset is already linked to an asset in the Assets table.

Assets table		Sub-assets table				
Name	Asset tag	Identifier	Parent Id	Name	Asset tag	
Computer A	COMPA001	CPAAA1	- -	CPAAA1	Printer X	PRNTX001
Computer B	COMPB001	CPBBB1				

2 Use the parent Id as the reconciliation key option cleared

The parent-child relationship is updated. The **Parent Id** field is updated and links the sub-asset Printer X to the asset Computer B.

Assets table		Sub-assets table				
Name	Asset tag	Identifier	Parent Id	Name	Asset tag	
Computer B	COMPB001	CPBBB1	- -	CPBBB1	Printer X	PRNTX001

To couple the reject of a parent node to its child node

In certain cases, you can indicate that the rejection of a parent node automatically triggers the rejection of its parent node.

- 1 Select a document-type element in the workspace.
- 2 Select the **Reject the parent node if rejection** option.

Example: In an Asset Management application, you do not want a record to be created in the Portfolio items table (**amPortfolio**) if the record that is linked to it in the Employees table (**amEmpIDept**) - the user of this asset - is rejected. This avoids having assets without users in the Asset Management database.

Using alternate reconciliation key sets

Connect-It enables you to use several reconciliation key sets. Each key set is assigned a given weight.

This is useful when the reconciliation keys do not let you reconcile your records.

Example: In the source document, a field selected as the reconciliation key doesn't have a value even though the destination application requires that it be populated.

By indicating that a reconciliation key belongs to a second set Connect-It uses this key if the reconciliation performed using the first key set fails.

For more information on reconciliation key sets, refer to the **User's Guide**, chapter **Putting a scenario into production**, section **Reconciliation keys**.

Case-sensitive reconciliation

For this option, the behavior of Connect-It is as follows:

- **Case-sensitive reconciliation** option selected
The reconciliation key values are case-sensitive. Example: If a field containing the e-mail address of your employees is used as a reconciliation key, the values 'jmartin@company.com', 'JMARTIN@company.com' and 'jMartin@company.com' will correspond to three different records in the destination application.
- **Case-sensitive reconciliation** option cleared
The reconciliation key values are no longer case-sensitive. Example: If a field containing the e-mail address of your employees is used as a reconciliation key, the values 'jmartin@company.com', 'JMARTIN@company.com' and

'jMartin@company.com' will correspond to the same record in the destination application.

 **Note:**

This option only appears if the database engine is case sensitive. For connectors using an ODBC connection, this option is selected by default.

Select a reconciliation behavior.

For each destination field, you can select one of the following options:

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

By default, the **Replace** option is selected. The **Append** option is useful for **memo**-type fields. Example: You can append comments to comments already in a database table.

Managing replication conflicts

If a replication conflict is encountered, you have the choice of three options:

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

For a useful presentation of replication conflicts in a ServiceCenter and Asset Management scenario, refer to the chapter [Out-of-the-box scenarios](#) [page 375], section [ServiceCenter scenarios](#) [page 406], and see the following sub-sections:

- [The difference between the reconciliation and replication of data.](#) [page 410]
- [Managing the replication of data in Connect-It](#) [page 411]

Using the Advanced Reconciliation tab

The advanced reconciliation tab enables you to:

- Set the reconciliation parameters relating to the collections of a consumed document.
- Indicate the action performed by the connector in case of a replication conflict.

Figure 2.3. Consumption directives - **Advanced reconciliation** tab

Collection reconciliation

Activate collection reconciliation

Action on members not found in the source document (default is delete):

seAssignment = 6

Collection reconciliation filter:

Model.Nature.Code = 'SOFT_INSTALL'

Activating the reconciliation of collections

The **Activate collection reconciliation** option enables you to indicate certain parameters. These parameters let you control updating or inserting destination collection members according to the source collection members.

Managing the absent members of the source collection

The **Action on members not found in the source document** field enables you to indicate that:

- The members absent from the source collection are not deleted from the destination application.
- A field of records corresponding to these absent members have a significant value.

Example: You assign to the collection's **Status** field the value **To delete**. You now need to find all the records that have **To delete** as the value of their **Status** field in the destination-application's administration.

Use the following syntax to indicate this field:

```
[field name] = "[value of your choice]"
```

Example:

```
Status = "To be deleted"
```

Filtering certain collection members

When updating a collection, the **Reconciliation filter** field for the collection allows you to:

- Delete members of the collection that are not found in the source document from the destination application.

- Limit the scope of this deletion by filtering certain collection members in the database of the destination application.

Example :

Let's suppose that in an **Asset information** mapping of a scenario between the Desktop Discovery connector to Asset Management connector, a **Software** source collection is mapped to the **SoftInstall** destination collection. These two collections represent the software installed on an asset (a computer in this case).

Sometimes certain software applications might be uninstalled. By selecting the option **Action on members not found in the source document**, all the software that is not in the **SoftInstall** collection is deleted from the database of the Asset Management application.

However, to avoid deleting certain software (not identified by a Desktop Discovery scan, for example), you can write a clause in the **Collection reconciliation filter** field.

Figure 2.4. Reconciliation - Filtering collections

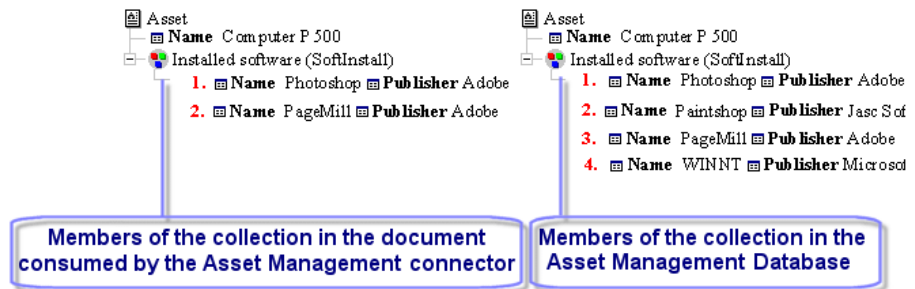
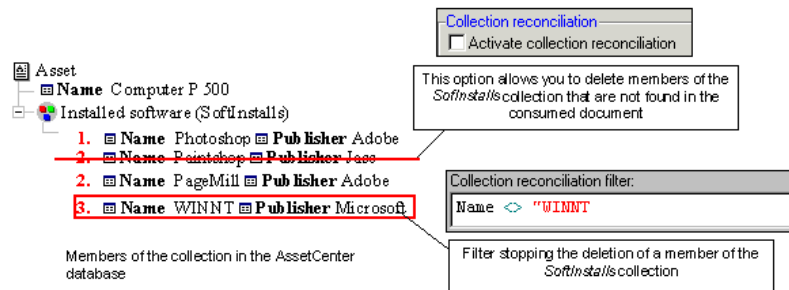
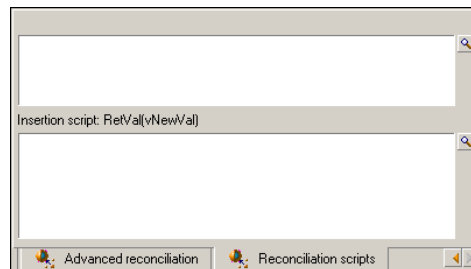


Figure 2.5. Reconciliation - Deleting members of the destination collections



Using the Reconciliation scripts tab (for the Asset Management connector only)

The **Reconciliation scripts** tab is available when the Asset Management connector is in consumption mode.

Figure 2.6. Consumption directives - **Reconciliation scripts** tab

This tab enables you to enter the scripts required to reconcile two document types, a produced one and a consumed one.

These scripts enable you to perform the following actions on the target connector's document type:

- Update if the value exists in the target database
- Insert if the value is missing from the target database

Syntax used

The syntax for the update and insert scripts is the following:

- vNewVal: Value of the source document
- vOldVal: Value contained in the target database
- vOldId: Current value of the ID of the record that will be modified

You can use square brackets [] and Basic functions in your scripts.

For example:

```
// Keeps the old value
RetVal = vOldVal
If vNewVal >= vOldVal and [dtLastModif] > "1980-11-12 10:11:12" Then
pifSetPendingDocument("Document not finished")
DbExecAql("INSERT INTO .....")
// Gets the new value
RetVal = vNewVal
End If
```

How reconciliation scripts work

The reconciliation scripts are an extra step in refining how your mappings are managed.

These scripts concern data contained in the document types produced by a connector and consumed by another. Depending on whether the fields in the target database contain data or not, scripts enable you to modify the value of a mapped field (update) or create it (insert).

Example :

The source connector produces a document type A containing the following fields:

- Name
- Id
- Comment

The ID field is the reconciliation key.

The Asset Management destination connector consumes the document type A, and is connected to a destination database containing the same fields.

The values of the document are as follows:

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

The values of the destination database are as follows:

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

The update script for the Name field is the following

```
RetVal=vOldVal
```

The insert script for the Comment field is the following

```
RetVal=vNewVal
```

After executing the reconciliation scripts, the document consumed by the Asset Management connector will contain the following information:

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

For further information on reconciliation and reconciliation scripts, refer to the **User's Guide**, section **Putting a scenario into production**, sub-section **Reconciliation**.



3 Peregrine Systems connectors

CHAPTER

The Peregrine Systems connectors enable you to process data from Peregrine Systems applications.

Asset Management connector

Base connector

- Production (source)
- Consumption (destination)

This section describes the Asset Management connector.

About the Asset Management connector

The Asset Management connector is dedicated to the following three external applications: AssetCenter, AssetCenter Cable and Circuit, and InfraCenter for Workgroups.

Connect-It uses the DLLs and APIs provided with these applications.

The Asset Management application client must be installed on the computer on which Connect-It is installed.

Remarks concerning the configuration of the Asset Management applications

amdb.ini file (AssetCenter version 4.4)

If you use AssetCenter version 4.4, you must edit the **amdb.ini** file which describes the connections and add the following parameter: **UseSql92Join=1**.

Amdb.ini file (versions before AssetCenter 4.2.1)

This file contains the list of database connections created on a computer on which the Asset Management application is installed. This file is created automatically when the Asset Management application is installed. It is located in the Windows system folder.

Example

```
C:\WINNT\amdb.ini
```

When you perform the configuration of the Asset Management connector, Connect-It reads this file and proposes the list of available connections. If your computer does not have access to an Asset Management application, you cannot use the Asset Management connector.

amdb.ini files (from version 4.2.1 of AssetCenter)

From the version 4.2.1 of AssetCenter, the available connections on a computer are declared in several **amdb.ini** files. These files are located in the:

- Windows system folder if it is a system connection.
- One of the sub-folders of **C:\Documents and Settings** with the name of the **Windows** users that created the connection, if it is a user connection.

Example: C:\Documents and Settings\Doe\amdb.ini.

Features tables

Features allow you to extend the Asset Management data-model.

In Connect-It, features appear as fields or collections in the document types published by the Asset Management connector.

Example :

For example: In the **amAsset** document type, the version of DOS used by an asset (a computer, in this case) appears in the **fv_DOSVersion** field.

To fully integrate data from Desktop Discovery and Network Discovery, you must make sure that the features involved in the data transfers have been created correctly in your Asset Management database. Otherwise, these values given in the documents cannot be imported.

The Connect-It datakit includes a text file (**stdfeat.txt**) that you must import into your Asset Management database. Importing this file automatically creates records in the Features table.

For information about the import process in AssetCenter, refer to the "Database administrator" guide.

Mandatory fields in an Asset Management database

In an Asset Management application, a given field or link may be mandatory by default or have been customized this way by the administrator of the Asset Management application.

In the case of reconciliation, each structure published by the Asset Management application corresponds to a record. If an element in this structure is a mandatory field and is not populated, the structure is rejected.

Example :

For example: In the Assets table, the **AssetTag** field is mandatory. If the element that represents it in a document type corresponding to the Assets table is not populated, the entire structure is rejected.

API: DLL

To be able to use the Asset Management connector correctly, you must perform a full installation of AssetCenter, AssetCenter Cable and Circuit, or InfraCenter or a customized installation in which the "API" package is selected.

Managing batches

Special mapping is used for elements managed in batches. The mapping of monitors is performed in the **AddOn** collection and its nature is **MON**. To ensure that identical monitors attached to a given computer are not merged into a single record, this nature is managed as a batch of monitors (the value of the management constraint is 'Asset tag'). In mappings of out-of-the-box scenarios for the AssetManagement connector, the Code element of the **AddOn** collection is mapped with the monitor's SerialNumber field so that the merge mechanism does not apply.

UNIX installation

If you wish to use an AssetCenter database in UNIX, you must follow the installation procedures given in the AssetCenter **Installation** guide, **Installing and configuring in UNIX**.

 **Note:**

Make sure you fully respect the procedures concerning the declaration of environment variables and the creation of the connections file (**amdb.ini**).

Once your AssetCenter database is correctly installed, proceed as follows:

- 1 Develop your scenario in Windows then adapt your Asset Management connector connection sets to your UNIX configuration
 - 2 Once your connections are declared and your scenario is created, save it in the **ConnectIt/ scenario** folder of the Windows folder
 - 3 Copy the scenario saved in your **ConnectIt/ scenario** in UNIX
 - 4 You can then create a service (demon) and put your scenario into production
- For more information on creating a UNIX service, refer to the **User Guide**, chapter **Putting a scenario into production**, section **Creating a service**.

Remarks on AssetCenter migration

Connect-It is used in conjunction with specialized AssetCenter migration tools that not only transfer data but modify the database structure as well. You must not use Connect-It by itself to convert the old format AssetCenter production database.

Limitations of the Asset Management connector












- The Asset Management applications do not process dates less than (before) 1970 or greater than (after) 2038.
If a scenario between an Asset Management connector and another connector uses a date that does not match these criteria (such as a date used in a mapping script), an error message will appear during the scenario's execution.
- The syntax used in the WHERE and ORDER BY clauses must not include square brackets [] for field names.

Configure the Asset Management connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 3.1. Configure the Asset Management connector

 Name and describe the connector [page 17]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 67]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Manage transactions (advanced mode) [page 38]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 69]
 Define document types (advanced mode) [page 42]
 Parallelize consumption (advanced mode) [page 44]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

Warning:

This page is only displayed if the **Display advanced configuration wizards** option is selected. (See **Options** in the **Edit** menu.)

Dynamic library

This option enables you to specify the full path of a DLL containing the API associated with your Asset Management application. Example: for AssetCenter

3.51, C:\Program

Files\Peregrine\AssetCenter\bin32\aampi35.dll. By default, the connector automatically locates the DLL to use.

Expose the primary key of the tables

If you select this option, the document types published by the Asset Management connector contain fields that correspond to the primary keys of the database tables.

Example: If you select this option, the **IAstId** fields appears in the **amAsset** document type published by the connector. The identifier of each record in the Assets table is the primary key of this table.

Expose the foreign keys of the tables

If you select this option, the document types published by the Asset Management connector contain fields that correspond to the foreign keys of the database tables.

Executing reconciliation scripts

To use this page, refer to chapter [Connector directives](#) [page 47], section [Consumption directives - Reconciliation scripts tab](#) [page 60].

Executing an initial import

This option enables you to avoid SELECT queries in reconciliation scripts for initial imports, i.e. when the database is empty or the items to be inserted do not yet exist. This shortens the processing time.

No SELECT will be performed on:

- a root element
- a 1-1 link if the **Follow the link** option is enabled and the parent document does not exist
- A 1-N link if the parent element was just inserted

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Manage transactions (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Manage transactions \(advanced mode\)](#) [page 38].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

This page enables you to specify details for certain elements of consumed document types. An advanced option enables you to define full name structures (**FullName**) for each document type, only if the full names used are not the standard names. For information, a **FullName** field is a unique field generated by the application.

For example, for the Employees table (**amEmplDept**), the option is populated as follows:

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

The structure defined in the value field of the option is the following:

- **IParentId**: Hierarchic field
- **Name**: Hierarchic field used to create the hierarchy
- **bDepartment**: Test field enabling you to know whether the record is last record linked to the full name
- **[Name], [FirstName], [IdNo]**: Structure of the last record of the full name. The fields must be in square brackets [...].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Production directives of the Asset Management connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

For more information about AQL syntax, refer to the appendix [AQL queries](#) [page 427].

The following table lists the available operators and operands for these two clauses.

Operators
=
<>
<
>
AND
OR
NOT
LIKE (special character: %)

Operand type	Examples
Number	1 (without quotes)
Character	'a' (single quotes)
Date	'01/12/24' (international standard)

Operand type	Examples
String	'string' (single quotes)

Sysdate field

In the WHERE clauses written in AQL, the use of the **sysdate** field must be replaced with the function **getdate()**.

Example: The SQL query: **SELECT * from AmAsset where dinstall>sysdate** must be replaced with the AQL query **SELECT * from AmAsset where dinstall>getdate()**.

Example: If you wish to recover all record modified since less that a day:
DaysDiff(GetDate(), dtLastModif) <= 1

Outer joins

If the produced document type contains a non-exclusive link (corresponding to the **LEFT OUTER JOIN** SQL function) then the **Make an outer join** is available. This option enables you to recover all the records from a table for which the link is non-exclusive.



Note:

This option only functions with AssetCenter 4.3 and higher.

For example, for a mapping between the Computers table and the Employees table recovering all the computers assigned to an employee, the behavior is the following:

- Option not selected: Only the computers assigned to an employee are recovered
- Option selected: All the computers are recovered including those that are not assigned

Follow link

This option can be enabled in the **Reconciliation** tab for any selected structure.

This option is useful in two cases:

- For overflow tables

The specificity of these tables is that they are linked to the reference tables by a 1-1 link. For this reason, using a reconciliation key is optional. Following the link of a item in a reference table enables you to retrieve information from this linked table.

- When you cannot define a reliable reconciliation key (for example, if you cannot perform a reconciliation using the Employee ID but only with their Name and First Name).

The **Follow link** option is particularly useful for the Portfolio Items table (amPortfolio) and enables you to follow the link between this table and the Assets table (amAsset) without having to define a reconciliation key for a given element.

 **Note:**

We recommend using this option in **Update or insert mode**.

Collection-to-collection mapping

When mapping N-N links (collection-to-collection mapping), the **Use the parent ID as a reconciliation key** option is selected by default. Unselect this option in order to avoid errors with this type of mapping.

Consumption directives of the Asset Management connector

For information about how to enter a connector's consumption directives, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51].

To set a connector's consumption directives, you must enter reconciliation parameters in the **Reconciliation** and **Advanced reconciliation** tabs.

For more information about these reconciliation tabs, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51], sub-section [Reconciliation](#) [page 52].

Document type produced by the AssetManagement connector

The document type produced by the AssetManagement connector exposes AssetCenter tables.

In particular, the produced document type accounts for the normalization functionality developed for AssetCenter version 4.4.

This functionality can be seen in the scenario provided for the Peregrine Desktop Inventory 8 connector.

Figure 3.1. PDI 8 - AM mapping

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

Since AssetCenter version 4.4, it is possible to associate software installation models and their names with the results of an inventory from an external application; For example, this enables you to associate the names of software models (from an inventory tool) with their internal names in AssetCenter (from a catalog, for example). This is defined in the **amInventModel** table and is based on a reconciliation key created from the name of the external application and a unique identifier of the software (PDI|1234, for example). This key also includes the name of the software and its version.

The scenario **pdi/pdi8ac44/pdiac.scn** uses this new normalization functionality. The software model is mapped to the **InventModel** link in the **AddOn** collection. Each unknown inventory model is recorded in the Software Installations table (amSoftInstall) and linked to a temporary model **sysUNKNOWN_SOFT** thus creating a normalization proposal; The normalization proposal is processed using a wizard. In this way, you associate a temporary model with an existing definitive model in AssetCenter. Once the normalization process is completed, the update is propagated to all inventoried applications linked to the computer.



Note:

You must create a reconciliation script including the **PifIgnoreSubDocumentReconc** function when performing updates so that the definitive model does not get reset to **sysUNKNOWN_SOFT**.

The reconciliation cache

The reconciliation cache reduces the number of database queries. The cache is stored in memory.

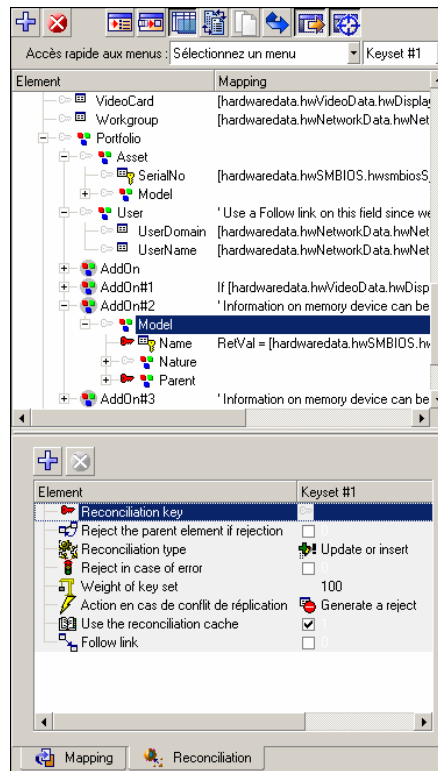
The reconciliation cache is useful when:

- constants are used in a mapping
- when the mapping concerns relatively small tables such as the Models (amModel) or Natures (amNature) tables
- when the mapping concerns tables that will not be updated

When a record is found in the reconciliation cache, it is assumed that the contents of the document has not changed; As a consequence, this record is not updated.

The cache is purged when the maximum number of documents is reached. This number is defined in the connector options (**Edit/Options/Connector/Maximum number of documents in the reconciliation cache** menu).

The reconciliation cache is useful in conjunction with inventory tools such as Peregrine Desktop Inventory, for example.



In the `inventorySrc-amComputerDst` mapping of the scenario `pdia.scn`, the reconciliation cache is used for the **Model** structure. We assume that the AssetCenter models themselves remain unchanged but the related information changes (in this case, the asset).

Out-of-the-box scenarios - Asset Management connector

The following out-of-the-box scenarios use this connector:

- [Action Request System scenarios](#) [page 376]
- [Asset Insight scenarios](#) [page 378]
- [Asset Management scenarios](#) [page 379]
- [Enterprise Discovery scenarios](#) [page 381]

- E-mail scenarios [page 379]
- iInventory scenarios [page 383]
- OpenView Inventory Manager Scenarios [page 385]
- LANDesk scenarios [page 386]
- LDAP scenarios [page 389]
- Peregrine Network Discovery scenarios [page 394]
- MQSeries scenarios [page 390]
- NT Security scenarios [page 392]
- RDS scenarios [page 399]
- Peregrine Desktop Inventory scenarios [page 396]
- SAP IDOC scenarios [page 399]
- SAP BAPI scenarios [page 402]
- SMS scenarios [page 405]
- ServiceCenter scenarios [page 406]
- TS Census scenarios [page 418]
- Tivoli scenarios [page 420]
- Unicenter AMO scenarios [page 424]
- Web Service scenarios [page 425]
- Winpark Actima scenarios [page 425]

Desktop Administration 6.x connector

Base connector

- Production (source)
- Consumption (destination)

This section presents the Desktop Administration 6.x connector.

About the Desktop Administration 6.x connector

The Desktop Administration 6.x lets you process the data contained in a Desktop Administration 6.x database.

Connect-It uses the DLLs and APIs provided with these applications.

The Desktop Administration 6.x application client must be installed on the computer on which Connect-It is installed.

Remarks on the configuration of Desktop Administration 6.x

Amdb.ini file (versions before 4.0.1 of Desktop Administration)

This file contains the list of database connections created on a computer on which the Desktop Administration 6.x application is installed. This file is created automatically when the Desktop Administration 6.x application is installed. It is located in the Windows system folder.

Example

```
C:\WINNT\amdb.ini
```

When you perform the configuration of the Desktop Administration 6.x connector, Connect-It reads this file and proposes the list of available connections. If your computer does not have access to an Desktop Administration 6.x application, you cannot use the Desktop Administration 6.x connector.

amdb.ini files (from version 6.0.1 of Desktop Administration)

From the version 6.0.1 of Desktop Administration, the available connections on a computer are declared in several **amdb.ini** files. These files are located in the:

- Windows system folder if it is a system connection.
- One of the sub-folders of **C:\Documents and Settings** with the name of the **Windows** users that created the connection, if it is a user connection.

Example: C:\Documents and Settings\Doe\amdb.ini.

Mandatory fields of a Desktop Administration 6.x database

In an Desktop Administration 6.x application, a given field or link may be mandatory by default or have been customized this way by the administrator of the Desktop Administration 6.x application.

In the case of reconciliation, each structure published by the Desktop Administration 6.x application corresponds to a record. If an element in this structure is a mandatory field and is not populated, the structure in which it is found is rejected.

API: DLL

To be able to easily use the Desktop Administration 6.x connector, you must perform a complete or custom installation of the Desktop Administration 6.x application in which you select the **API** package.

Known limitations of the Desktop Administration 6.x connector

The Desktop Administration 6.x applications do not process dates less than (before) 1970 or greater than (after) 2038.












If a scenario between a Desktop Administration 6.x connector and another connector uses a date that does not match these criteria (such as a date used in a mapping script), an error message will appear during the scenario's execution.

Configure the Desktop Administration 6.x connector

A wizard enables you to configure this connector. For more information about configuring connectors, refer to the section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 3.2. Configure the Desktop Administration 6.x connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 79]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Manage transactions (advanced mode) [page 38]
 Configure the cache (advanced mode) [page 39]
 Define document types (advanced mode) [page 42]
 Parallelize consumption (advanced mode) [page 44]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

Warning:

This page is only displayed if the **Display advanced configuration wizards** option is selected. (See **Options** in the **Edit** menu.)

Dynamic library

This option enables you to specify the full path of the DLL containing the APIs associated with your Desktop Administration 6.x application.

Expose the primary key of the tables

If you select this option, the document types published by the Desktop Administration 6.x connector contain fields that correspond to the primary keys of the database tables.

Expose the foreign keys of the tables

If you select this option, the published document types will include the fields corresponding to the foreign keys of the database tables.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Manage transactions (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Manage transactions \(advanced mode\)](#) [page 38].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Production directives of the Desktop Administration 6.x connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

For more information about the AQL syntax, refer to the appendix [AQL queries](#) [page 427].

The following table lists the available operators and operands for these two clauses.

Operators
=
<>
<
>
AND
OR
NOT
LIKE (special character: %)

Operand type	Examples
Number	1 (without quotes)
Character	'a' (single quotes)
Date	'01/12/24' (international standard)
String	'string' (single quotes)

Sysdate field

In the WHERE clauses written in AQL, the use of the **sysdate** field must be replaced with the function **getdate()**.

Example: The SQL query: **SELECT * from AmAsset where dinstall>sysdate** must be replaced with the AQL query **SELECT * from AmAsset where dinstall>getdate()**.

Consumption directives of the Desktop Administration 6.x connector

For information about how to enter a connector's consumption directives, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51].

To set a connector's consumption directives, you must enter reconciliation parameters in the **Reconciliation** and **Advanced reconciliation** tabs.

For more information about these reconciliation tabs, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51], sub-section [Reconciliation](#) [page 52].

Out-of-the-box scenarios - Desktop Administration 6.x connector

There is no scenario available for this connector.

Enterprise Discovery connector

Base connector

◆ Production (source)

This section describes the Enterprise Discovery connector

For additional information on customizing and using the Enterprise Discovery tool, refer to the separate documentation.

The Enterprise Discovery connector allows to process a Network Discovery database.





An Enterprise Discovery database contains information about the enterprise's entire IT portfolio. For each computer stored in the database, the out-of-the-box scenarios transfer the data concerning those records to an AssetCenter or ServiceCenter database.






Configure the Enterprise Discovery connector

A wizard allows you to configure the connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16]

The following table lists all the pages of the connector configuration wizard.

Table 3.3. Configure the Enterprise Discovery connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]

-  [Configure the reconnection parameters \(advanced mode\) \[page 31\]](#)
-  [Determine server delay \(advanced mode\) \[page 33\]](#)
-  [Configure schedule pointers \(advanced mode\) \[page 36\]](#)
-  [Configure the cache \(advanced mode\) \[page 39\]](#)
-  [Define document types \(advanced mode\) \[page 42\]](#)

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

- **Server field:** specify the port used to connect to the server.
If Connect-It was installed on a workstation other than the Enterprise Discovery server, populate this field with the DNS or the IP address of the Enterprise Discovery server; the DNS name or the IP address comes before the colon. For example:
- **User field:** Specify the login. You are advised to use the **admin** login, which has all authorizations. The login used must have the required read and write authorizations; the login profile is defined in the Enterprise Discovery Web user interface.
- **Password field:** specify the password for this login.
- **Database field:** specify the name of the database used. The default name is **Aggregate**.

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

 Note:

The query enabling you to recover the current date is included in the application code and may not be edited.

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

 Note:

The current date of the PND server is used as the upper limit by default.

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Define document types (advanced mode)

Fill in the extension file according to your version of Enterprise Discovery.

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Published document-type

The Enterprise Discovery connector publishes only one **Devices** document type. This document type publishes the inventory information contained in an Enterprise Discovery database.

Out-of-the-box scenarios - Enterprise Discovery connector

The following out-of-the-box scenarios use this connector:

- ◆ Enterprise Discovery scenarios [page 381]

Peregrine Desktop Inventory 7.x (PDI) connector

Base connector

◆ Production (source)

About the Peregrine Desktop Inventory 7.x connector

The PDI connector processes the XML files obtained using Peregrine Desktop Inventory scans. These files can be compressed in GZ files, which are provided in the Connect-It datakit (`[[Connect-It installation folder]\datakit\pdi*.xml.gz`).

Configure the PDI connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 3.4. Configure the PDI connector

☰	Name and describe the connector [page 17]		
☰	Selecting a connection protocol [page 86]		
☰	FTP server [page 86]	☰	Local or network file(s) [page 87]
☰	Select files or folders [page 87]		
☰	Define the post-processing actions [page 88]		
☰	Choose a DTD/XSD [page 89]		
☰	Configure the JVM [page 89]		



Path of a file or folder

Often when configuring a connector, you must indicate the path of a file or folder in the configuration wizard. If this file or folder is in a folder associated with a network drive on your computer, you must never indicate the letter of this drive; only indicate the folder associated with this letter.

Indicating the network drive in the path stops the scenario from working correctly when it is associated with a service. For more information about this, consult the **User's guide**, chapter **Putting a scenario into production**, section **Creating a Connect-It service**, sub-section **Creating a Connect-It service in Windows/Scenario path**.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Selecting a connection protocol

The available options are:

- FTP server
- Local or network file(s)

FTP server

If you have chosen to read the text files on an FTP site, you must:

- 1 Select the FTP connection parameters
- 2 Select an action to perform after the files have been processed.

FTP server connection

On this page, you must populate three fields, which enable you to connect to your FTP server:

- Server
Enter the name of your FTP server
- Login
Enter the login that enables you to access the chosen site.
- Password

- Enter the password corresponding to your login.
- Connect in passive mode
Select this option to connect the server in passive mode.
 - Test the connection
This button enables you to test the validity of your connection.
-

 **Note:**

During a connection in FTP mode, you might see the following error: `ERROR: 12015`.

This error means that the directory is not available, usually because of too many simultaneous connections.

Select files or folders

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select files or folders](#) [page 27].

Define post-processing actions

After a file is read by the PDI connector, Connect-It gives you three options:

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

If you choose this last option, you must indicate the path of the folder to which you want to move the processed file.

You must specify one of these options in the case of failure, and one in the case of success, in processing the text files by your connector.

Local or network file(s)

If you choose to read local or network files, you must:

- 1 Choose a file or folder
- 2 Select an action to perform after the files have been processed.

Choose a file or folder

This page enables you to select the file(s) that your connector must read. You can choose from:

- **Read a file**

- **Read files in a folder**

Read a file

If you select this option, the wizard will ask you to enter the path of the text file on your computer or your network.

Read files in a folder

If you select this option, the wizard will ask you to populate the following fields:

- **Folder name**

Enter the path of the folder that contains your files.

- **Extension**

Indicate the file name extension that the connector must read. By default, the value of these fields is **gz**. These files are compressed files that contain XML files corresponding to scans done by Peregrine Desktop Discovery.

Read the sub-folders

If you select this option, the connector will also read the files in the sub-folders of the selected folder.

Select files or folders

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select files or folders](#) [page 27] .

Define the post-processing actions

After a file is read by the connector, Connect-It gives you three options:

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

If you choose this last option, you must indicate the path of the folder to which you want to move the processed file.

You must specify one of these options in the case of failure, and one in the case of success, in processing the text files by your connector.

In order to apply a post-processing action to the documents consumed by the other connectors and mapping boxes of your scenario, you must use the

processing reports that each one produces. For more information about processing reports, refer to the **User's guide**, chapter **Processing reports**.

Using post-processing actions

To use post-processing actions, you must:

- ◆ Create a mapping between the **UrlFileInfo.Path** structure of the document type produced by the source connector and the **UrlFileInfo.Path** element of the **SuccessReport** document type.

Choose a DTD/XSD

To process the XML files corresponding to the Peregrine Desktop Inventory scans, the connector must use a DTD or an XSD.

A default DTD is provided with Connect-It (**[Connect-It installation folder]\config\pdi\pdi.dtd**).

Configure the JVM

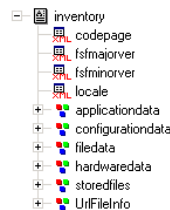
This page is only visible if you have selected an XSD in the [Choose a DTD/XSD](#) [page 94] page of the configuration wizard.

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Produced document-type

The PDI connector produces only one document type whose elements are presented in the following screen capture:

Figure 3.2. PDI connector - produced document-type



Scheduling the Peregrine Desktop Inventory connector

AssetCenter Server enables you to schedule the recovery of the .xml.gz files recovered by Desktop Inventory .

For further information on scheduling scenarios, refer to the AssetCenter **Administration Guide**, chapter **AssetCenter Server**, section **Configuring the modules monitored by Connect-It Server**.

Out-of-the-box scenarios - PDI connector

The following out-of-the-box scenarios use this connector:

- ◆ [Peregrine Desktop Inventory scenarios](#) [page 396]

Peregrine Desktop Inventory 8.x connector (PDI)

Base connector

◆ Production (source)

About the Peregrine Desktop Inventory 8.x connector

The PDI connector processes the XML files obtained using Peregrine Desktop Inventory scans. These files can be compressed in GZ files, which are provided in the Connect-It datakit (**[Connect-It installation folder]\datakit\pdi*.xml.gz**.)








Configure the PDI connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 3.5. Configure the PDI connector

 [Name and describe the connector](#) [page 17]

-  [Selecting a connection protocol \[page 86\]](#)
 -  [FTP server \[page 86\]](#)  [Local or network file\(s\) \[page 87\]](#)
 -  [Select files or folders \[page 88\]](#)
 -  [Define the post-processing actions \[page 88\]](#)
 -  [Choose a DTD/XSD \[page 89\]](#)
 -  [Configure the JVM \[page 94\]](#)
-

 **Warning:**

Path of a file or folder

Often when configuring a connector, you must indicate the path of a file or folder in the configuration wizard. If this file or folder is in a folder associated with a network drive on your computer, you must never indicate the letter of this drive; only indicate the folder associated with this letter.

Indicating the network drive in the path stops the scenario from working correctly when it is associated with a service. For more information about this, consult the **User's guide**, chapter **Putting a scenario into production**, section **Creating a Connect-It service**, sub-section **Creating a Connect-It service in Windows/Scenario path**.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors \[page 15\]](#), section [Name and describe the connector \[page 17\]](#).

Selecting a connection protocol

The available options are:

- FTP server
- Local or network file(s)

FTP server

If you have chosen to read the text files on an FTP site, you must:

- 1 Select the FTP connection parameters
- 2 Select an action to perform after the files have been processed.

FTP server connection

On this page, you must populate three fields, which enable you to connect to your FTP server:

- Server
Enter the name of your FTP server
 - Login
Enter the login that enables you to access the chosen site.
 - Password
Enter the password corresponding to your login.
 - Connect in passive mode
Select this option to connect the server in passive mode.
 - Test the connection
This button enables you to test the validity of your connection.
-

 **Note:**

During a connection in FTP mode, you might see the following error: `Error : 12015.`

This error means that the directory is not available, usually because of too many simultaneous connections.

Select files or folders

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select files or folders](#) [page 27] .

Define post-processing actions

After a file is read by the PDI connector, Connect-It gives you three options:

- Leave it in the folder
- Delete it from the folder
- Move it to another folder
If you choose this last option, you must indicate the path of the folder to which you want to move the processed file.

You must specify one of these options in the case of failure, and one in the case of success, in processing the text files by your connector.

Local or network file(s)

If you choose to read local or network files, you must:

- 1 Choose a file or folder
- 2 Select an action to perform after the files have been processed.

Choose a file or folder

This page enables you to select the file(s) that your connector must read. You can choose from:

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

Read a file

If you select this option, the wizard will ask you to enter the path of the text file on your computer or your network.

Read files in a folder

If you select this option, the wizard will ask you to populate the following fields:

- **Folder name**
Enter the path of the folder that contains your files.
- **Extension**
Indicate the file name extension that the connector must read. By default, the value of these fields is **gz;xsf**. These files are compressed files that contain XML files corresponding to scans done by Peregrine Desktop Discovery.

Read the sub-folders

If you select this option, the connector will also read the files in the sub-folders of the selected folder.

Select files or folders

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select files or folders](#) [page 27] .

Define the post-processing actions

After a file is read by the connector, Connect-It gives you three options:

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

If you choose this last option, you must indicate the path of the folder to which you want to move the processed file.

You must specify one of these options in the case of failure, and one in the case of success, in processing the text files by your connector.

In order to apply a post-processing action to the documents consumed by the other connectors and mapping boxes of your scenario, you must use the processing reports that each one produces. For more information about processing reports, refer to the **User's guide**, chapter **Processing reports**.

Using post-processing actions

To use post-processing actions, you must:

- ◆ Create a mapping between the **UrlFileInfo.Path** structure of the document type produced by the source connector and the **UrlFileInfo.Path** element of the **SuccessReport** document type.

Choose a DTD/XSD

To process the XML files corresponding to the Peregrine Desktop Inventory scans, the connector must use a DTD or an XSD.

A default DTD is provided with Connect-It (**[Connect-It installation folder]\config\pdi\pdi.dtd**).

Configure the JVM

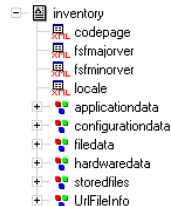
This page is only visible if you selected an XSD in the [Choose a DTD/XSD](#) [page 94] page of the configuration wizard.

To use this option, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Produced document-type

The PDI connector produces only one document type whose elements are presented in the following screen capture:

Figure 3.3. PDI connector - produced document-type



Scheduling the Peregrine Desktop Inventory connector

AssetCenter Server enables you to schedule the recovery of the .xml.gz files recovered by Desktop Inventory .

For further information on scheduling scenarios, refer to the AssetCenter **Administration Guide**, chapter **AssetCenter Server**, section **Configuring the modules monitored by Connect-It Server**.

Out-of-the-box scenarios - PDI connector

The following out-of-the-box scenarios use this connector:

- ◆ [Peregrine Desktop Inventory scenarios](#) [page 396]

Peregrine Network Discovery connector

Base connector

◆ Production (source)

This section presents the Network Discovery connector

The Network Discovery connector enables you to process Network Discovery databases.











A Network Discovery database contains information related to a company's computer population. The scenarios transfer the information concerning the computers in the database to an AssetCenter or ServiceCenter database.

Configure the Peregrine Network Discovery connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 3.6. Configure the Peregrine Network Discovery connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

 Note:

The query enabling you recover the current date is included in the application code and may not be edited.

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

 Note:

The current date of the PND server is used as the upper limit by default.

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

Populate the extension file according to your version of Peregrine Network Discovery:

- Specify the **config/pnd/pnddb51.cfg** file in your Connect-It folder for version 5.1
- Specify the **config/pnd/pnddb52.cfg** file in your Connect-It folder for version 5.2

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Published document-type

The Peregrine Network Discovery connector publishes a single document type: **Devices**. This document type publishes the inventory information contained in the Peregrine Network Discovery database.

Out-of-the-box scenarios - Peregrine Network Discovery connector

The following out-of-the-box scenarios use this connector:

- ◆ [Peregrine Network Discovery scenarios](#) [page 394]

BizDoc connector

Base connector

- Production (source)
- Consumption (destination)

The BizDoc connector enables you to dialog with the Peregrine Systems web client application server.

Known limitations of the BizDoc connector







Errors are saved to log files at the server level and not locally.

Configure the BizDoc connector (read)

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 3.7. Configure the BizDoc (read) connector

 Name and describe the connector [page 17]
 Choosing a processing mode [page 99]
 Configure BizDoc server connection [page 99]
 Choose a DTD/XSD [page 100]
 Determine server delay (advanced mode) [page 101]
 Configure schedule pointers (advanced mode) [page 36]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Choosing a processing mode

This page enables you to indicate if the connector is used in read or write mode.

Configure BizDoc server connection

In this page you must indicate the HTTP connection parameters:

- Address
Enter an address of the type: **[protocol] ://[address] : [port]/[path]**.
The [address] part is sometimes the only mandatory part.
The usual port number for an HTTP server is 80.
- Login
Enter the login that enables you to access the chosen site.
- Password
Enter the password corresponding to your login.

Secured connection (HTTPS protocol)

This option enables you to indicate if you want to connect to your site via a secured connection (HTTPS).

Important:

If the [protocol] part of the Web site's address does not corresponds to the HTTPS protocol, the selection of the **Secured connection (HTTPS protocol)** option forces the use of an HTTPS protocol.

The usual port number for an HTTPS server is 443.

Example of an HTTP address

Address	Secured connection (HTTPS protocol)	Connection mode
https://a.b.c.d/	Selected	https port 443
https://a.b.c.d:9000/	Selected	https port 9000
http://a.b.c.d:9000/	Not selected	https port 9000
http://a.b.c.d:9000/	Selected	https port 9000
https://a.b.c.d:9000/	Not selected	https port 9000
http://a.b.c.d:443/	Not selected	https port 443
https://a.b.c.d:80/	Selected	https port 80

Script

This field is populated automatically.

The **.cgi** script is provided by your BizDoc administrator

Paging options

The **Paging size** and **Start of first page** options enable you to define the size of sub-queries when querying the BizDoc server.

Do not verify the server identity

Use this option to support a proxy server in a secure connection.

If a proxy server is installed, the address shown in the server certificate does not match the one shown for the proxy server. Selecting this option disables the automatic verification of the server's identity.

Choose a DTD/XSD

To process a file, the connector must use a DTD (Document Type Definition) or an XSD (Extended Schema Definition). This page enables you to specify (using the ***.xsd** command) the DTD or the XSD used in the **DTD/XSD** field.

 **Note:**

The DTD is provided by the server administrator.

Determine server delay (advanced mode)

In the **Script** field, specify the `.cgi` script used to give the local time on the server.

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)







To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the BizDoc connector (write)

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 3.8. Configure the BizDoc (write) connector

 Name and describe the connector [page 17]
 Choosing a processing mode [page 101]
 Configure BizDoc server connection [page 101]
 Choose a DTD/XSD [page 102]
 Determine server delay (advanced mode) [page 33]
 Parallelize the consumption of the documents [page 45]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Choosing a processing mode

This page enables you to indicate if the connector is used in read or write mode.

Configure BizDoc server connection

On this page, you must populate three fields enabling you to connect to the HTTP server on which you want to write files.


- Address
This field is mandatory. Enter an address of the type:
[protocol] ://[address] : [port]/[path].
- Login
Enter the login that enables you to access the chosen site.
- Password
Enter the password corresponding to your login.

Secured connection (HTTPS protocol)

To use this option, refer to the section [Secured connection \(HTTPS protocol\)](#) [page 277].

Choose a DTD/XSD

In the **Select a DTD/XSD** page, indicate the DTD or XSD used (using the ***.xsd** command) in the **DTD/XSD** field. The document types published by the BizDoc connector correspond to this XSD.

By clicking , the **File location** page is displayed. This page enables you to enter the path of your DTD or XSD depending on your connection protocol.



Note:

The DTD is provided by the server administrator.

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Parallelize the consumption of the documents

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize the consumption of the documents](#) [page 45].

Published document types

The BizDoc connector publishes the following structure for all document types:

- ◆ UriFileInfo

This structure is mainly used for post-processing actions.

The **UrlFileInfo** structure contains the following fields:

- **creationdate**
This field corresponds to the creation date of the document
- **lastmodificationdate**
This field corresponds to the last modification date of the document
- **lastmodificationdate**
This field corresponds to the line number of the document

SCAuto listening connector

Optional connector

- Production (source)
- Consumption (destination)

The SCAuto listening connector emulates a ServiceCenter SCAutomate server (SCAuto). It can receive events and process client requests.

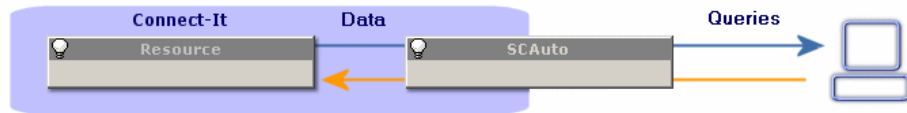
SCAuto listening connector in event-receiving mode

The SCAuto listening connector in event-receiving mode receives events that enable it to produce documents each time that it is started in manual or scheduled mode. This connector uses communication by socket.



SCAuto listening connector in client-request-processing mode

Using the connector in client-request-processing mode depends on the use of a resource connector that processes requests addressed to your Connect-It server by clients.



Limitations of the SCAuto listening connector

This connector does not support the following adapters:

- The Japanese version of the SCAuto OpenView Network Node Manager adapter version 3.x.
- SCAuto Email adapter
- All the adapters using the following commands:
 - QUERY2
 - INSERTBLOB
 - CREATEBLOB
 - DESCRIBE OBJECT
 - SELECT OBJECTS
 - END SELECT OBJECTS
 - GETNEXT OBJECT
 - STORE OBJECT


If the SCAuto listening connector supports the DELETE command of the SCAuto SDK, its use does not allow the deletion of events. To delete events, you must use processing reports. For more information about processing reports, refer to the **User's guide** chapter **Processing reports**.


Configure the SCAuto listening connector


A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 3.9. Configure the SCAuto listening connector

-  Name and describe the connector [page 17]

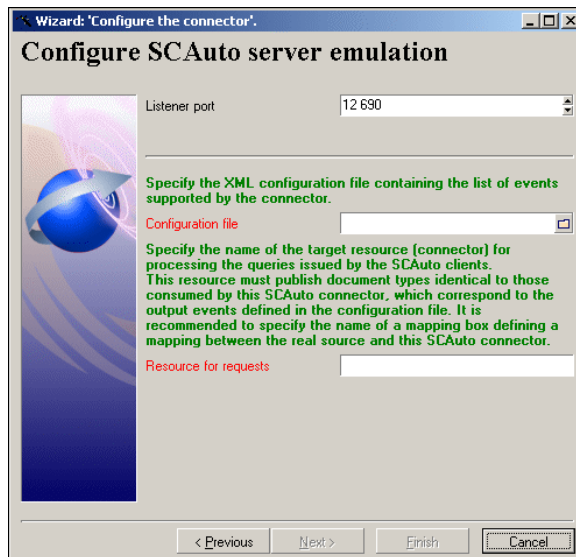
-  Configure SCAuto server emulation [page 105]

-  Configure the JVM [page 106]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Configure SCAuto server emulation



This page enables you to enter the connection parameters to your SCAuto server.

Listening port

Specify the listening port that enables the connector to receive client requests. This field is mandatory.


Configuration file

Indicate the path of the XML configuration file containing the descriptions of the document types that the SCAuto listening connector publishes.

This field is mandatory.

These document types correspond to the SCAuto event types. A Connect-It scenario enables you to obtain this configuration file from ServiceCenter event types.

To obtain the configuration file:

- 1 Launch the Scenario builder.
- 2 Open the **scacfg.scn** scenario.
This file is located in the following file: **[Connect-It installation folder]\scenario\scauto\scacfg**.
- 3 Configure the scenario's ServiceCenter connector.
- 4 Open the scenario connectors by clicking .
- 5 Start all the schedulers (Ctrl + F5).

The scenario creates the XML file in the **[Connect-It installation folder]\scenario\scauto\scacfg** file.

Resource for the requests

Specify the name of the tool that processes the requests sent to your Connect-It server.

Configure the JVM

This page is only visible if you selected an XSD in the [Choose a DTD/XSD](#) [page 94] page of the configuration wizard.

To use this option, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Additional information

Modifying the SCAuto listening-connector configuration file

Modifying the SCAuto listening-connector configuration file enables you to support more (or less) event types. You can also decide to process character-string collections as Memo-type fields.

The configuration file contains two sections:

- **InputEventTypes**

Events consumed by the SCAuto listening connector.

The consumed events correspond to those requested by the SCAuto adapters.

- **OutputEventTypes**

Events produced by the SCAuto listening connector.

The produced events correspond to those produced by the SCAuto adapters.

An event type is represented as a structure having the same name as the event type. This structure contains sub-nodes representing the fields of that event.

Four types of simple fields are supported:

- String (Character)
- Byte (Logical)
- Long (Number)
- TimeStamp (Date)

The types in parentheses are the names of the equivalent types in ServiceCenter. There are several ways in which you can represent array-type fields: as a collection of simple fields (which is the default behavior of the **scacfg.scn** scenario); as a simple Memo-type field whose value is a paragraph (inside which each array element value will be represented by a line). **Attention:** Only the character string arrays should be represented in this way. If this is not the case, the array elements will still be considered as character strings.

A simple field is characterized by an index, a name and a type.

Example

```
ATTRIBUTE index="11" name="orig.operator" type="String"/
```

An array field is characterized by an index, a name, a separator, an element name and an element type. The name of the element is arbitrary and can be different from that of the field.

Example

```
< COLLECTION index="2" name="comments.2" separator="|" >
< ATTRIBUTE name="comments" type="String"/>
< /COLLECTION>
```

In this example, the array field can also be represented by the following line:

```
< ATTRIBUTE index="2" name="comments.2" type="Memo" separator="|" />
```

In this last representation, a three-element array ("abc", "def" and "ghi") will give the attribute value:

```
"abc
def
ghi"
```

If the **separator** attribute is not there, or if it is empty, the default separator '|' will be used.

There are two particular separators: **lf** and **cr**. They are interpreted as **end of line** and **carriage return**, respectively. With the exception of these two particular separators, a separator must be composed of one single character other than ^.

Out-of-the-box scenarios - SCAuto listening connector

The following out-of-the-box scenarios use this connector:

- `scauto\scacfg\scacfg.scn` [page 415]
- `scauto\sca-sc\sca-sc.scn`

For this scenario, you must use the **BasicScaCfg.xml** configuration file during the configuration of the scenario. This file is located in the same folder as the scenario.

RDS connector

Base connector

- Production
- Consumption

This section presents the RDS connector

Known limitations












We highly recommend using the connector with an ODBC driver and ODBC Administrator using identical version levels of their DLL files.

Configure the RDS connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 3.10. Configure the RDS connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Manage transactions (advanced mode) [page 38]
 Configure the cache (advanced mode) [page 39]
 Define document types (advanced mode) [page 42]
 Parallelize consumption (advanced mode) [page 44]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Manage transactions (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Manage transactions \(advanced mode\)](#) [page 38].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Production directives of the connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Sysdate field

In the WHERE clauses written in AQL, the use of the **sysdate** field must be replaced with the function **getdate()**.

Example: The SQL query: **SELECT * from AmAsset where dinstall>sysdate** must be replaced with the AQL query **SELECT * from AmAsset where dinstall>getdate()**.

For more information about the AQL syntax, refer to the appendix [AQL queries](#) [page 427].

NULL

When a "Numeric" type field is not populated (its value is NULL), Connect-It sets its value to "0". Similarly, the absence of a link is comes out as "Link = 0" or "foreign key = 0". Example: "Location=0" or "lLocald=0".

Consumption directives of the RDS connector

For information about how to enter a connector's consumption directives, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51].

To set a connector's consumption directives, you must enter reconciliation parameters in the **Reconciliation** and **Advanced reconciliation** tabs.

For more information about these reconciliation tabs, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51], sub-section [Reconciliation](#) [page 52].

Published document-type

The document types published by the RDS connector correspond to tables in the RDS database.

Provided scenarios - RDS connector

The following out-of-the-box scenarios use this connector:

- ◆ [RDS scenarios](#) [page 399]

ServiceCenter connector

Base connector

- Production (source)
- Consumption (destination)

This section describes the ServiceCenter connector.

About the ServiceCenter connector

The ServiceCenter connector enables you to obtain data from or insert data into a ServiceCenter database.

Within a same given scenario, you can use multiple ServiceCenter connectors connected to a same server or different versions of a server.

Remarks on the configuration of ServiceCenter

Creating input events

The ServiceCenter connector cannot directly consume documents because ServiceCenter does not authorize the direct updating of records for data integrity reasons. Instead, a corresponding input event must be created in ServiceCenter. When the documents are consumed, this event is placed in an event queue. Next, a mapping in ServiceCenter maps the elements of this event to fields in ServiceCenter.

Example :

By default, there is no input event corresponding to the description of software installed on a computer.

In order for information concerning software to be taken into account by the ServiceCenter connector, you need to:

- 1 Create a synchronous input event for these software items (called **pcsoftware** in the supplied scenario).
 - 2 Map the fields of the input event to those in the **pcsoftware** table.
-

The Connect-It datakit includes a file (**Evsoft.un1**) that allows you to perform these two actions automatically in ServiceCenter.

An **UNL** file is available for each out-of-the-box scenario involving ServiceCenter from **Peregrine Systems** technical support.

For details on how to import these files, refer to the ServiceCenter documentation.

Input event synchronization

Input events in ServiceCenter are processed either synchronously or asynchronously. In asynchronous mode, ServiceCenter defines how the event queue is managed. In synchronous mode, the events are processed as soon as they are inserted in the ServiceCenter event queue.

When developing your scenarios, use the synchronous mode if you want to verify immediately that your scenario is working.

Selecting the asynchronous processing mode prevents you from immediately verifying in ServiceCenter that the data has been processed correctly: A warning message is issued to indicate that the document is pending processing. To verify the errors, verify the input queues in ServiceCenter.

In synchronous mode, it indicates whether a document has been processed successfully or rejected.

Note:

Performance is improved when working in asynchronous mode (improvements in excess of 50%).

It is recommended using synchronous mode if performance levels are deemed suitable.

Processing an synchronous mode will produce warnings in the document log. To disable these warning, refer to section [Warning for events in asynchronous mode](#) [page 127].

Join files

In ServiceCenter, join files enables you to expose fields from multiple table within one single screen.

For example, the **Device** and **Computer** tables contain fields exposed in the **joincomputer** join file.

These join files are declared in a configuration wizard and are visible in the produced document types.

Using the ServiceCenter connector in Unix

To use the ServiceCenter connector in Unix:

- 1 Create the following text files:
 - **sc.ini**
 - **sc.log**
- 2 Write **log=sc.log** in the **sc.ini** file.
- 3 Place these two files in the **[Connect-It installation folder]/bin** folder.










Configure the ServiceCenter connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

Certain options concerning this connector can be enabled using the **Edit/Options/ Connector** menu. For more information, refer to the **User's Guide**, chapter **Scenario builder**, section **Connector**.

The following table presents all the pages in the connector's configuration wizard.

Table 3.11. Configure the ServiceCenter connector

 Name and describe the connector [page 17]
 Define the connection parameters [page 114]
 Advanced configuration (advanced mode) [page 115]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Define document types (advanced mode) [page 42]
 Parallelize consumption (advanced mode) [page 44]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Define the connection parameters

Server name: Enter the name of the server as follows:

Example :

[Server name].[ServiceCenter client port number]

SC.peregrine.com.12670

 Note:

When ServiceCenter is installed, **12670** is assigned as the default value for the client server number.

Login

Enter the login required to connect to the ServiceCenter server. The profile of this login must allow you to execute the actions performed by your scenario (reading data or sending input events).

Password

Enter the password associated with the login.

Test

This button enables you to test your connection.

To test your connection:

- Enter your connection parameters.
- Click **Test**.
The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.
- Click **Close** to return to the configuration wizard.

Advanced configuration (advanced mode)

This page enables you to set the following parameters:

- List of virtual tables: This field enables you to select the linked tables to be exposed for reading and defined in the **joindefs** table. By default, all the linked tables starting with **join** are exposed.
Enter a list of values separated by semi-colons (;). Wildcard characters (*, \$, ?) are authorized.

 **Note:**

If an event references a linked table in a definition of a structure collection, then the table is automatically loaded (and exposed to be read), even if this not been specified in the connector configuration wizard.

- Format for date and time type fields of input events: Select this option if the format of input events is different from that of the server.
- Format for date and time type fields of output events: Select this option if the format of output events is different from that of the server.

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Production directives of the ServiceCenter connector

These directives use WHERE and ORDERBY clauses, which allow you to filter the produced documents.

The following table lists the available operators and operands for these two clauses.

Operators

 =

 ~=

 >

 <

 AND

 OR

 NOT

 #

Operand type

 Number

 Character

 Date

Examples

 1 (without quotes)

 "a" (double quotes)

 '2002-01-10' (international standard)

String

"string" (double quotes)

Example :

The devicepc document type represents the equivalent table in ServiceCenter in order for the connector to only produce documents representing Compaq computers:

- In use since January 1, 2001
- Still on the network monitored by Network Discovery

Use the following WHERE clause:

Example :

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

For the ORDERBY clause, you must indicate which fields are to be used to perform an ascending sort (alphanumeric) of the documents. Separate these fields with commas. For example: For a produced e-mail document-type, you can sort these documents by recipient, then author, with the following ORDERBY clause:

```
user.to, user.from
```

Additional information

This section presents additional information about the connector.

scdb.cfg file

The ServiceCenter database descriptions are contained in the an **scdb.cfg** file. In this section, when you are asked to modify this file, you must modify the file corresponding to the ServiceCenter database used:

- **scdb3.cfg**: For a ServiceCenter 3 database
- **scdb4.cfg**: For a ServiceCenter 4 database
- **scdb5.cfg**: For a ServiceCenter 5 database
- **scdb51.cfg**: For a ServiceCenter 5.1 database
- **scdb6.cfg**: For a ServiceCenter 6 database

Deleting events processed successfully by the ServiceCenter connector

Deleting events processed successfully by the ServiceCenter connector enables you to purge the ServiceCenter queues.

Example: the ServiceCenter connector processes output events transmitted to the E-mail connector (sending).

To delete these events, you must use the processing reports produced by each connector.

For more information about processing reports, refer to **User's guide**, chapter **Processing reports**.

Example of using processing reports with ServiceCenter

To follow this example, you must ask Peregrine Systems technical support for the following files:

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

UNL files

UNL files enable you to create the **ConnectItDel** input event.

To load UNL files into ServiceCenter:

- 1 Open a ServiceCenter session as an administrator.
- 2 Open the Database Manager.
- 3 Select the **Options/Import/Load** menu.
- 4 Load the following files:
 - **delevreg.unl**
 - **delevmap.unl**

These files enables you to create the **ConnectItDel** input event. This event lets the ServiceCenter connector delete PMO-type output events in the queue by identifying them with the **evsysseq** field.

To create a **PMO**-type output event in ServiceCenter:

- 1 Open a ServiceCenter session as an administrator.
- 2 Create an event in the **eventout** queue by performing the following actions:
 - Select the **Utility programs** tab.
 - Click **Event services**.
 - Select the **Queues** tab.
 - Click **Output event**.

- Write the event and specify that it is a **PMO**-type event.
- Click **Add**.

Processing reports

The **delete.scn** scenario uses:

- The ServiceCenter connector as the source to process PMO-type output events.
- A mapping box to transform the documents produced by the ServiceCenter connector.
- The Asset Management connector as the destination to create records in the Helpdesk tickets table (amTicket).
- A mapping box to consume the processing reports produced by the Asset Management connector.

If the document consumed by the Asset Management connector was correctly processed, the processing report sent to the ServiceCenter connector via the second mapping box enables the deletion of the **PMO**-type event in the queue.

The mapping between the processing report produced by the Asset Management connector and the **ConnectItDel** document type enables the deletion of the event.

The following table presents the mapping between the processing report of the Asset Management connector and the ServiceCenter connector.

Table 3.12. Mapping between the processing report produced by the Asset Management connector and the **ConnectItDel** document type from the ServiceCenter connector.

Element of the ConnectItDel document type	Element of the processing report or script
evsysseq	[\$ParentDoc\$.SParentDoc\$.EventFormatFields.evsysseq]
<p>Remarks: This script enables the recovery of a unique identifier (evsysseq field) that tells ServiceCenter that the event has been deleted from the queue.</p> <p>For more information about how to use the \$ParentDoc\$ variable, refer to the User's guide, chapter Processing reports, section Using processing reports in predefined scenarios, note Using the \$ParentDoc\$ variable.</p>	
evtype	"pmo"
<p>Remarks: This mapping enables you to secure the reconciliation using the event type. This is the pmo event type in our example.</p>	

Inserting attachments in the ServiceCenter database

For a document consumed by the ServiceCenter connector to be able to insert attachments in a target database, you must modify the document type on which it is based (document type corresponding to an input event).

To modify this document type, you must edit the `scdb.cfg` file located in the following sub-folder: `[Connect-It installation folder]/config/sc`.

For a document type to support the processing of attachments, you must:

- Add it to a collection relating to the attachments.
- Initialize the parameters linking these attachments stored in the `SYSBLOB` file to the target record.

Two cases arise:

- 1 The attachment is associated with an event.

In this case, the ServiceCenter event manager associates the attachment to the target record.

- 2 The attachment is inserted directly in the `SYSBLOB` file.

Modifying the `scdb.cfg` file

This modification is common to two types of attachment processing.

In the following sections, the extract of code shows you how the `pmo` document type (help ticket creation) is modified so that it supports attachment processing. Example of an attachment: a screen capture attached to a helpdesk ticket.

Step 1: Declare the document type (structure) corresponding to the event

The following code corresponds to the declaration of the `pmo` document type:

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

To indicate that the document type does not behave like the other document types, you must declare it as an exception in the `AllTables` structure. Like this:

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

Another solution is to create a variable containing the list of events for which the attachments are processed. Then, you have to reference this variable in the declaration of exceptions:

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

Step 2: Add the corresponding collection to the attachments

Adding an **attachments** collection in the document-type declaration enables it to process the attachments. This collection must have the following elements:

- A **name** field corresponding to the name of the attachment. (Example: **sc.ini**).
This mandatory field must not contain the file path.
- A **blob** field corresponding to the binary contents of the attachment.

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

The **CIRCULAR** property enables you to change a template that will add the name and blob fields to the **attachments** collection.

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

Step 3: Configure the attachments

```
{ STRUCT pmo
{ ARRAY attachments

[...]

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

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

```
[...]
```

MODEOUT

The value **0** enables you to indicate that the attachments can only be processed in consumption mode the value **1** enables you to indicate that the attachments can also be processed in production mode. This is notably the case for document types available in both production and consumption.

NODETYPE

Enables you to indicate that the **Attachments** collection manages binary-type fields (BLOB-type fields for Connect-It). This element is mandatory and must have **BLOB** for its value.

BLOBTYPE

This parameter enables ServiceCenter to classify these BLOB-type fields. The value **5** corresponds to an attachment, the only BLOB-type field supported by ServiceCenter in insertion.

BLOBFORMAT

This parameter enables ServiceCenter to specify a BLOB-type file storage format. For the attachments, this parameter must be **SC** for its value.

INSERTBLOB

This parameter indicates whether there is an insertion or an insertion/replacement of attachments in the target record. The value by default, **1**, corresponds to the insertion only.

APP

Indicates the ServiceCenter file in which the attachment is inserted. The name of the file is contained in the **application** field of the **SYSBLOB** file.

Figure 1: ServiceCenter associates attachments with an event

If your version of ServiceCenter enables you to associate attachments with an event, the event manager automatically makes the association between the SYSBLOB and the target record. To find out if ServiceCenter supports this, an **attachments** tab appears when you consult the event queue in the manager. In this case, the reconciliation between the target record and the SYSBLOB file is based on the **number** field of the target record.

The following extract of code shows the modification of the pmo document type so that it supports the processing of attachments.

```
{ STRUCT pmo
NODETYPE = EVENT
```

```

{ ARRAY attachments
[...]
// Insert the attachments into the event
BLOBRECONCTYPE = EVENT
APP           = eventin
}

{ ATTRIBUTE AllFields
}

```

BLOBRECONCTYPE

The **EVENT** value indicates that the attachment is associated with the event.

APP

The **eventin** value indicates in the **SYSBLOB** file the table in which the attachment is inserted.

The event manager then replaces the eventin name by the name of the table of the record associated with the event. Example: the **problem** table during the creation of a **pmo** event.



Note:

As soon as the association between attachments and event is possible, you must use this type of processing.

The configuration of **pmo pmu**, **pmc** and **smin** events in the **scdb.cfg** file enables you to use this type of processing.

Figure 2: The topic field helps define the join between the SYSBLOB file and the target record

When ServiceCenter cannot associate attachments to an event, you must specify how the value of the **topic** field is obtained. This SYSBLOB file field defines the join with the target record.

The following contexts are possible:

- 1 The value of the **topic** field is contained in the document consumed by the ServiceCenter connector.

Example: In the consumed document-type corresponding to an **ICMpc** event (event concerning the computers), the **logical.name** field enables you to create the link between the **SYSBLOB** field and the target record of the **devicepc** in the ServiceCenter database.

The value of this field is directly extracted from the fields mapped in the event, using the following syntax:

```
{ STRUCTURE ICMpc
[...]
```

```
{ STRING logical.name
MANDATORY = 1
}
```

```
{ ARRAY attachments
[...]
```

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

```
}
```

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

```
}
```

BLOBRECONCTYPE

The **INTERNAL** value of this parameter indicates that the value of the **topic** field is contained in the fields of the document consumed by the connector.

PIFLINK

The `@{..field_name}` value enables you to recover the value of a field in the document consumed by the connector. The path of the field is relative to the **attachments** collection in the example.

The points (..) enable you to indicate that you must go up a level to find the **logical.name** field.

Like in the file path, each set of points (..) corresponds to a level above the current one. Thus, `@{...field_name}` enables you to go up two levels.

Note:

If a field contains one point (.), its name must be surrounded by single quotation marks ('.'). Example: `PIFLINK = @{'..'logical.name'}`

In our example, the **logical.name** is made mandatory because its value is necessary to insert the attachment.

- 2 The value of the **topic** field is obtained using a request sent by the ServiceCenter database.

In this case, a field different from the **topic** field is used to perform the reconciliation of records. **Example:** the **contact.name** field is necessary to

populate the **topic** field, but only the **first.name** and **last.name** fields are present in the consumed document.

```

{ STRUCTURE eventcontacts
[...]
```

```

{ STRING last.name
MANDATORY = 1
}

{ ARRAY attachments
[...]
```

```

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

```

}

{ ATTRIBUTE AllFields
EXCEPTION = last.name
}
}

```

BLOBRECONCTYPE

The **QUERY** value indicates that the value of the **topic** field is obtained using a query performed on the ServiceCenter database.

TOPICFIELD

In the example, the value **contact.name** indicates that this field is used to populate the value of the **topic** field.

PIFLINK

In this example, the value **@{..'last.name'}** indicates the value used in the WHERE clause, sent to the ServiceCenter database.

Note:

This processing mode requires the event be treated in a synchronous way.

- 3 The value of the **topic** field can be obtained after the document is processed by the ServiceCenter event manager.

Important: This processing mode does not apply when your version of ServiceCenter does not support the association between event and attachments.

This value is obtained when the event's **evid** field is read. The title of this field is **Problem ID** or **Incident ID**, depending on the version of ServiceCenter you use.

This processing mode corresponds to the creation of a work order (**event pmo**) or a helpdesk call (**event smin**).

```
{ STRUCT pmo
[... ]

{ ARRAY attachments
[... ]

BLOBRECONCTYPE = EVID

[... ]
}

[... ]
}
```

BLOBRECONCTYPE

The value **EVID** indicates the value of the **topic** field is obtained using the **evid** field of the event.

The **PIFLINK** parameter is not necessary.



Note:

This processing mode requires the event be treated in a synchronous way.

Warning for events in asynchronous mode

When events configured in asynchronous mode are used to insert data in ServiceCenter, Connect-It cannot verify if these events were correctly processed.

In this case, a warning message appears:

- In the Connect-It log when the session is opened.
- In the Document log for each document processed.

In order that these messages no longer appear:

- 1 Select the **Edit/ Options** menu.
- 2 Unfold the **ServiceCenter** node in the **Connector** section.
- 3 Assign the value **No** to the **Display a warning for events used in asynchronous mode** option.
- 4 Click **OK**.

Interpretation of text fields

In a ServiceCenter database, long text fields are held in the form of an array. Connect-It interprets these arrays and brings together the lines of text in a single long text field.

If you wish to interpret these arrays as collections, for example a list of values from the local.software computers table, you must modify the **sc.cfg** file located in the **Connect-It \config\sc** application folder.

- 1 Edit the **sc.cfg** file
- 2 Search for the **NotMemo** section

The first line of each section contains the name of the table for which the processing exception is to be applied. This table name is prefixed with the asterisk character (*) in order to take into account other tables containing the same field.

For example:

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

Here, the **PC** table contains three fields that will be processed as collections.

- 3 Add the sections and fields to not be interpreted and then save.

Note:

Syntax used: Field and table names are case sensitive. If the field or the table name contains a period, the whole name must be included in single quotes '(...)'.

Interpretation of date and time fields

In a ServiceCenter database, date and time fields are held in the form of an array. Connect-It interprets these arrays and presents the date and time fields in the form of a collection.

These date and time fields pose problem with they express a duration. Connect-It cannot differentiate between a field containing a date and a field containing a duration.

To interpret a date and time field as a duration, you must modify the **scdbN.cfg** file (where N is the version number of ServiceCenter) as shown in the following example:

- 1 Make sure that the table or event is correctly declared.

```
// List of tables/events having a processing to add a link or process b
lob or have duration field
#define LINK_TABLES ocmo, eventregister, contacts, problem, enclapplica
tion, application, probsummary, incidents, ocml, rmlin, clocks
#define LINK_DEVICES device, deviceparent, computer, joincomputer, dis
playdevice, joindisplaydevice, furnishings,
joinfurnishings, handhelds, joinhandhelds, mainframe, joinmainframe, ne
tworkcomponents, joinnetworkcomponents, officeelectronics,
joinofficeelectronics, softwarelicense, joinsoftwarelicense, storage, j
oinstorage, telecom, jointelecom
#define EVENT_ATTACHMENT ICMcomputer, ICMapplication, ICMdisplaydevice
, ICMfurnishings, ICMhandhelds,
ICMmainframe, ICMnetworkcomponents, ICMofficeelectronics, ICMsoftwareli
cense, ICMstorage, ICMtelecom, pmo, pmu, pmc, smin
```

- 2 If the table or the event is not declared as an exception, add the following script:

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

{ ATTRIBUTE AllFields
}
}
```

- 3 Declare the duration field as a long integer with the DURATION property:

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

{ LONG handle.time
DURATION = 1
}

{ ATTRIBUTE AllFields
EXCEPTION = handle.time
}
}
```

- 4 If the ServiceCenter connector is open, close it and then open it again. The **handle.time** field now appears as a 32-bit long integer and no longer as a date and time field.
- 5 If the fields has already been mapped, you must:
 - In production mode: Edit the document type, delete the field before adding it again.
 - In consumption mode: Edit the mapping, delete the field from the mapping before adding it again.

Out-of-the-box scenarios - ServiceCenter connector

The following out-of-the-box scenarios use this connector:

- Enterprise Discovery scenarios [page 381]
- OpenView Inventory Manager Scenarios [page 385]
- iInventory scenarios [page 383]
- LANDesk scenarios [page 386]
- MQSeries scenarios [page 390]
- Peregrine Desktop Inventory scenarios [page 396]
- Peregrine Network Discovery scenarios [page 330]
- RDS scenarios [page 399]
- ServiceCenter scenarios [page 406]
- Tivoli scenarios [page 420]
- TS Census scenarios [page 418]

ServiceCenter Web Service connector

Optional connector

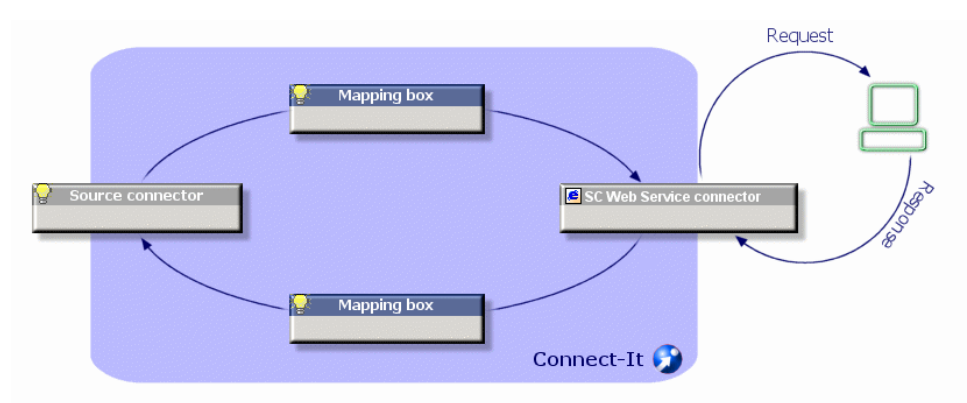
- Consumption (destination)
 - Consumption - spontaneous production (request - response)
-

The ServiceCenter Web Service connector lets you interact with a ServiceCenter 6.1 Web service.

In an integration scenario, the ServiceCenter Web Service connector consumes a document and sends it as a query to a Web service, consumes a document,

and sends it as a query to a Web service, then receives a reply which it automatically transforms into a produced document.

Figure 3.4. ServiceCenter Web Service connectors - Type of document consumed - produced



Configure the ServiceCenter Web Service connector

A wizard enables you to configure the connector.

The following table lists all the pages of the connector configuration wizard.

Table 3.13. Configure the ServiceCenter Web Service connector

Name and describe the connector [page 17]
Define the connection parameters [page 132]
SSL configuration (advanced mode) [page 132]
Configure the JVM [page 132]
Parallelize consumption [page 132]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Define the connection parameters

Use this page to indicate the connection parameters of the connector to the ServiceCenter Web Service connector.

- Server name: name of the ServiceCenter server in computer:port format
- Service name: name of the ITIL ServiceCenter, incident management, configuration management, problem management, change management, service desk and service level management services.
- Login: ServiceCenter login
- Password: password for this login

Select the **Use an SSL connection** option if you want to use a secure connection.

SSL configuration (advanced mode)

This page allows you to define:

- the Java certificates file for the secure connection (**CA certificates file**).
By default, Connect-It searches for information concerning the Java certificate file in the JRE, in the following files: /lib/security/jsse.cacerts and /lib/security/cacerts.
- the private key file (**Certificates and client keys file (keystore)**). This file is required if the private key is stored in a database.
- the password for the private key file (**Keystore password**).

Configure the JVM

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Parallelize consumption

To use this page, refer to chapter [Configuring connectors](#) [page 15].

Out-of-the-box scenarios

The following out-of-the-box scenarios use this connector:

- ◆ [ServiceCenter Web Service scenarios](#) [page 416]



4 Application connectors

CHAPTER

The application connectors are those which enable you to process the data from external applications or specific operating systems.

Action Request System connector

Base connector

- Production (source)
- Consumption (destination)

The Action Request System connector enables you to process data from an Action Request System database.

Prerequisites

Action Request System connector: dynamic libraries

To use the Action Request System connector, you must make sure that the path of these libraries is defined in the appropriate environment variable (%PATH% in Windows, \$LD_LIBRARY_PATH in Linux and Solaris, \$LIBPATH in AIX).

Known limitations of the Action Request System connector

In read (production) and write (consumption) mode, the Action Request System connector processes the following Action Request System-type data fields:

- Text
- Diary

In read mode (production), this field enables you to recover all the history available for this field.

In write mode (consumption), this field enables you to insert a string containing the name of the user who writes the data and the time and date it is written.

- Integer
- Real number
- Radio button

To use this field, you must indicate the number of the element that you want to initialize. You cannot directly indicate its value.

- Drop-down list
- Date and time
- Attachment

The connector publishes a structure for each attachment. This structure presents the following three fields:

- **FileName**

Contains the full path of the attachment.

- **AttachMemoVal**

In the case of an attachment file, this field contains the contents of the file.

- **AttachBlobVal**

This file always contains the binary contents of the attachment.

In write mode (consumption), you must map the full path of the attachment to the **ARS attachment FileName** field.

If you want to create a copy of the memo or Blob file in a shared folder, you must map the path of the shared folder to the **ARS attachment FileName** field. Then you need to map the contents of the file to the **AttachMemoVal** or **AttachBlobVal** field.

 Note:

Since the version 5.0 of the Action Request System, the memo and blob files are stored directly in the database. You no longer need to have a copy of these files in a shared folder.

In read mode (production), you must perform the following mappings:

- **AttachMemoVal** field to a **memo**-type field
- **AttachBlobVal** field to a **Blob**-type field
- **FieldName** field to a **Blob**-type field

In the mapping edit window, you must make sure the **Load Blob from file** option is selected.

For more information about this option, refer to the **User's guide**, chapter **Mapping document types**, section **Mapping destination document types to source document types/Mapping Blob-type fields**.

Unsupported field types

- Data types containing BLOB fields (**AR_DATA_TYPE_BYTES**)
- Attachments (**AR_DATA_TYPE_ATTACH**)
- The **STAT_HISTORY**, **VALUE_SET**, **LOCAL_VARIABLE**, and **QUERY** fields

Configure the Action Request System connector





A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].





The following table presents all the pages in the connector's configuration wizard.

 Note:

Certain pages in this wizard are not accessible in advanced mode.

Table 4.1. Configure the Action Request System connector

 Name and describe the connector [page 17]
 Define the connection parameters [page 136]
 Advanced configuration (advanced mode) [page 29]
 Configure the cache (advanced mode) [page 39]

-  [Configure the reconnection parameters \(advanced mode\) \[page 31\]](#)
-  [Determine server delay \(advanced mode\) \[page 33\]](#)
-  [Configure schedule pointers \(advanced mode\) \[page 36\]](#)
-  [List of last-modification fields of the schemas \(advanced mode\) \[page 137\]](#)

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Define the connection parameters

These pages enable you to enter the connection parameters to your Action Request System server.

Server name

Enter the name of your server. You can either enter the name of your server on the network or its IP name.

Connection port

Enter the server connection port number. If you do not specify a value, the default connection port is used.

RPC number

Enter the RPC number if you are using a specific server.

Login

Enter your Action Request System user name. Your profile must allow you to execute the actions performed by your scenario (reading or writing data).

Password

Enter the password associated with the login. This password is not visible when entered and is stored in an encrypted form.

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Dynamic library to use

Enter in this field the full path to your dynamic library (**arapi45.dll**) that enables you to connect to the Action Request System server. You must use a version 5.0 or higher of these libraries.

Format of the pointer

In this field, enter the name of the user format to apply to the pointer.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

List of last-modification fields of the schemas (advanced mode)

The Action Request System schemas have a field indicating the last-modification date of a record in a schema. In most cases, the **Modified-date** field is used. Nevertheless, depending on your schemas and your version of Action Request System, the name of this field can vary. In this case, the editable zone that appears on this page enables you to indicate:

- The document type published by the connector corresponding to an Action Request System schema, for which you must specify a last-modification date field.
- The name of the last-modification date field used in the Action Request System schema.


The default value of the last-modification date field is **Modified_date**. This value is used for all document types in which you do not enter values in the editable zone.

You can specify a different name for the last-modification date field in each document type published by the Action Request System connector.


 **Warning:**

You must not use the name of the last-modification field indicated in the database, but rather the one used in the Action Request System scenario.

To enter a new association schema/ last-modification date field:

- 1 Click .
- 2 Click the activated text zone in the **Diagram** column and enter the name of the document type (schema) published by the Action Request System connector.
- 3 Click in the corresponding **Field** column and enter the name of the last modification field used in this schema.

To delete an existing field:

- 1 Select the line corresponding to this field.
- 2 Click .

Production directives - Action Request System connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

The following table lists the available operators and operands for these two clauses.

Operators

=

<>

<

Operators

>
 AND
 OR
 NOT
 LIKE (special character: %)

Operand type	Examples
Number	1 (without quotes)
Character	'a' (single quotes)
Date	'01/12/24' (international standard)
String	'string' (single quotes)

Consumption directives of the Action Request System connector

For information about how to enter a connector's consumption directives, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51].

To set a connector's consumption directives, you must enter reconciliation parameters in the **Reconciliation** and **Advanced reconciliation** tabs.

For more information about these reconciliation tabs, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51], sub-section [Reconciliation](#) [page 52].

Additional information - Action Request System connector

This section presents additional information about the connector.

Names of the fields in the document types published by the Action Request System connector

- If you connect to an ARS version 5 server, the attachments are directly saved on the server.
- In the document types published by the Action Request System connector, the "#" character is replaced by the "_" character.

Example: the **Field#1** field is displayed as **Field_1** in the published document type.

In both cases, the queries using either **champ#1** or **champ_1** will produce the same result. In effect, Action Request System uses the displayed name or the internal Ids indifferently.

- In the WHERE or ORDERBY clauses:
 - To use the display name of a field, use the following syntax:

```
[DisplayName]
```

- To specify the system name of a field, use the following syntax:

```
[DatabaseName]
```

Out-of-the-box scenarios - Action Request System connector

The following out-of-the-box scenarios use this connector:

- [ars\rsm\rsm4ac36\cmposcn](#) [page 377]
- [ars\rsm\rsm4ac36\sharedat.scn](#) [page 377]

TEDW integration scenarios

The TEDW integration scenarios are located in the Connect-It installation folder, **tivoli\tedwExcept**. These scenarios are documented individually in the **Doc** folder of Connect-It.

Lotus Notes connector

Optional connector

- Production (source)
 - Consumption (destination)
-

The Lotus Notes connector enables you to:

- In production mode, insert data from a Lotus Notes database into an external application (Example: a ServiceCenter database).
- In consumption mode, insert data from an external application into a Lotus Notes database.

 Note:

To use the Lotus Notes connector, you must correctly install and configure the Lotus Notes client on the machine where Connect-It is installed.

In particular, the PATH in Windows must include the path of the Lotus Notes folder.

Known limitations of the Lotus Notes connector

The Lotus Notes connector enables you to process the following field types only:

- Text
- Text list
- Number
- Date
- Lotus Notes Rich Text Format

This format enables you to preserve the formatting options used in Lotus Notes documents. In this case, all links to Notes documents and attachments are lost. Only the page format is conserved.

- Attachments






Configure the Lotus Notes connector

Before modifying the connector, you must first declare in the system environment variables the access path of the `vim32.d11` file in the Lotus folder.


A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 4.2. Configure the Lotus Notes connector

 Name and describe the connector [page 17]
 Define the connection parameters [page 142]
 Configure the reconnection parameters (advanced mode) [page 31]
 Configure schedule pointers (advanced mode) [page 36]
 Advanced options (advanced mode) [page 144]

 [Define document types \(advanced mode\) \[page 42\]](#)

 [Parallelize consumption \(advanced mode\) \[page 44\]](#)

Name and describe the connector

To use this page, refer to chapter [Configuring connectors \[page 15\]](#), section [Name and describe the connector \[page 17\]](#).

Define the connection parameters

The second page of the **Basic configuration of the connector** wizard enables you to configure the Lotus Notes connection.

The four fields of this page enable you to specify a:

- Lotus Notes identification file (**.id** extension)
- A password
- Lotus Notes server
- Lotus Notes database

Lotus Notes ID

Indicate the **.id** file used to connect to your Lotus Notes database. This file contains your Notes ID and your password. Example: `C:\Program Files\notes\data\user.id`. This identification file must allow you to perform the required actions (read, write, delete) on the database selected in the **Database** field.

Warning:

This file must be the one used on the Lotus Notes client machine on which Connect-It is installed.

Password

Enter the password needed to connect to a Lotus Notes database. This password is not visible when being entered and is stored in an encrypted format.

The **nextpwd.dll** DLL file enables this password to be memorized and avoids you having to enter it each time a scenario is launched which includes the Lotus Notes connector.

On installing the Lotus Notes connector, this DLL is located in the **bin 32** folder of the Connect-It installation folder.

In order not to request the password each time the connector starts, follow these steps:

- 1 add the Connect-It bin32 folder to the Windows environment variables,
- 2 Add the following parameter to the notes.ini file after the [Notes] section:

```
EXTMGR_ADDINS=nextpwd
```

- 3 save your changes and restart Connect-It.



Note:

If you encounter conflicts with other programs, rename the **nextpwd.dll** file and update the **notes.ini** file as described previously.

Server name

Select the server on which your database is installed. This field should display the servers to which your computer has access. However, if the name of the server that you want to use does not appear in this list, you can enter its address manually into the field. If you want to connect to a database installed on your computer, select **Local**.

Database

Enter the Lotus Notes database that you want to use in your Connect-It scenario. The entries in the drop-down list of the **Database** field:

- Depend on the chosen server name.
- Correspond to the databases contained in the **data** folder of your Lotus Notes installation folder, if you have selected **Local** in the **Server name** field.

If the database of your choice is not shown in this field, you can enter it manually. In this case, you must indicate the **full path** of this database.

View all database fields option

This option enables the connector to expose all the fields of a Lotus Notes database. If you do not select this option, only those fields declared in the forms are present in the document types published by the connector.

This option exposes the **\$FILE** collection, which allows you to manage attachments.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configure the reconnection parameters \(advanced mode\)](#) [page 31], section [Test connection query](#) [page 32].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Advanced options (advanced mode)

This page enables you to give specific information for certain produced document type elements.

The status of a text field in Lotus Notes can change depending on what it contains:

- A text type field containing a character string is interpreted as an text type attribute.
- A text field that contains a character string containing semi-colons as separators is interpreted as a collection of attributes.

This page of the Lotus Notes connector configuration wizard enables you to identify the attributes that will be exposed as a collection of attributes:

- The **Name** column contains the full path in Connect-It (to which you must add the name of the document) that identifies the attribute in question.
- The **Value** column contains the **NotesList** string identifying a list and not a single attribute.

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Production directives of the Lotus Notes connector

The production directives of the Lotus Notes connector enable it to filter and sort data from the Lotus Notes database. Data is automatically sorted in Lotus

Notes and depends on the modification date of the documents containing this data.

To filter the data that will appear in the documents produced by the connector, you must write a WHERE clause.

WHERE clause

This clause enables you to filter records in your Lotus Notes database. You can, for example, filter the records extracted from your Lotus Notes database created from January 1, 2000. You just need to create the following clause for the field containing the creation date of a record.

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

The following table lists the available operators and operands for this clause.

Operators	Action
=	Equals
<>	Different than
>	Greater than
<	Less than
&	And
	Or

Operand type	Examples
Number	1 (without quotes)
Character	"a" (double quotes)
Date	[10/10/00] (date corresponding to the regional settings)
String	"string" (double quotes)

Design-form name

Each document type published by the Lotus Notes connector corresponds to a document from your Lotus Notes database. Each of these documents is associated with a form. The name of this form is included as a field of this document.

Sometimes several documents contain the same data but are associated with different forms. (example: a **Supplier** form and a **Contact** form). In order for your connector to produce the document corresponding to the form of your choice, you must select this name from the drop-down field in the **Design-form name** field.

The **@All** form is used to display data that is associated with each of the Lotus Notes forms. Using the **@All** form is the same as not applying a WHERE clause for a specific form.

Consumption directives of the Lotus Notes connector

The production directives of the Lotus Notes connector enable it to reconcile the records contained in the Lotus Notes database with the values of the documents consumed by the connector.

Description of the reconciliation tab

Design-form name

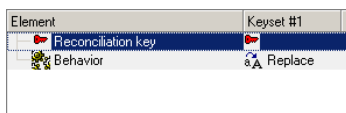
Each document type published by the Lotus Notes connector corresponds to a document from your Lotus Notes database. Each of these documents is associated with a form. The name of this form is included as a field of this document.

Sometimes several documents contain the same data but are associated with different forms. (example: a **Supplier** form and a **Contact** form). In order for your connector to create the document corresponding to the form of your choice, you must select this name from the drop-down field in the **Design-form name** field.

Font

This field enables you to select the character font used to format the data written by the connector.

Reconciliation key



Use this option to indicate whether the current selection is used as a reconciliation key.

To indicate that a field or a structure is a reconciliation key, you must:

- Select this element in the detail window.
- Then, select the **Reconciliation key** option or click directly on the transparent key in the pane showing the consumed-document type.

Case-sensitive reconciliation

For this option, the behavior of Connect-It is as follows:

- **Case-sensitive reconciliation** option selected

The reconciliation key values are case-sensitive. Example: If a field containing the e-mail address of your employees is used as a reconciliation key, the values 'jmartin@company.com', 'JMARTIN@company.com' and 'jMartin@company.com' will correspond to three different records in the destination Lotus Notes database.
- **Case-sensitive reconciliation** option cleared

The reconciliation key values are no longer case-sensitive. Example: If a field containing the e-mail address of your employees is used as a reconciliation key, the values 'jmartin@company.com', 'JMARTIN@company.com' and 'jMartin@company.com' will correspond to the same record in the destination Lotus Notes database.

Key set

→ [Using alternate reconciliation key sets](#) [page 56]

→ **User's Guide**, chapter **Putting a scenario into production**, section **Reconciliation keys**.

Reconciliation type



The options in the **Reconciliation type** field allow you define the reconciliation type for each parent node of your document type.

You can determine the reconciliation type for each non-terminal node of your consumed document-type. To do this:

- 1 Select a non-terminal node in your consumed document-type (root node, structure or collection).
- 2 Select the option of your choice in the **Reconciliation type** field.

The four available options are:

- Update or insert

In this case, the data from the document consumed by the connector makes it possible to insert or update records in the destination Lotus Notes database.

- Update only
In this case, the data from the document consumed by the connector makes it possible to update records in the destination Lotus Notes database.
- Insert only
In this case, the data from the document consumed by the connector makes it possible to insert new records in the destination Lotus Notes database.
- Delete
In this case, the data from the document consumed by the connector makes it possible to delete records from the destination Lotus Notes database.

Additional information about the Lotus Notes connector

This chapter presents additional information about the connector.

The case of binary Lotus-Notes fields

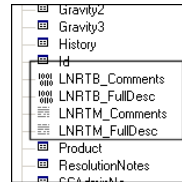
A Lotus Notes database contains documents with binary fields. They contain text and formatting information such as color, font, etc.

In the document types published by the Lotus Notes connector, each of these fields is represented by two distinct fields:

- A variable-length binary field
This field contains text and formatting information. This field is prefixed by LNRTB_ (Lotus Notes Rich Text Binary).
In your mappings, these fields can only be mapped to other binary Lotus Notes fields (in a Lotus Notes to Lotus Notes scenario).
- A long text field
This field contains text only. This field is prefixed by LNRTM_ (Lotus Notes Rich Text Memo).
In your mappings, these fields can be mapped to other text type fields.

 Note:

In the document types consumed by the Lotus Notes connector, only variable-length binary fields are available.



Data processing in scheduled mode

If you use a local copy of a Lotus Notes database, the connector will not process the data modified between two sessions.

In order for the connector to process your Notes documents, you must synchronize your local copy with the Notes server between two scheduled sessions.

In the properties of a Notes document, there are two available modification dates:

- An initial modification date (**initially**).
- A modification date in this file (**in this file**).

Because the connector uses the initial modification date (updated each time the server is synchronized), it does not process the files that only had their 'in this file' date modified while editing the document.

Attachments

You can handle attachments by using the **\$FILE** collection. This collection contains two elements:

- **Content:** Contents of the attached file in binary format
- **Name:** Name of the attachment

Out-of-the-box scenarios - Lotus Notes connector

The following out-of-the-box scenarios use this connector:

- ◆ [ldap\ldapnote\names.scn](#) [page 390]

MQSeries connector

Optional connector

- Production (source)
- Consumption (destination)

MQSeries is an IBM application that enables the unique, asynchronous and stable transmission of data on numerous hardware and software platforms.

MQSeries is an infrastructure for the communication between applications, either on the same machine or on different machines separated by one or more networks.

MQSeries can handle all of the most popular communication protocols and provides routes between networks using different protocols. With the integration scenarios, MQSeries enables us to exchange XML documents.

The bridges and gateways of MQSeries enable you to easily access numerous application systems and environments, such as Lotus Notes, Web browsers and Java applets.

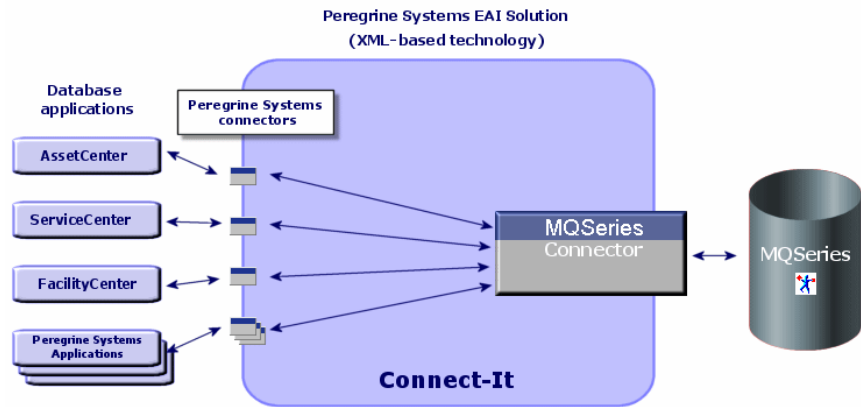
The numerous functions of MQSeries guarantee the transmission of data, even in the case of underlying system or network infrastructure failure.

MQSeries data circulates in the form of **messages** containing the data that is exchanged between different applications.

The messages are stored in data structures called **queues**. Under normal circumstances, messages can be placed in a queue or can be taken from the queue by applications or by a **queue manager**.

External applications can put XML documents in an MQSeries queue. Being connected to the appropriate queue, the MQSeries connector then reads these messages and sends them to the Peregrine Systems applications specified in

the scenario. The exchange of data between MQSeries and Peregrine Systems applications can be bi-directional.



Prerequisites

The MQSeries application client must be installed on the computer on which Connect-It is installed.

Configure the MQSeries connector (read)





The basic configuration of the MQSeries connector in read mode enables you to specify the MQSeries queue in which it must read the data that it then transforms into Connect-It documents.

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 4.3. Configure the MQSeries connector (read)

Name and describe the connector [page 17]
Choosing a processing mode [page 152]
Configure the queue manager connection [page 152]
Choosing a queue [page 153]
Define post-processing actions [page 154]

-  [Choose a DTD/XSD \[page 155\]](#)
-  [Configure the reconnection parameters \(advanced mode\) \[page 31\]](#)
-  [Determine server delay \(advanced mode\) \[page 33\]](#)
-  [Configure schedule pointers \(advanced mode\) \[page 36\]](#)

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Choosing a processing mode

The second page of the **Basic configuration** wizard enables you to choose in which mode you want to use the MQSeries connector. Select the **Read** mode.

Configure the queue manager connection

You must specify the following parameters in order for the connector to connect to a queue manager:

- **Use an extended connection**

This option is selected by default. Clear this option to use a standard connection.
- **Server name**

Enter the DNS name or IP address of your MQSeries server on your network.
- **Connection port**

Enter the listening port of your queue manager. By default, the value of this port is **1414**.
- **Connection channel name**

Enter the name of the connection channel that the queue manager uses as a communication path. The default value of the channel is **SYSTEM.DEF.SVRCONN**.
- **Queue manager**

Enter the name of your queue manager. In order to use the scenarios provided with the MQSeries connector, the name of this manager is **connectit**. If this field is not populated, the MQSeries connector will connect to the default queue manager of MQSeries.
- **Alternate user ID**

Specify an alternate user ID, if required, for opening the queue manager.

You can specify different alternate user IDs for each queue (main, success, failure).

 **Note:**

If you clear the **Use extended connection** default option, only the **Queue manager name** field can be populated.

Test button

This button enables you to test your connection.

To test your connection:

- 1 Enter your connection parameters.
- 2 Click **Test**.

The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.

- 3 Click **Close** to return to the configuration wizard.
-

 **Note:**

When the connector is opened, the operating system of the server is automatically detected.

Choosing a queue

This page enables you to enter the name of the queue from which the MQSeries connector must read the data. To use the scenarios provided with the MQSeries connector, the name of this queue is **PEREGRINE.IN**.

Advanced configuration (advanced mode) option

This option enables you to define, for a queue:

- The name template for creating dynamic queues

Enter the format template for creating the dynamic queue name that will be used: The character '*' is replaced by the queue manager in order to guarantee the uniqueness of the dynamic queue's name.

You can choose from three options for closing dynamic queues:

- **Delete permanent dynamic queue if it is empty (MQCO_DELETE)**
 - Temporary dynamic queues are also physically deleted.

- **Retain dynamic queue if it is permanent and delete it if it is temporary**
- **Purge messages from permanent dynamic queue and delete it (MQCO_DELETE_PURGE)**
Temporary dynamic queues are also physically deleted.
- **Authentication**
For each queue, you can specify an alternate user ID.

Define post-processing actions

This page enables you to indicate how to process the messages from the queue after having been read by the MQSeries connector.

In case of failure or success in processing a message, you have the choice of the following options.

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

When you select this option, you can specify whether you wish to keep the whole context of the message (default behavior), just the identity context, or no context at all.

A context is formed as follows:

- **ID context:** Information on the user inserting the message into the queue
- **Original context:** Information concerning the inserted message (date, application that inserted the message)

In order to apply a post-processing action to the documents consumed by the other connectors and mapping boxes of your scenario, you must use the processing reports that each one produces. For more information about processing reports, refer to the **User's guide** chapter **Processing reports**.

Using post-processing actions

To use post-processing actions, you must:


- ◆ Create a mapping between the **PutDate** and **MsgID** elements of the **MessageInfo** structure of the document type produced by the source connector and the same elements of the **MessageInfo** structure of the **SuccessReport** document type consumed by the same source connector.

Choose a DTD/XSD

To process an XML file, the MQSeries connector must use a DTD (Document Type Definition) or an XSD (XML Schema Definition). This page enables you to specify the DTD or the XSD used in the **DTD/XSD** field.

 **Note:**

If you use the MQSeries connector in Learning mode (see below), the name that you indicate in the **DTD/XSD** field corresponds to a DTD file in which the contents will be created once the learning process is finished. If the name indicated corresponds to an already existing file, Connect-It automatically saves this file and assigns it a number (Example: **request_01.dtd**). If this connector is relaunched several times in Learning mode, the different DTDs created will all have a number corresponding to their order of creation. (Example: **request_01.dtd**, **request_02.dtd**, **request_03.dtd**, etc. The **request_03.dtd** field corresponds to the DTD file that was most recently saved.)

By clicking , the **File location** page is displayed. This page enables you to enter the path of your DTD depending on your connection protocol.

File location

You have the choice of three options:

- **Local/Network**
Enter the full path of your DTD or XSD file located on the computer or network on which Connect-It is installed.
 - **FTP**
Enter the FTP parameters in the **Server**, **Login**, **Password**, and **Path** fields. The **Path** field enables you to specify the location of your DTD file on the server. Example: If your DTD is located in the **myfolder/DTD** folder on the **ftp.mycompany.com** server, you must populate the **Server** and **Path** fields with **ftp.mycompany.com** and **/myfolder/DTD/mydtd.dtd**.
Attention: You cannot write the DTD in learning mode on an HTTP site.
-

 **Note:**

Do not include a slash mark (/) at the end of the server name; instead include it at the start of the path of the DTD or XSD file.

- **HTTP**
Enter the HTTP parameters in the **Server**, **Login**, **Password**, and **Path** fields.

The **Path** field enables you to specify the location of your DTD or XSD file on the server. Example: If your DTD is located in the **myfolder/DTT** folder on the **http://mycompany.com** server, you must populate the **Server** and **Path** fields with **http://mycompany.com** and **/myfolder/DTT/mydtd.dtd**.

 **Note:**

Do not include a slash mark (/) at the end of the server name; instead include it at the start of the path of the DTD or XSD file.

Publish a document type for each first-level collection

By choosing this option, the MQSeries connector publishes as many document types as there are first-level collections declared in the selected DTD or XSD.

Example :

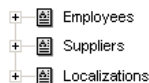
A DTD, whose root element is **Company**, contains three first-level collections: **Employees**, **Suppliers**, and **Localizations**. If you do not select this option, the connector will publish only one document type having **Company** as its root element.

 **Note:**

The first-level collections are those that appear directly under the root element of a document type published by a connector.



Using the **Publish a document type for each first-level collection** option, the MQSeries connector publishes one document type per first-level collection. In this case, the root element of the DTD (the **Company** element) no longer appears in the tab of the document types published by the XML connector.



Determine the root elements

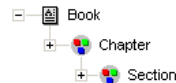
A DTD or XSD is composed of elements that can contain other elements. (For example: In a DTD intended for book publishing, **Section** elements are included in the **Chapter** elements, which are in turn included in the **Book** element.) The only element that cannot be included in another element is a root element. A DTD or XSD can contain any number of root elements or none at all.

To determine the root elements, which correspond to as many document types as published by the MQSeries connector, Connect-It gives you different options:

Publish a document type for each root element found in the DTD/XSD (recommended)

If you choose this option, the MQSeries connector searches all the root elements of the DTD/XSD and publishes a document type for each root element found.

Example: In a DTD intended for book publishing, only one root element (**Book**) is found. In this case, the MQSeries connector will only publish one document type.



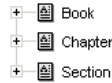
Note:

When no root elements can be found (all the elements that can include other elements), Connect-It will take the first element found in the DTD or XSD as its only root element. Example: A DTD corresponds to the organization of a relational database in which all the elements (each one representing a table) include one another: The Assets table is linked to the Users table, which is linked to the Assets table, etc.

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

If you select this option, you can indicate the root elements of your choice, separating each one with a comma.

Example: In a DTD intended for book publishing, the user decides to choose the elements **Book**, **Chapter**, and **Section**. In this case, the MQSeries connector will publish one document type per root element chosen by the user.



Note:

For a DTD representing a relational database in which all the elements (each representing a table) include one another, this option enables you to obtain a published document-type for each one of these tables.

Learning mode

In learning mode, the MQSeries connector creates a DTD file depending on the messages read in the queue specified in the Choose a queue page.

Attention: This function is not available for the XSDs.

Note:

The DTD obtained in learning mode must be saved in a local or network file. In this case, it is not possible to save it to an HTTP or FTP site.

To start learning mode:

- Configure the MQSeries connector by selecting the **Learning mode** option.
- Select your MQSeries connector in your scenario's diagram.
- Select the **Tools/ Produce** menu.
- or
- Right-click and choose **Produce** from the menu that appears.
- or
- Press **F5**.

To disable learning mode, reconfigure the MQSeries connector by clearing this option on the **Choose a DTD** page.

Conflict between XML and DTD/XSD documents

In principle, the XML elements read by the MQSeries connector must match those defined in your DTD. Nevertheless, it is always possible that new elements

appear. This type of conflict is frequently encountered when the DTD or XSD used is obtained in learning mode using a limited number of XML files containing a small number of elements.

To handle these conflicts, Connect-It provides two different options:

- **Reject document and continue**
In case of conflicts, the read messages containing non-defined elements in the DTD are rejected. (Whether these documents are saved or not depends on the option selected on the **Action after processing** page.) In order that these files be properly processed, you must restart the MQSeries connector in learning mode.
- **Issue warning and continue**
In the case of a conflict, the messages are processed normally by the MQSeries connector. A warning in the document log, recognizable by the ⚠ icon, appears in the tracking lines of the MQSeries connector in read mode.

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

If you select this option, the connector will not produce documents, processing reports or errors in the case where the root element of a produced document does not correspond to the document type on which it was based.

This option is useful when the scenario uses several XML connectors. In effect, the XML file can correspond to another document type defined in another XML connector in the scenario.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the JVM

This page is only visible if you have selected an XSD in the [Choose a DTD/XSD](#) [page 94] page of the configuration wizard.

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].








Configure the MQSeries connector (write)

The configuration of the MQSeries connector in write mode enables you to specify the MQSeries queue in which messages must be written from the Connect-It documents that it consumes.

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 4.4. Configure the MQSeries connector (write)

 Name and describe the connector [page 17]
 Choosing a processing mode [page 160]
 Configure the queue manager connection [page 160]
 Choosing a queue [page 162]
 Choose a DTD/XSD [page 163]
 Configure the reconnection parameters (advanced mode) [page 31]
 Parallelize consumption (advanced mode) [page 44]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Choosing a processing mode

The second page of the **Basic configuration** wizard enables you to choose in which mode you want to use the MQSeries connector. Choose the **Write** mode.

Configure the queue manager connection

You must specify four parameters for the MQSeries connector to connect to a queue manager.

- **Server name**
Enter the DNS name or IP address of your MQSeries server on your network.
 - **Connection port**
Enter the listening port of your queue manager. By default, the value of this port is **1414**.
 - **Channel name**
Enter the name of the channel that the queue manager uses as a communication path. The default value of the channel is **SYSTEM.DEF.SVRCONN**.
 - **Name of the queue manager**
Enter the name of your queue manager. To use the scenarios provided with the MQSeries connector, the name of this manager is **connectit**. If this field is not populated, the MQSeries connector will connect to the default queue manager of MQSeries.
 - **Alternate user ID**
Specify an alternate user ID, if required, for opening the queue manager. You can specify different alternate user IDs for each queue (main, success, failure).
-

 **Note:**

If you clear the **Use extended connection** default option, only the **Queue manager name** field can be populated.

Test button

This button enables you to test your connection.

To test your connection:

- 1 Enter your connection parameters.
- 2 Click **Test**.
The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.
- 3 Click **Close** to return to the configuration wizard.

 Note:

When the connector is opened, the operating system of the server is automatically detected.

Choosing a queue

This page enables you to enter the name of the queue from which the MQSeries connector will write the data. To use the scenarios provided with the MQSeries connector, the name of this queue is **PEREGRINE.OUT**.

Advanced configuration (advanced mode) option

This option enables you to define, for a queue:

- The name template for creating dynamic queues
Enter the format template for creating the dynamic queue name that will be used: The character '*' is replaced by the queue manager in order to guarantee the uniqueness of the dynamic queue's name.
- You can choose from three options for closing dynamic queues:
 - **Delete permanent dynamic queue if it is empty (MQCO_DELETE)**
Temporary dynamic queues are also physically deleted.
 - **Retain dynamic queue if it is permanent and delete it if it is temporary**
 - **Purge messages from permanent dynamic queue and delete it (MQCO_DELETE_PURGE)**
Temporary dynamic queues are also physically deleted.
- Authentication
For each queue, you can specify an alternate user ID.

Specify the identity context option


Select this option to be able to specify the identity context of the messages you will write in the queue.

Selecting this option modifies the structure of the produced document type **MessageInfo** and displays the following elements: **UserID**, **AccountingToken**, **APplIdentityData**.

This option is not selected by default.

Choose a DTD/XSD

To process an XML file, the MQSeries connector must use a DTD (Document Type Definition) or an XSD (XML Schema Definition). This page enables you to specify the DTD or the XSD used in the **DTD/XSD** field.

By clicking , the **File location** page is displayed. This page enables you to enter the path of your DTD or XSD depending on your connection protocol.

File location

You have the choice of three options:

- **Local/Network**

Enter the full path of your DTD or XSD file located on the computer or network on which Connect-It is installed.

- **FTP**

Enter the FTP parameters in the **Server**, **Login**, **Password**, and **Path** fields.

The **Path** field enables you to specify the location of your DTD or XSD on the server. Example: If your DTD or XSD are located in the **myfolder/dtd** folder on the **ftp.mycompany.com** server, you must populate the **Server** and **Path** fields with **ftp.mycompany.com** and **/myfolder/DTD/mydtd.dtd**.



Note:

Do not include a slash mark (/) at the end of the server name; instead include it at the start of the path of the DTD or XSD.

- **HTTP**

Enter the HTTP parameters in the **Server**, **Login**, **Password**, and **Path** fields.

The **Path** field enables you to specify the location of your DTD or XSD file on the server. Example: If your DTD or XSD is located in the **myfolder/DTD** folder on the **http://mycompany.com** server, you must populate the **Server** and **Path** fields with **http://mycompany.com** and **/myfolder/DTD/mydtd.dtd**.



Note:

Do not include a slash mark (/) at the end of the server name; instead include it at the start of the path of the DTD or XSD file.

Do not reference the resulting DTD in the XML file

This option is selected by default.

Write resulting DTD inside the XML file

If you choose this option, the MQSeries messages written by the MQSeries connector will have the DTD in their headers.

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

If you choose this option, you must specify in the available field the location to which you want to save the DTD of the MQSeries messages.

Warning:

Starting a scenario will open several sessions: a session corresponding to each source connector started in your scenario by a scheduler (Example: The source connector produces documents every week.)

A DTD or XSD is only written in the external file the first time that you use the scenario. This means that for all the following sessions of that scenario, the DTD or XSD in the external files always corresponds to the one obtained in the first session.

In order for a new DTD or XSD to be written to the external file, you must stop the scenario and then restart it.

Determine the root elements

A DTD is composed of elements that can contain other elements. (Example: In a DTD intended for book publishing, **Section** elements are included in the **Chapter** elements, which in turn are included in the **Book** element.) The only element that cannot be included in another element is a root element. A DTD can contain several root elements.

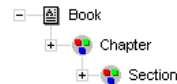
To determine the root elements, which correspond to as many document types as published by the MQSeries connector, Connect-It gives you different options:

- Publish a document type for each root element found in the DTD/XSD (recommended)
- Publish a document type for each root element selected by the user (values are separated by commas)

Publish a document type for each root element found in the DTD/XSD (recommended)

If you choose this option, the MQSeries connector searches all the root elements of the DTD or XSD and publishes a document type for each root element found.

Example: In a DTD or XSD reserved for book publishing, only one root element (**Book**) is found. In this case, the MQSeries connector publishes only one document type.



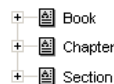
Note:

When no root elements can be found (all the elements that can include other elements), Connect-It will take the first element found in the DTD or XSD as its only root element. Example: A DTD or XSD corresponds to the organization of a relational database in which all the elements (each one representing a table) include one another: The Assets table is linked to the Users table, which is linked to the Assets table, etc.

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

If you select this option, you can indicate the root elements of your choice, separating each one with a comma.

Example: In a DTD or XSD intended for book publishing, the user decides to choose the elements **Book**, **Chapter**, and **Section**. In this case, the XML connector will publish one document type per root element chosen by the user.





Note:

For a DTD representing a relational database in which all the elements (each representing a table) include one another, this option enables you to obtain a published document-type for each one of these tables.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Configure the JVM

This page is only visible if you have selected an XSD in the [Choose a DTD/XSD](#) [page 94] page of the configuration wizard.

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Production directives of the MQSeries connector

The production directives of the MQSeries connector enable you to filter the messages that the connector reads in the queue, specified during the configuration.



These directives are:

- WHERE clause
- Series of recovery options

To enter these directives:

- 1 Double-click the MQSeries connector.

- 2 Select a document type from the **Produced document types** tab in the dialog box that appears.
- 3 Select the root element of the produced document type.
- 4 Enter your production directives.

WHERE clause

This clause enables you to filter the messages in the queue read by the MQSeries connector. This clause uses the following MQSeries syntax:

```
[key word]=[value]
```

The following key words are available:

- **MsgID** (Message ID)
- **GroupID** (ID of message group)
- **CorrelID** (ID of correlation)

The value of key words must be entered in a hexadecimal form.

In the MQSeries-Asset Management scenario provided with the MQSeries connector, the following CorrelIDs are used for the produced document types:

- **CONNIT.MQAM.REQUEST.ACK**
(434f4e4e49542e4d51414d2e524551554553542e41434b) for the **ExtRequestAcks** document type
- **CONNIT.MQAM.RECEIPT.ACK**
(434f4e4e49542e4d51414d2e524543454950542e41434b) for the **ExtReceiptAcks** document type
- **CONNIT.MQAM.VENDOR** (434f4e4e49542e4d51414d2e56454e444f52) for the **Vendors** document type
- **CONNIT.MQAM.COSTCENTER**
(434f4e4e49542e4d51414d2e434f535443454e544552) for the **CostCenters** document type

In the MQSeries-ServiceCenter scenario provided with the MQSeries connector, a CorrelID is used for the **External Contacts** produced document type:

CONNIT.MQSC.CONTACT (434f4e4e49542e4d51414d2e434f535443454e544552)

These CorrelIDs must be applied in the MQSeries messages placed in the **PEREGRINE.IN** queue by external applications.

Type of messages to recover

This list enables you to select the type of messages recovered by the connector.

Conversion of messages

For all produced document types, the **Enable conversion of messages** option is available. This option enables the conversion of queue messages to the following code pages:

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

This option is not activated by default.

Get messages only if all messages in the group are available option

This option enables you to specify how the MQSeries connector recovers messages from the queue.

Consumption directives of the MQSeries connector

The consumption directives of the MQSeries connector enable you to specify the manner in which the connector will write messages in the queue, specified during the configuration.



These directives are:

- The activation or not of the automatic segmentation of messages.
- The specification of particular IDs.
- The specification of the type of message to send.

To enter these directives:

- 1 Double-click the MQSeries connector.
- 2 Select a document type from the **Consumed document types** tab in the dialog box that appears.
- 3 Select the root element of the consumed document type.
- 4 Enter your consumption directives.

Or

- 1 Double-click a mapping box linked to the MQSeries connector in write mode.
- 2 Choose to edit or create a mapping in which the MQSeries connector is the destination connector.
- 3 Select the **Message options** tab.
- 4 Enter your consumption directives.

Automatic segmentation

If you authorize the segmentation of messages, the messages exceeding the maximum size limit in your queue will be cut into several physical messages. These physical messages are then regrouped into one logical message.

If you prohibit the segmentation of messages, the messages exceeding the maximum size limit in your queue will be rejected by the MQSeries connector.

Specification of the IDs

These options enable you to specify particular IDs:

- Group ID
- Correl ID

The IDs must be entered in a hexadecimal form.

In the MQSeries-Asset Management scenario provided with the MQSeries connector, the following CorrelIDs are used for the consumed document types:

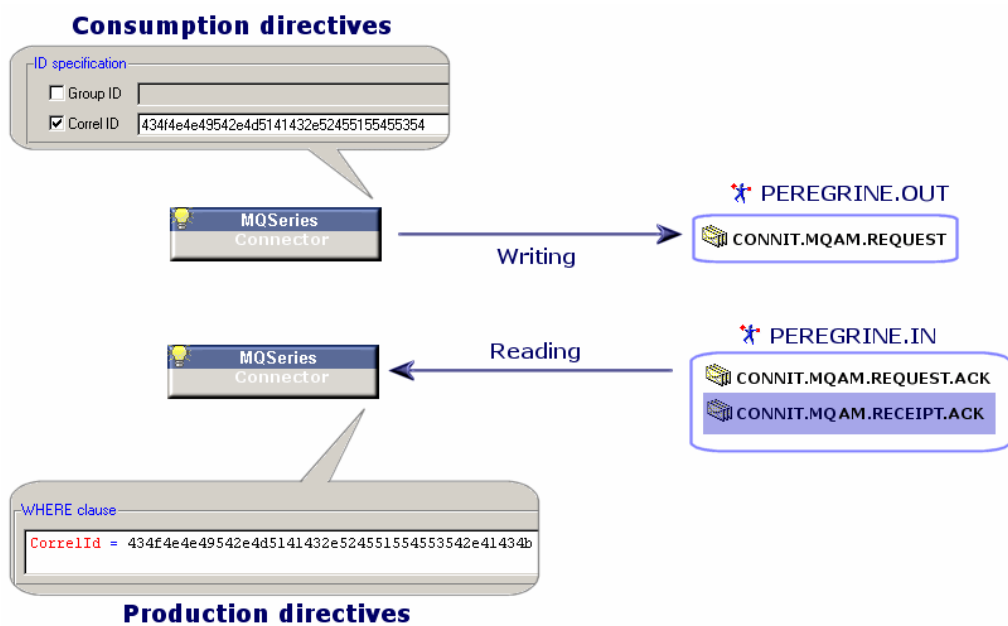
- **CONNIT.MQAM.REQUEST** (434f4e4e49542e4d51414d2e52455155455354) for the **Request** document type
- **CONNIT.MQAM.RECEIPT** (434f4e4e49542e4d51414d2e52454345495054) for the **Receipt** document type

In the MQSeries-ServiceCenter scenario provided with the MQSeries connector, a CorrelID is used for the **Contacts from ServiceCenter** consumed document type: **CONNIT.MQSC.CONTACT**

(434f4e4e49542e4d51414d2e434f535443454e544552).

The specified IDs identify your messages corresponding to purchase orders or receiving slips in the **PEREGRINE.OUT** queue. The external applications that read these messages must use the same IDs to recover these messages.

Figure 4.1. MQSeries connector - Identifying messages using Correl ID



Type of message.

This frame enables you to:

- Select the type of message sent by the connector.
- Choose the queue containing the responses to the **Request** type messages.

Additional information

This section presents additional information about the connector.

MessageInfo structure in the consumed document types

A **MessageInfo** structure is available in the consumed document types published by the connector in write mode. This structure can be used to specify certain parameters of the message to be sent.

This structure contains the following fields:

- **Expiry**

This field enables you to define the expiration delay of a message, after which, the message is deleted by the queue manager. The value of this field is expressed in tenths of a second. If a message does not expire, the value of this field is -1.

- **Persistence**

This field concerning the persistence of the message can be set to the following three values:

- 0

If the queue manager is restarted, the message is not persistent.

- 1

If the queue manager is restarted, the message is persistent.

- 2

Default persistence of queue. This is the default value of the field.

- **Priority**

This field enables you to specify the priority of the message. The value of this field must be an integer equal or greater than 0. If this field does not have a value, the priority of the queue is used by default.

- **CorrelationID**

This field specifies the value of the document's **CorrelationID**. If you map this field, this value overrides the value of the **CorrelationID** specified in the connector's directives.

- **GroupID**

This field specifies the **GroupID** of the message to send. If you map this field, this value overrides the value of the **GroupID** specified in the connector's directives.

- **AccountingToken**

This field is an element of the identity context of the message. It only appears if the **Specify the identity context** option is selected in the connector configuration wizard.

- **AppIdentityData**

This field is an element of the identity context of the message. It only appears if the **Specify the identity context** option is selected in the connector configuration wizard.

- **UserID**

This field is an element of the identity context of the message. It only appears if the **Specify the identity context** option is selected in the connector configuration wizard.

MessageInfo structure in the document types produced in read mode

In the produced document types published by the connector, a **MessageInfo** structure is available. This structure contains some of the parameters of the message read.

For more information about processing reports, refer to the **Processing reports** chapter in the Connect-It User's guide.

This document type contains the following fields:

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

Using the MessageInfo document type

In write mode (sending messages), the connector publishes a document type, **MessageInfo**, produced each time a document corresponding to a message to be sent is consumed. This document type has the following fields, which correspond to the parameters of the message sent:

- **AccountingToken**
- **ApplIdentityData**
- **MsgType**
- **PutDate**
- **ReplyToQ**
- **Report**
- **UserID**

- **MsgID**
- **CorrelationID**
- **GroupID**

Out-of-the-box scenarios - MQSeries connector

The following out-of-the-box scenarios use this connector:

- [mqseries\acmq\mqtoprgn.scn](#) [page 390]
- [mqseries\acmq\prgntomq.scn](#) [page 391]
- [mqseries\sc51mq\mqsc.scn](#) [page 391]
- [mqseries\sc60mq\mqsc.scn](#) [page 392]
- [mqseries\sc61mq\mqsc.scn](#) [page 392]

NT Security connector

Base connector

◆ Production (source)

This section describes the NT Security connector. This connector enables you to recover information concerning the NT domains of the computer on which Connect-It is installed. This information concerns the computers, users, and workgroups of the NT domains.

This connector produces but cannot consume documents.

Compatibility of the NT Security connector

The NT Security connector is compatible with the 32-bit Windows network operating system.

 **Note:**

If you decide to use the new MS Active Directory security model, we recommend that you use it with the LDAP connector, since it is not supported by the NT Security connector.

Configure the NT Security connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 4.5. Configure the NT Security connector

 [Name and describe the connector](#) [page 17]

 [Enter an NT domain](#) [page 174]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Enter an NT domain

Populate the **Domain** field with the name of the domain from which the NT Security information is recovered. By default, the domain name corresponds to the domain of the NT session in which Connect-It is started.

Note:

If you want to recover information coming from several domains, you need to enter the names of the domains to be processed and separate them with semi-colons (;).

If you want to recover all the NT domains accessible via your computer, enter an asterisk (*). Be warned, though, that this operation can overload your computer's memory and stop the execution of Connect-It.

Warning:

The short description that appears under the connector's name in the Scenario diagram is **always** the current name of the domain and the user of Connect-It. If you enter another domain in this page, it will not appear in the connector's short description.

Recovering the list of computers in a domain

Using the option **Use the domain controllers to recover a domain's list of computers** enables you to recover the full list of computers available on a given NT domain.

Document-types published by the NT Security connector

The NT Security connector publishes three document types:

- Machine
- NtDomain
- User

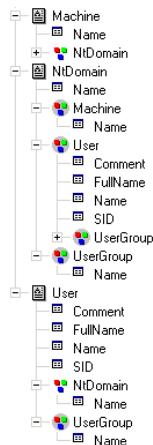
 **Note:**

The NtDomain document type corresponds to the NT domains specified during the configuration of the connector.

The different sub-nodes (structures and collections) of these document types published by the NT Security connector are:

- Computers of the domain (**Machine** collection)
- Users of the domain (**User** collection)
- User groups of the domain (**UserGroup** collections)

Figure 4.2. NT Security connector - Published document types



NT Security connector production directives

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

The production directives of the NT Security connector enable you to filter the members of the collections of the document types.

Filtering the elements of an NT domain by name

Populating the **Filter items by name** field enables you to filter members and their collections in produced document types. While creating this filter, you must use the special characters * and ?

Warning:

To create a file, you must select a collection in the document type produced by the connector.

Table 4.6. NT Security connector - Examples of production directives

Collection	Filter	Enables to recover
Machine	Platform 1	The Platform1 computer
Machine	a*	All the computers from the domain whose name starts with a .
User	Doe?	All the users from the domain whose name starts by Doe and ends with an unknown letter.

Scheduling the NT Security connector

AssetCenter Server enables you to schedule the recovery of users declared in an NT domain.

For further information on scheduling scenarios, refer to the **AssetCenter Administration Guide**, chapter **AssetCenter Server**, section **Configuring the modules monitored by Connect-It Server**.

Out-of-the-box scenarios - NT Security connector

The following out-of-the-box scenarios use this connector:

- ◆ [NT Security scenarios](#) [page 392]

Mercury Connectors

These connectors enable data related to Mercury MAM or BAC suites to be read.

These connectors are described in the **ServiceCenter-MAM/BAC Integration Solution** guide which is supplied with Connect-It.

Out-of-the-box scenarios - Mercury connectors

Below are the out-of-the-box scenarios that use these connectors:

- ◆ [Mercury Scenarios](#) [page 388]

Tivoli Enterprise Console connector (sending)

Optional connector

◆ Consumption (destination)

The Tivoli Enterprise Console connector (TEC) in sending mode enables you to send events to Tivoli Enterprise Console (TEC).

This connector is a TEC adapter using the TEC EVD APIs to send events.



Prerequisites

The Tivoli Enterprise Console connector uses Java Libraries from the EFI (Event Integration Facility) API which are supplied with the Tivoli Enterprise Console installation.

These libraries are:

- **evd.jar**
- **log.jar**

They are located in [TEC installation folder]/bin/generic_unix/TME/TEC.

To use the connector, you can:






- copy the libraries in the 'lib' subfolder in the Connect-It installation folder.
- modify the Connect-It global 'classpath' to add these libraries (**Java/Configure JVM** menu).
- add these libraries individually to the 'classpath' (Connector Configuration Wizard, **Configure JVM** page).

Configure the Tivoli Enterprise Console adapter (sending)

Using a wizard, you can configure this connector.

The following table presents all the pages in the connector's configuration wizard.

Table 4.7. Configure the Tivoli Enterprise Console adapter (sending)

 Name and describe the connector [page 17]
 Configure TEC server configuration [page 178]
 Specifying the event classes [page 179]
 Advanced configuration (advanced mode) [page 181]
 Configure the JVM [page 181]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Configure TEC server configuration

This page enables you to enter the configuration parameters to the TEC server.

Server

Specify the name or address of the TEC server.

This field is mandatory.

Port

Specify the server port: The default port is 5529.

This field is mandatory.

The server does not support UTF-8.

By default, all the events sent to the TEC server are coded in UTF-8.

Select this option if your server does not support UTF-8 (versions earlier than the version 3.7 of TEC).

If this option is selected, the code page used by the server must be indicated in the **Server code page** field.

Server code page

If the previous option was selected, you must indicate in this field the code page used by the server.

Specifying the event classes

BAROC files

You need to specify the BAROC files that contain the TEC event classes. The event classes of these files must correspond to those in the TEC server, which are defined in the **BAROC** files.

The Tivoli Enterprise Console BAROC files contain a hierarchy of event classes. The order of declaration of the BAROC files is therefore meaningful in order to reflect this hierarchy. Thus, if the file two.baroc defines an event A that inherits event B defined in the file one.baroc, then this file must be the first in the list of files.

Note:

The scenarios based on a TEC connector prior to version 3.4 must be reconfigured.

This field is mandatory.


The fields contained in the event classes can be extended by an extension file.

Extension file



The BAROC files defining the event classes handle the following data types: INTEGER, INT32, STRING and REAL.

The **Extension file** field enables you to specify a BAROC extension file. This is an XML file that extends the definitions of the event classes contained in the BAROC files. This file:

- Specifies a description for each event class and for each event-class field. This description is shown in the Connect-It document type editor.
- Extends the event class field types. This makes it possible to declare fields as 'Date and time' type for easier handling in Connect-It.

A wizard enables you to create and modify this extension file. You can access it by clicking the magnifier .


To create / modify the extension file:

- 1 Specify the path of the extension file
- 2 Click 
- 3 Define the regional settings used to convert event class fields, in particular for 'Date and time' type fields (**Locale** field). By default, the regional settings of Connect-It are used.
- 4 Populate the list of event classes that will be extended ()
- 5 Click **Next** to extend the definitions of the event class fields.

Note:

The event classes and their fields must be already defined in the BAROC fields otherwise an error will be recorded in the document log when the connector is opened.

However, you are not obligated to extend all event classes or all fields of an event class. The event classes and fields that are not extended remain as defined in the BAROC files.

-
- 6 Click  to extend a field and populate its type. For 'Date', 'Time' and 'Date and time' type fields, you can specify the format used. The default format used for this field type is the number of seconds elapsed since January 01, 1970.
 - 7 Click **Finish** once all the fields have been defined.

Advanced configuration (advanced mode)

Activate the event buffer

By selecting this option, the events sent to the server are stored in a buffer file in case the connection to the TEC server is interrupted.

 **Note:**

If you do not select this option, any interruption to the connection with the TEC server will result in the rejection of the documents by the server.

Buffer file

Indicate the full path of your buffer file.

This field is mandatory if the **Activate event buffer** option is selected.

Maximum size (KB) of the buffer file

Here you specify the maximum size of the buffer file. If the maximum size is reached, and the connection to the server is interrupted, the TEC connector will reject any new document that it must consume.

Configure the JVM

This page is only visible if you selected an XSD in the [Choose a DTD/XSD](#) [page 94] page of the configuration wizard.

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Out-of-the-box scenarios

There are no out-of-the-box scenarios provided.

Tivoli Enterprise Console connector (receiving)

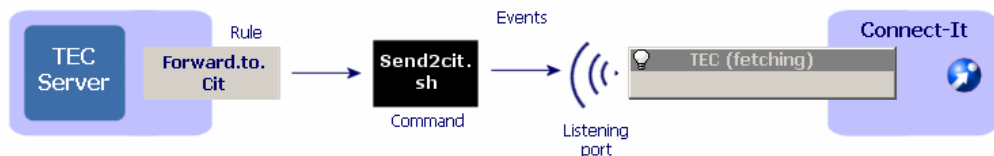
Optional connector

◆ Production (source)

The Tivoli Enterprise Console connector (TEC) in receiving mode is an event-driven connector that enables you to receive events from Tivoli Enterprise Console.

A listening port specified in the connector's configuration enables it to listen for events sent by TEC.

Rules defined in TEC enable you to send events to the TEC connector. Refer to the sub-section [Creating event-sending rules in TEC](#) [page 184] of this section for more information.



Prerequisites

The Tivoli Enterprise Console connector uses Java Libraries from the EFI (Event Integration Facility) API which are supplied with the Tivoli Enterprise Console installation.

These libraries are:

- **evd.jar**
- **log.jar**

They are located in [TEC installation folder]/bin/generic_unix/TME/TEC.

To use the connector, you can:

- copy the libraries in the 'lib' subfolder in the Connect-It installation folder.
- modify the Connect-It global 'classpath' to add these libraries (**Java/Configure JVM** menu).
- add these libraries individually to the 'classpath' (Connector Configuration Wizard, **Configure JVM** page).

Configuring TEC to send events





In order for the TEC connector to receive events, you must create rules in TEC. For information about how to create rules in TEC, refer to the TEC Rules Builder guide.

Configure the TEC connector (receiving)

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 4.8. Configure the TEC connector

 Name and describe the connector [page 17]
 Configure events listening [page 183]
 Specifying the event classes [page 184]
 Configure the JVM [page 184]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Configure events listening

This page enables you to configure the deployment parameters for the TEC connector (receiving).

Listening port

Specify the listening port used by the connector to receive events sent by TEC. This listening port corresponds to the one indicated in the `send2cit.sh` script parameters. Consult the sub-section [Configuring the send2cit.sh file](#) [page 186] of this section for more information.

This field is mandatory.

Specifying the event classes

BAROC files

You need to specify the BAROC files that contain the TEC event classes. The event classes of this file must correspond to those in the TEC server, which are defined in the **BAROC** files.

This field is mandatory.

The Tivoli Enterprise Console BAROC files contain a hierarchy of event classes. The order of declaration of the BAROC files is therefore meaningful in order to reflect this hierarchy. Thus, if the file two.baroc defines an event A that inherits event B defined in the file one.baroc, then this file must be the first in the list of files.



Note:

The scenarios based on a TEC connector prior to version 3.4 must be reconfigured.

The fields contained in the event classes can be extended by an extension file. For information on customizing the fields contained in the event classes, refer to section [Extension file](#) [page 180] of this chapter.

Extension file

Refer to sub-section [Extension file](#) [page 180]

Configure the JVM

This page is only visible if you selected an XSD in the [Choose a DTD/XSD](#) [page 94] page of the configuration wizard.

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Additional information

Creating event-sending rules in TEC

In order for TEC to send the events of an event class to Connect-It, you need to create a rule in TEC that launches the **send2c:it** command.

This command enables you to send back the events created in TEC to the TEC connector (receiving). For example, it enables TEC administrators to receive error warnings and thus implement specific problem-resolution procedures.

 **Note:**

The `send2cit` command is a Unix shell script (`.sh`). However, this script can also be used with a Windows installation of TEC; a Unix shell is automatically provided with TEC.

To send events to the connector, you must configure `send2cit` to give it the location of the connector, i.e. the name or address of the server hosting Connect-It and the listening port of the connector. Refer to section [Configuring the `send2cit.sh` file](#) [page 186].

Location of the `send2cit` file

The `send2cit` command is available in the following folder: **[Connect-It installation folder]/datakit/tecevent_package.exe**. The compressed auto-executable folder `tecevent_package.exe` contains the following files:

- `getValue.exe`
- `tecevent.jar`
- `tecevent_lang.jar`
- `send2cit.sh`
- `tecevent.config`
- The **Windows** folder contains the `getvalues.exe` file.
- The **Solaris**, **AIX** and **HP-UX** folders each contain a `getvalues` file.

All these files (and the files `getvalues.exe` or `getvalues` corresponding to the operating system that is used) must be copied on the TEC server inside a unique folder.

Example - Sending events from the `TEST_CLASS` event class to Connect-It

The following example is a simple example of creating a rule in TEC to send an event to the TEC (receiving) connector when it is received by TEC.

To create a rule enabling you to send events from the `TEST_CLASS` event class:

- 1 Launch Tivoli Desktop.
- 2 Create a new base rule.
- 3 In this base rule, create a new set of rules called, for example, **Forward**.
- 4 In this set of rules, create a new rule simply called, for example, **fwd_test_class**.

- 5 Specify that this simple rule applies to the **TEST_CLASS** event class.
- 6 Edit the rule actions.

Edit the rule actions

To edit the rule's actions:

- 1 Create an action when you receive an event.
- 2 Select **Launch a command**.
- 3 Select the **send2cit.sh** command.
- 4 Edit the command's arguments:
 - -p XXX
XXX corresponds to the listening port of the TEC connector (receiving).
This argument is **mandatory**.
 - -s citserver
citserver is the name of the server on which Connect-It is installed.
This argument is optional. If this argument is not defined, the server is the computer on which TEC is installed.
 - -l logfile
Logfile is the full path of the log file.
This argument is optional. If this argument is not defined, a log file will not be created.
- 7 Save the base rule (Forward).
- 8 Build the base rule.
- 9 Load the base rule.

After you load the base rule, each **TEST_CLASS** event received by TEC is resent to Connect-It and processed by the TEC connector (fetching).

For more information about creating rules in TEC, refer to the TEC documentation.

Configuring the send2cit.sh file

The **send2cit.sh** file may be configured:

- from the command line
- using a configuration file

If you use a configuration file to configure the **send2cit.sh** file, this overrides the command line options.

Configuring from the command line

Prior to Connect-It version 3.4.0, the send2cit command could be configured from the command line. This functionality is still supported, however we now recommend using a configuration file.

The command parameters are case-sensitive.

 **Note:**

The **-h** command-line parameter displays help on the send2cit command.

Location of the TEC (receiving) connector used from the command line

The location of the TEC (receiving) connector is given by two parameters:

- The name or the address of the server hosting Connect-It. This is specified using the **-s** parameter. The default value is 'localhost'.
- The listening port of the connector. It is specified using the **-p** parameter.

For example:

```
send2cit -s connectitsvr -p 12345
```

Using a log file from the command line

The send2cit command enables you to write to a log file.

You can specify the log file using the **-l** parameter. You can also specify a maximum size, in kilo-bytes, for a log file by using the **-m** parameter.

For example:

```
send2cit -s connectitsvr -p 12345 -l /tec/connectit/send2cit.log -m 2048
```

Configuration using a configuration file

The send2cit command can read its settings from a configuration file. The configuration file makes it possible to use advanced functionality of send2cit.

The send2cit configuration file uses the same syntax as the property files:

```
# Comment
Property = Value
OtherProperty = Other value
```

A sample configuration file is provided with Connect-It.



The **-listCfgProps** command-line parameter displays the list of properties of the configuration file and their descriptions.

Defining the configuration file to use

By default, send2cit uses the **tecevent.config** file if found next to the **send2cit.sh** file. If this file does not exist, or if you wish to use different one, you can use the **-cfg** command-line parameter.

For example:

```
send2cit -cfg /tec/connectit/myConfig.conf
```

Location of the TEC (receiving) connector

The location of the TEC (receiving) connector is given using by two properties in the configuration file:

- **Server:** Name or address of the server hosting Connect-It. The default value is 'localhost'.
 - **Port:** Listening port of the TEC (receiving) connector.
-



The **-s** and **-p** command line parameters, if they are used, override the values given the configuration file.

Using a log file

Two properties in the configuration file enable you to specify a log file:

- **LogFile:** Full path of the log file to use.
 - **MaxLogSize:** Maximum size in kilobytes, of the log file. The default size is '2000'.
-



The **-l** and **-m** command line parameters, if they are used, override the values given the configuration file.

Defining generic event classes

In a Tivoli Enterprise Console configuration, an event class is often used as a parent class for set of event classes.

For example, an event class, NetworkProblem, will be used as a parent class for event classes NP1, NP2, NP3, NP4... The NPx event classes don't clog up the fields of the NetworkProblem parent class, however their names identify the network problem.

To process these events in Connect-It, one might want to use the same mapping for all event classes deriving from NetworkProblem, to open a ticket in ServiceCenter, for example.

To achieve this, we can define a generic event class, NetworkProblem, in the send2cit configuration file that groups the NPx classes together. When send2cit is invoked to send an event from the NPx class to Connect-It, it will thus send a NetworkProblem event.

When generic event classes are defined, the real event class remains available in Connect-It in the 'EVENT_CLASS' element of the document type produced by the TEC (receiving) connector.

Generic event classes are defined in the configuration file using the prefix 'G_':

```
G_MyGenericClass = SpecificClass1 SpecificClass2 SpecificClass3
```

In our example, to define the generic event class, NetworkProblem, the configuration file must contain the line:

```
G_NetworkProblem = NP1 NP2 NP3 NP4
```

Handling connection problems

It is possible to specify a connection timeout for the TEC (receiving) connector. The connection is considered as having failed beyond this time. This delay is specified using the **TimeOut** property and its value is expressed in milliseconds. By default, when the connection to the TEC (receiving) connector fails and the event log is not activated, an error is raised and an error code is returned by send2cit.

It is also possible to tell send2cit to send an event to TEC when the connection fails. This event thus enables the TEC administrator to rectify the problem. The class of the event sent is **CIT_TecEvent_ConnectionFailure**. To activate this functionality, the following properties are used:

- **SendEventOnCnxFailure**: The value of this property must be set to 'true' to enable sending the event.

- **CnxEventFailureSeverity:** Severity of the event sent. The default value is 'WARNING'.

Sending events to TEC requires additional configuration of send2cit and TEC. For further information, refer to section [Sending events to TEC](#) [page 191].

Using an event cache

The cache folder stores events that could not be sent if an error connection occurs between the TEC server and the TEC (receiving) connector. When the connection is restored, the contents of the cache is sent back to the server. In this way, no events are lost.

Two properties are used to specify the use of a cache:

- **UseCache:** The cache is enabled if the value of this property is set to 'true'.
- **CacheDir:** This property specifies the folder to be used to store the event cache

If you enable the cache, you must specify a folder to use.

Overview of the cache

When send2cit fails to send an event to the TEC (receiving) connector, this event is stored in the cache in the form of a file in the cache folder. The name of the folder is formed as **<server>_<port>_XXXX.cache**, where XXXX is a sequence guaranteeing the uniqueness of the file name.

When the next call to send2cit is made, if the connection to the connection is reestablished, send2cit starts by purging the cache before sending the new event, i.e. it starts by sending back the events stored in the cache.

Limiting the size of the cache

Limiting the size of the cache means limiting the number of events saved in the cache folder.

By default, the size of the cache is limited to 100 events. To change this value, use the **MaxCacheSize** property.

The **MaxCacheSizeAction** property determines the behavior of send2cit when an event cannot be sent and the cache has already reached its maximum size.

The possible values are:

- **Drop:** The event is rejected, an error message is generated and an error code is returned by send2cit.

- **Error:** The event is placed in the cache, but an error message is generated and an error code is returned by send2cit.
- **Warning:** The event is placed in the cache, and a warning message is generated.
- **None:** The event is placed in the cache, without any other action. This value is the equivalent of disabling the cache.
- **SendEvent:** The event is placed in the cache but an event is sent to the TEC server.

The default behavior is 'None', i.e. the size of the cache is not limited by default. When the **MaxCacheSizeAction** property is set to **SendEvent**, an event of the **CIT_TecEvent_CacheFull** class is sent to Connect-It.

The **CacheFullEventSeverity** property enables you to specify the severity level of the event sent. The default severity is 'MINOR'.

Sending events to TEC requires additional configuration of both send2cit and TEC. For further information, refer to section [Sending events to TEC](#) [page 191].

Sending events to TEC

In order for send2cit to be able to send events to TEC if a connection problem occurs or if the cache reaches its maximum size, send2cit and TEC must be configured.

Configuring send2cit to send events to TEC

The following properties are used to send events to TEC:

- **TECServer:** Name or address of the TEC server to send the event to. The default value is 'localhost'.
- **TECPort:** Connection port of the TEC server to send the event to. The default value is '5529'.
- **SendEventCommand:** Command used to send the event. The default value 'postmsg'; In this case, the postmsg command must be in the PATH. It is possible to specify the full path of the command.
- **EventSource:** Source of the event sent

Configuring send2cit to receive events from TEC

The event classes sent by send2cit are defined in the BAROC file **cit_event.baroc** provided with send2cit. In order for TEC to receive events sent by send2cit, you must therefore import this file into TEC.

The send2cit event classes derive from the same parent class **CIT_TecEvent_Error**, which defines the four following fields:

- `tecevent_error_code`
- `src_event_handle`
- `src_date_reception`
- `src_server_handle`

The **tecevent_error_code** field contains the send2cit error code that provoked the sending of the event.

The three other fields makes it possible to correlate the event with the event at the origin of the send2cit error.

The source of the events sent by send2cit, and defined by the **EventSource** property in the send2cit configuration file, must also be declared in TEC.

send2cit error codes

- 1 Error while recovering the environment variables containing the event data to send.
- 2 Bad command-line parameters.
- 3 Connection problem while sending the event to the TEC connector (receiving).
- 4 Error while reading the configuration file.
- 5 Error while processing the command-line parameters.
- 6 Error while processing the environment variable: A variable is missing.
- 7 Unknown error.

Out-of-the-box scenarios

There are no out-of-the-box scenarios provided.

5 Protocol connectors

CHAPTER

Protocol-type connectors are those which enable you to process data when the processing uses a particular communication protocol.

Database connector

Optional connector

- Production (source)
- Consumption (destination)

The Database connector enables you to process data from an ODBC data source or directly from an Oracle, Sybase, DB2 or MySQL database.

 **Warning:**

To use the ODBC connections, the ODBC Administrator must have been previously installed.

Compatibility of the Database connector

The Database connector has been successfully tested with the following drivers and the ODBC administrator version 3.520.4403.2:

ODBC	Oracle	Sybase	DB2
Access version 4.00.4403.02	Oracle 7 version 2.05.0301	Sybase 11.9	DB2 7.x
Excel 4.00.4403.02	Oracle 8.0 version 8.00.0500	Sybase 12.x	DB2 8.x
Sybase 11.9 3.11.0001	Oracle 8i version 8.01.0500		
Sybase SQL Anywhere version 5.05.052739			
Text version 4.00.4403.02			
Microsoft SQL Server version 3.70.08.20			
IBM DB2 version 7.x (from DB2 versions 6 and higher)			
INFORMIX 3.33 32 BIT version 3.33.00.10688			

The Database connector should be compatible with any ODBC driver version **3.xx**.

 Note:

The Database connector cannot connect to an AssetCenter database.

Known limitations of the Database connector

We highly recommend using the Database connector with an ODBC driver and ODBC administrator using identical version levels of DLL files.

Example :

The Database connector will not work if you use ODBC Administrator version 3.0 and an Access 4.00 ODBC driver.

Type of supported fields

The document types published by the Database connector are made of fields. Each of these fields has a particular type: text field, lob field, Blob field, etc. Depending on the database driver used, the field types supported by the Database connector are different.

DB2 MVS - Microsoft ODBC driver for DB2

The following configurations have been used:

- DB2 Connect DB2 SDK 8.1.10
- ODBC IBM DB2 driver 8.01.00.36
- Z/OS V1.2 DB2 V7.2 server

Field type	Read-only support	Write support
Blob	No	No
character(n)	Yes	Yes
date	Yes	Yes
decimal(p,s)	Yes	Yes
float(p)	Yes	Yes
integer	Yes	Yes
long varchar	Yes	Yes
timestamp	Yes	Yes
varchar	Yes	Yes

DB2 - Native connection

Field type	Read-only support	Write support
bigint (64-bit integer)	Yes (for 32 bit integers only)	Yes (for 32 bit integers only)
blob	No	No
character	No	No
clob	Yes	Yes
date	Yes	Yes
datalink	No	No
decimal	Yes	Yes
dbclob	No	No
double	Yes	Yes
char	Yes	Yes
graphic	No	No
integer	Yes	Yes
real	Yes	Yes

Field type	Read-only support	Write support
time	Yes	Yes
smallint	Yes	Yes
timestamp	Yes	Yes
varchar	Yes	Yes
varchographic	No	No

Oracle 8.0 - Microsoft ODBC driver for Oracle

Field types	Supported
bfile	No
Blob	Yes
char	Yes
clob	No
date	Yes
float	Yes
integer	Yes
long	Yes
long raw	Yes
nchar	Yes
nclob	No
nvarchar	No
number	Yes
ms_label	Yes

Oracle 8.0 - Native connection

Field type	Write support	Read-only support
bfile	No	No
Blob	No	No
char	Yes	Yes
clob	No	No
date	Yes	Yes
float	Yes	Yes
integer	Yes	Yes
long	Yes	Yes
long raw	Yes	Yes
nchar	No	No
nclob	No	No
nvarchar	No	No
number	Yes	Yes

Field type	Write support	Read-only support
ms_label	No	No

MSSQL 7.0 - SQL Server driver, version 2000.80.194.00

Field type	Supported
binary	Yes
bit	Yes
char	Yes
datetime	Yes
decimal	Yes
float	Yes
image	Yes
int	Yes
money	Yes
nchar	No
ntext	Yes
numeric	Yes
nvarchar	No
real	Yes
smalldatetime	Yes
smallint	Yes
smallmoney	Yes
text	Yes
timestamp	Yes
tinyint	Yes
uniqueidentifier	No
varbinary	Yes
varchar	Yes

MySQL - Native connection

Field type	Read-only support	Write support
bigint	Yes	Yes
bit	Yes	Yes
blob	Yes	Yes
bool	Yes	Yes
char	Yes	Yes
date	Yes	Yes
decimal	Yes	Yes
double	Yes	Yes

Field type	Read-only support	Write support
double precision	Yes	Yes
enum	No	No
float	Yes	Yes
int	Yes	Yes
longblob	Yes	Yes
longtext	Yes	Yes
mediumblob	Yes	Yes
mediumtext	Yes	Yes
numeric	Yes	Yes
set	No	No
smallint	Yes	Yes
text	Yes	Yes
time	Yes	Yes
timestamp	Yes	Yes
tinyblob	Yes	Yes
tinyint	Yes	Yes
tinytext	Yes	Yes
varchar	Yes	Yes

Sybase12 - Sybase System 11 driver version 3.11.00.01

Field type	Supported
binary	Yes
char	Yes
datetime	Yes
decimal	Yes
float	Yes
image	Yes
int	Yes
money	Yes
nchar	Yes
ntext	Yes
numeric	Yes
nvarchar	Yes
real	Yes
smalldatetime	Yes
smallint	Yes
smallmoney	Yes
text	Yes
tinyint	Yes
varbinary	Yes

Field type	Supported
varchar	Yes

Sybase12 - Native connection

Field type	Read-only support	Write support
binary	No	No
bit	No	No
char	Yes	Yes
datetime	Yes	Yes
decimal	Yes	Yes
float	Yes	Yes
image	Yes	Yes (version 2.70 and higher)
int	Yes	Yes
money	Yes	Yes
nchar	No	No
ntext	No	No
numeric	Yes	Yes
nvarchar	No	No
real	Yes	Yes
smalldatetime	Yes	Yes
smallint	Yes	Yes
smallmoney	Yes	Yes
text	Yes	Yes
tinyint	Yes	Yes
varbinary	No	No
varchar	Yes	Yes













Configure the Database connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

Certain options concerning this connector can be enabled using the **Edit/Options/ Connector** menu. For more information, refer to the **User's Guide**, chapter **Scenario builder**, section **Connector**.

The following table presents all the pages in the connector's configuration wizard.

Table 5.1. Configure the Database connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Manage transactions (advanced mode) [page 38]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]
 Parallelize consumption (advanced mode) [page 44]

Prerequisites

Before configuring your connector, make sure a valid connection to your **ODBC** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and passwords.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

 Note:

If you use the **AssetCenter Database** ODBC driver, you must append the name of the AssetCenter database connection to the login using the pipe character '|' as a separator.

For example, Admin|ACDemo420En.

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

- **SQL92 supported:** if you use the **AssetCenter Database** ODBC driver, disable this option.
- **Perform an initial import:** this option avoids using a reconciliation script in a SELECT query for an initial import, i.e. when the database does not contain any records or when the elements to be inserted do not exist yet. This allows to reduce the record processing time.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Manage transactions (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Manage transactions \(advanced mode\)](#) [page 38].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Production directives of the Database connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

The ORDERBY clause enables you to indicate the field used to define the order in which documents are produced by the database connector. You can enter a list of fields separating them with commas each time.

Note:

The WHERE and ORDERBY clauses apply at the root level of the document type produced and at the level of the collection.

For more information about AQL syntax, refer to the appendix [AQL queries](#) [page 427] and section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Sysdate field

In the WHERE clauses written in AQL, the use of the **sysdate** field must be replaced with the function **getdate()**.

Example: The SQL query: **SELECT * from AmAsset where dinstall>sysdate** must be replaced with the AQL query **SELECT * from AmAsset where dinstall>getdate()**.

NULL

When a "Numeric" type field is not populated (its value is NULL), Connect-It sets its value to "0". Similarly, the absence of a link is comes out as "Link = 0" or "foreign key = 0". Example: "Location=0" or "Locald=0".

Consumption directives of the Database connector

For information about how to enter a connector's consumption directives, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51].

To set a connector's consumption directives, you must enter reconciliation parameters in the **Reconciliation** and **Advanced reconciliation** tabs.

For more information about these reconciliation tabs, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51], sub-section [Reconciliation](#) [page 52].

Additional information

This chapter presents the following additional information:

- ◆ Additional information about declaring your ODBC data source.

Additional information about declaring your ODBC data source.

When a scenario using a Database connector is launched as a service in 32-bit Windows, the ODBC data source must be declared as a system data source (system DNS) and not as a user data source (user DNS).

Verifying that an ODBC data source is a system data source

To verify that a data source is a system data source:

- 1 Start ODBC Administrator.
- 2 Check that the ODBC data source used by your connector is located in the **System DNS** tab.

If it is not, delete it, and then create a new ODBC data source.

Out-of-the-box scenarios - Database connector

There is no scenario available for this connector.

XML listening connector

Optional connector

◆ Production (source)

Using the queries it receives, the XML listening connector can:

- Produce documents from these queries.
- Interact with another connector that processes the queries

The XML data is extracted from the contents of the HTTP requests sent to the servlet.

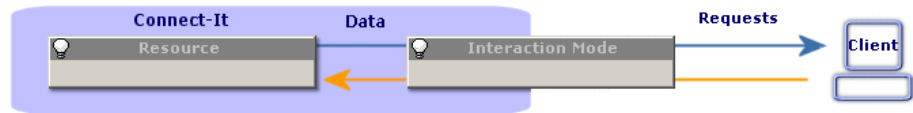
XML listening connector in publishing mode (document production)

The XML listening connector in publishing mode receives events that enable it to produce documents.



XML listening connector in interaction mode

The connector in interaction mode listens to the queries sent by clients. The connector transmits these requests to a resource connector, who in turn produces documents.



Compatibility of the XML listening connector

The connector is compatible with all Web servers supporting Java servlets.

Limitations of the XML listening connector

The limitations of the connector depend on the optional Web resource that is used.

About the Servlet

A servlet, integrated inside the Web server, has the task of:

- Handling incoming HTTP requests.
- Extracting the appropriate XML data from the incoming requests
- Creating a client socket if required
- Sending XML documents to the XML listening connector via the socket just created

Since the servlet must know the socket parameters (port, host) to be used when it communicates with the connector, using the XML listening connector creates a HTTP POST request for the servlet with the following form:

```
POST servletURL?socketHost=string?socketPort=string HTTP/1.1
Host: webServerHost
```

In fact, the servlet must support HTTP POST queries.

For example, to specify that the URL `http://www.peregrine.com/conit/connector` is to create a socket for the 'localhost' host on port '1700' when it receives data from a client computer, the HTTP POST is as follows:





```
POST conit/connector?socketHost=localhost?socketPort=1700 HTTP/1.1
Host: www.peregrine.com
```

Configure the XML listening connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 5.2. Configure the XML listening connector

 Name and describe the connector [page 17]
 Configure connection [page 206]
 Configure the processing of XML events [page 207]
 Configure the JVM [page 207]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Configure connection

Connection

This frame enables you to enter the connection parameters for the XML listening connector:

- **Port**

Specify the listening port used by the XML listening connector.

This port enables a Web server to create the communication channel (socket) between the Web server and the XML connector.

The option **Link the connection to a Web resource** is selected by default. Disabling this option enables you to open a connector without it being linked to Web server.

- **URL**

Specify the URL of the servlet processing the HTTP requests (POST or GET) that generate XML events.

- **Maximum number of connections**

Indicate the maximum number of simultaneous connectors that can be processed by the connector. The default value of this field is 20.

This field is only available if the configuration is made in advanced mode.

By default, the **Refuse connection after maximum limit reached** option is cleared. In this case, as soon as the maximum number of connections is reached, the new connections are put into a queue until they can be processed by the connector.

Configure the processing of XML events

This frame enables you to enter request-processing parameters:

- Policy of processing queries.

Indicate the connector's mode of processing queries. You can choose between **Publication** and **Interaction**.

- Configuration file for the publications.

Indicate the path of the XML file that specifies the document types published by the connector.

- Connector for interactions

Indicate the connector processing interaction requests. You must be able to use this connector in production mode.

Configure the JVM

This page is only visible if you selected an XSD in the [Choose a DTD/XSD](#) [page 94] page of the configuration wizard.

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Using the XML listening connector

Publication

When the XML listening connector is in publishing mode, the XML document received by the XML listening connector is considered as produced by the connector. It is then processed in the same way as any other document.

If the processing is successful, the produced document is sent back. If the processing fails, an error message is returned.

The published document is formed as follows:

```
<DocToPublish>
<STRUCTURE name='Employee' >
<ATTRIBUTE name='FirstName'>John</ATTRIBUTE>
<ATTRIBUTE name='LastName'>Smith</ATTRIBUTE>
</STRUCTURE>
</DocToPublish>
```

Le fichier de configuration pour les publications ou la propriété du 'SchemaFile' est fourni à Connect-It pour valider la publication. Par exemple :

```
<PublishingSchema>
<STRUCTURE name='Employee' >
<ATTRIBUTE name='FirstName' type='String' />
<ATTRIBUTE name='LastName' type='String' />
</STRUCTURE>
<STRUCTURE name='amAsset' >
<ATTRIBUTE name='AssetTag' type='String' />
<ATTRIBUTE name='dtLastModif' type='Date' />
</STRUCTURE>
<STRUCTURE name='amProduct' >
<ATTRIBUTE name='BarCode' type='String' />
</STRUCTURE>
</PublishingSchema>
```

Note:

The document sent must be fully compatible with the configuration file in order to be processed correctly.

Interaction

When the XML listening connector is in interaction mode, the XML document must bind the target resource query (i.e. perform a Result Set type operation).

Example

Recovering the 'AmAsset' type records from the default target database.

```
<operation>
<STRUCTURE name='AmAsset' >
<ATTRIBUTE name='AssetTag' type='String' />
<ATTRIBUTE name='FullName' type='String' />
</STRUCTURE>
</operation>
```


If a 'count' type attribute is specified, it set the maximum size of the resulting document.

Example

Recovering the ten first records from the default target database.

```
<operation count='10'>
<STRUCTURE name='AmAsset'>
<ATTRIBUTE name='AssetTag' type='String' />
<ATTRIBUTE name='FullName' type='String' />
</STRUCTURE>
</operation>
```

If a 'target' attribute is specified, the name of the connector specified is used instead of that defined in the wizard.

Example

```
<operation target='ODBC'>
<STRUCTURE name='AmAsset'>
<ATTRIBUTE name='AssetTag' type='String' />
<ATTRIBUTE name='FullName' type='String' />
</STRUCTURE>
</operation>
```

A directive may be specified in order to control the appearance of the documents recovered.

Example

Recovering records using a WHERE clause.

```
<operation>
<STRUCTURE name='AmAsset'>
<ATTRIBUTE name='AssetTag' type='String' />
<ATTRIBUTE name='FullName' type='String' />
</STRUCTURE>
<layer>
<STRUCTURE name='AmAsset'>
<where>AssetTag LIKE 'UTL%'</where>
</STRUCTURE>
</layer>
</operation>
```

This example returns to the client socket an XML document as a collection of elements:

```
<COLLECTION>
<STRUCTURE name='AmAsset'>
<ATTRIBUTE name='AssetTag'>UTL000338</ATTRIBUTE>
<ATTRIBUTE name='FullName'>/GENASSET-338/UTL000338</ATTRIBUTE>
</STRUCTURE>
```

```

<STRUCTURE name='AmAsset'>
<ATTRIBUTE name='AssetTag'>UTL000442</ATTRIBUTE>
<ATTRIBUTE name='FullName'>/GENASSET-442/UTL000442</ATTRIBUTE>
</STRUCTURE>
. . .
</COLLECTION>

```

 **Note:**

For all these examples, the root tag of the 'operation' queries corresponds to the operation to use on the target connector in order to perform the query. This is appropriate if the target connector is a Java connector and corresponds to the **Connector for interactions** field in the connector configuration wizard. For the non-Java connectors, this tag must have the value 'null'.

```

<null>
. . .
</null>

```

Example

Recovering records when the target connector is not a Java connector.

```

<null>
<STRUCTURE name='AmAsset'>
<ATTRIBUTE name='AssetTag' type='String' />
<ATTRIBUTE name='FullName' type='String' />
</STRUCTURE>
<layer>
<STRUCTURE name='AmAsset'>
<where>AssetTag LIKE 'UTL%'</where>
</STRUCTURE>
</layer>
</null>

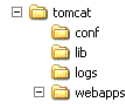
```

Example - Tomcat Web server

This section includes an deployment example and a servlet test. We assume that the client queries are HTTP GET type requests and that the appropriate XML document is included in a 'document' parameter of the query.

Deploying a servlet on the Tomcat server

The following illustration shows the structure of the Tomcat installation folder used in this example.



You must copy the Web application (**.war**) included in the datakit to the **webapps** folder. This file is used by the Tomcat server to deploy the servlet.

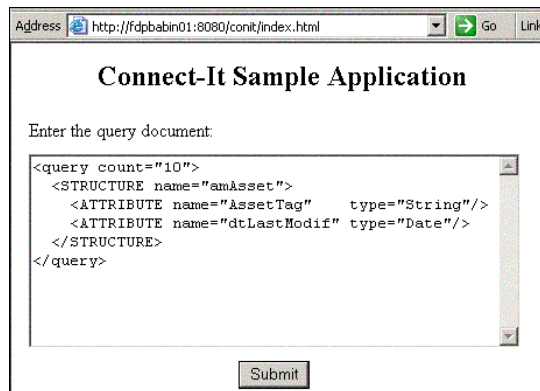
Testing the XML listening connector

- 1 Start the Tomcat Web server
- 2 Create a scenario in Connect-It using the XML listening connector. You can use the following parameters:

```

Port=1024
URL=http://localhost:8080/conit/queryDocument
Policy=Interaction
Resource=AssetManagement
  
```

- 3 Applying a scheduling to the scenario
- 4 In an internet browser, enter the following URL : `http://localhost:8080/conit`. The following HTML page is displayed:



- 5 Enter a document and submit it
- 6 Close the scenario and stop the Tomcat server

Production directives of the XML listening connector

These directives are present in queries processed in the form of XML documents by the connector.

Document types published by the XML listening connector

In publishing mode, the XML listening connector publishes the following document types, specified by the configuration file that is used:

Out-of-the-box scenarios - XML listening connector

There is no out-of-the-box scenario available for this connector.

E-mail connectors

Optional connectors

- Production (source)
 - Consumption (destination)
-

The e-mail connectors enable you to send and receive e-mails messages.

In the case of messages received by the e-mail (fetching) connector, their contents can be routed to an external application after processing in Connect-It (mapping). Example: An **E-mail-ServiceCenter** type scenario enables you to use the information from a message received by your e-mail server in order to create a ticket in ServiceCenter.

In the case of messages sent by the e-mail (sending) connector, data from an external application can be mapped to the different fields of an e-mail message in Connect-It. Example: In the provided Asset Management to E-mail scenario (**finreque.scn**, each record in the requests table gives rise to a message sent to the requester's supervisor.

This chapter presents:

- The known limitations of the e-mail connector.
- The configuration steps of the e-mail connector.

 **Warning:**

The e-mail connector works correctly with Lotus Notes and Outlook if these applications are installed on the computer on which Connect-It is installed.

Known limitations of the E-mail connector

Before using the e-mail connector in your scenarios, read the following two sections attentively.

Supported code pages

Please refer to the Compatibility Matrix available from Peregrine Systems Customer Support.

 **Warning:**

An English-language UNICODE version is available from Peregrine Systems Customer Support. Specify the connectors and code pages you need to process in order for them to determine whether the UNICODE version is compatible with your requirements.

Various limitations

The following list details the different limitations of the e-mail connector:

- The 'Priority' e-mail flag can only be processed with an SMTP server.
 - The connector only supports Base64 and Quoted-Printable encoding.
 - Unicode support is only available with SMTP and MAPI servers.
 - E-mails encoded in Quoted-Printable are correctly processed but error messages appear.
-

 **Warning:**





Configuration of the Connect-It client computer: In order for the e-mail connector to function correctly, Microsoft Internet Explorer 5.xx must be used on the computer on which Connect-It is installed.

Configure the E-mail connector (fetching)

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 5.3. Configure the E-mail connector (fetching)

 Name and describe the connector [page 17]
 Define the connection parameters [page 214]
 Processing message content [page 218]
 Define post-processing actions [page 218]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Define the connection parameters

This page enables you to select the messaging protocol of the e-mail (fetching) connector.

The number of fields in this page varies according to the protocol selected in the **Engine type** field. The protocols available for the e-mail (fetching) connector are:

- Lotus VIM (Vendor Independent Messaging)
Before modifying the connector, you must first declare in the system environment variables the access path of the **vim32.dll** file in the Lotus folder.
- POP3 (Post Office Protocol)
- Microsoft MAPI (Messaging Application Program Interface)
- Microsoft Extended MAPI
- IMAP

Lotus VIM

For the WIM protocol, two fields must be populated:

- Login

Indicate the login that allows you to access the VIM messaging system installed on your computer.

- Password
Enter the password associated with the login. This password is not visible when being entered and is stored in an encrypted format.

E-mail processing options

When using the VIM protocol, the following options are available:

- Retrieve unread e-mail messages only
- Use DOS/Windows text formatting: Carriage return (CR) and line return (LF).
This option must be selected if the destination application used DOS or 32-bit Windows as operating systems. In these two systems, carriage returns (CR) are followed by a line feed (LF). If the destination application uses UNIX as operating system, this option must be cleared.
- Activate the conversion of the e-mail body from the Lotus Notes internal code page
Select this option is certain characters in the message text are not correctly converted.

POP3

For the POP3 protocol, four fields must be populated:

- Pop3 server
Enter the name of your POP3 server.
- Connection port
Indicate the connection port used by the server on your computer. By default, the value of this field is '110'.
- Login
Indicate the login that allows you to access your POP3 server.
- Password
Enter the password associated with the login. This password is not visible when being entered and is stored in an encrypted format.

E-mail processing options

When using the POP3 protocol, the following options are available:

- ◆ Use DOS/Windows text formatting: Carriage return (CR) and line return (LF).

This option must be selected if the destination application used DOS or 32-bit Windows as operating systems. In these two systems, carriage returns (CR) are followed by a line feed (LF). If the destination application uses UNIX as operating system, this option must be cleared.

MAPI

For the MAPI protocol, two fields must be populated:

- Profile
Indicate the profile allowing you to access the MAPI messaging installed on your computer.
- Password
Enter the user password of your operating system (example: your Windows NT password). This password is not visible when being entered and is stored in an encrypted format.

E-mail processing options

When using the MAPI protocol, the following options are available:

- Retrieve unread e-mail messages only
- Use DOS/Windows text formatting: Carriage return (CR) and line return (LF).
This option must be selected if the destination application used DOS or 32-bit Windows as operating systems. In these two systems, carriage returns (CR) are followed by a line feed (LF). If the destination application uses UNIX as operating system, this option must be cleared.

Extended MAPI

For the Extended MAPI protocol, the following fields must be populated:

- Profile
Indicate the profile allowing you to access the Extended MAPI messaging system installed on your computer.
- Password
Enter the user password of your operating system (example: your Windows NT password). This password is not visible when being entered and is stored in an encrypted format.
- Directory
Enter the path of your directory.
You can enter a path to a local folder or to a folder on the e-mail server.

For example, for a folder on the server:

```
/Mailbox - Smith E/Inbox
```

For example, for a local folder:

```
/Personal Folders/Inbox
```

E-mail processing options

When using the Extended MAPI protocol, the following options are available:

- Retrieve unread e-mail messages only
- Use DOS/Windows text formatting: Carriage return (CR) and line return (LF).
This option must be selected if the destination application uses DOS or 32-bit Windows operating systems. In these two systems, carriage returns (CR) are followed by a line feed (LF). If the destination application uses UNIX, this option must be cleared.

IMAP

For the IMAP protocol, the following fields must be populated:

- IMPA Server
Enter the name of your IMAP server.
- Login
Indicate the login that allows you to access your IMAP server.
- Password
Enter the password corresponding to your login. This password is not visible when entered and is stored in an encrypted form.

Test button

This button enables you to test your connection.

To test your connection:

- 1 Enter your connection parameters.
- 2 Click **Test**.

The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.

- 3 Click **Close** to return to the configuration wizard.

Processing message content

The e-mail connector enables you to parse the text and attachments of e-mails received according to a DTD file. This file describes how the text and attachments of the e-mail are organized in XML elements.

Note:

The content of e-mails is analyzed, and all the text contained before the first XML tag (<?xml version="1.0">) and the last XML tag is deleted automatically. This analysis allows, for example, to isolate the content added automatically to XML elements by a mail server.

To process message content using a DTD file

- 1 Select the **Process contents of e-mail body** option.
- 2 Indicate the path of the DTD file enabling you to process the body of the e-mail in the **DTD file** field.

To process contents of the attachments using a DTD file:

- 1 Select the **Process contents of e-mail attachments** option.
- 2 Indicate the path of the DTD file enabling you to process the attachments of the e-mail in the **DTD file** field.

Define post-processing actions

This page enables you to specify which actions the connector will take after the documents that it produced have been processed.

Apply actions just after fetching messages

If select this option, the actions are applied according to the processing reports sent the scenario's other connectors.

For more information about processing reports, refer to **User's guide**, chapter **Processing reports**.

Available post-processing actions

In the two frames **In case of successful processing of message** and **On failure in processing a file** select one of the two following options:

- **Do not modify e-mail messages from server**
- **Flag e-mail messages as read**

- Delete e-mail messages from server

Using post-processing actions

To use post-processing actions, you must:

- ◆ Create a mapping between the **MailInfo.UniqueID** structure of the document type produced by the source connector and the **MailInfo.UniqueID** element of the **SuccessReport** document type.





For further information on the document types published by the E-mail connector, refer to section [Document types produced by the E-mail \(fetching\) connector](#) [page 222] of this chapter.

Configure the E-mail connector (sending)

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 5.4. Configure the E-mail connector (sending)

 Name and describe the connector [page 17]
 Configuring the connection [page 219]
 Define sending frequency [page 221]
 Parallelize consumption (advanced mode) [page 44]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Configuring the connection

The second page of the **Configure the connector** wizard enables you to select the messaging protocol of the e-mail (sending) connector.

The number of fields in this page varies according to the protocol selected in the **Engine type** field. The protocols available for the e-mail (sending) connector are:

- Lotus VIM (Vendor Independent Messaging)

Before modifying the connector, you must first declare in the system environment variables the access path of the **vim32.dll** file in the Lotus folder.

- Microsoft SMTP (Simple Mail Transfer Protocol)
- Microsoft MAPI (Messaging Application Program Interface)
- Microsoft Extended MAPI

Lotus VIM

For the WIM protocol, two fields must be populated:

- Login
Indicate the login that allows you to access the VIM messaging system installed on your computer.
- Password
Enter the password associated with the login. This password is not visible when being entered and is stored in an encrypted format.

E-mail processing options

If necessary, select one of the following options:

- Request a receipt
- Detect HTML messages

SMTP

For the SMTP protocol, four fields must be populated:

- SMTP server
Enter the name of your SMTP server.
- Connection port
Indicate the connection port used by your SMTP server. By default, the value of this field '25'.
- Display name
Specify the name to appear in the **'From'** field of the person receiving your message.
- Reply address
Enter the e-mail address (such as Reply@peregrine.com) to which the recipient of the message can reply to the message sent.

MAPI / Extended MAPI

For the MAPI and Extended MAPI protocols, these two fields must be populated:

- **Login**
Indicate the login that allows you to access the MAPI messaging system installed on your computer.
- **Password**
Enter the password associated with the login. This password is not visible when being entered and is stored in an encrypted format.

Test button

This button enables you to test your connection.

To test your connection:

- 1 Enter your connection parameters.
- 2 Click **Test**.
The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.
- 3 Click **Close** to return to the configuration wizard.

Detect HTML messages option

This option enables you to send messages containing HTML tags. The HTML code is interpreted and is not displayed in the source format for the body of the e-mail.

Define sending frequency

The third page of the **Configure the connector** wizard enables you to indicate a pause period between sending a given number of message to the messaging server. This can be used to avoid bottlenecks at the server level.

This page has two fields:

- **Number of messages**
Enter the number of messages sent to the server before pausing. Example: If you enter **100**, and **30 s** for the pause, 100 messages will be sent to the server before pausing for 30 seconds. After the pause, 100 messages are sent again.

 **Note:**

The values chosen for the **Pause length** and **Number of messages** will depend on the performance of your messaging system.

- **Pause duration**

This pause length uses the following conventions: **s** for seconds, **m** for minutes, **h** for hours. Example: For 100 messages, if you enter `30 s`, the connector will send the first 100 messages, stop for 30 seconds and then send the next 100 messages.

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Document types produced by the E-mail (fetching) connector

The e-mail (fetching) connector receives e-mails. It then produces instances of the **InMailMessage** document type containing the different components of the e-mail message. You can then map this information to a document type consumed by another connector.

InMailMessage document-type produced by the E-mail (fetching) connector

The following table presents the different components of a document type produced by the e-mail (fetching) connector.

Table 5.5. Document type produced by the E-mail (fetching) connector

Part of the produced document-type	Elements
InMailMessage root node	Message body (Body field) Message date (Date field) Message priority (Priority field) Subject of the message (Subject field)

Part of the produced document-type	Elements
From structure	<p>Electronic address of the author of the message (Address field)</p> <p>Name of the author of the message (Name field)</p> <p>Message author type (Type field)</p>
MailInfo structure	<p>Contains the following fields:</p> <ul style="list-style-type: none"> ■ UniqueID: This field is used by the processing reports sent by the other connectors. This field is the unique identifier of the mail message ■ SignedBy (VIM protocol): The person that signed the mail ■ CertifiedBy (VIM protocol): certification issued by the server ■ Trusted (VIM protocol): The message is signed and certified but the signature could not be verified. <p>It enables you to uniquely identify the messages produced by the connector.</p>
Attachment collection	<p>Contents of attachments represented by three fields:</p> <ul style="list-style-type: none"> ■ BlobFromMail field containing the attached data ■ FileFromMail field containing the file name ■ MimeType indicating the attachment type
Cc collection	<p>Electronic addresses of the persons to whom the message was sent as a carbon copy (Address field)</p> <p>Names of the persons to whom the message was sent as a carbon copy (Name field)</p> <p>Carbon copy type (Type field)</p>
To collection	<p>Electronic addresses of the persons to whom the message was sent (Address field)</p> <p>Names of the persons to whom the message has been addressed (Name field)</p> <p>Sending type of message recipients (Type field)</p>

Processing contents of messages produced by the e-mail (fetching) connector

When you specify a DTD file for the connector (refer to this section's [Configure the E-mail connector \(fetching\)](#) [page 214] sub-section), the bodies of messages produced by the e-mail (fetching) connector can be parsed. This process includes new elements in the produced document-type. Each element corresponds to an element in the DTD file.

Example 5.1. Example of a document type extended using a DTD

In the sample E-mail (fetching) to Asset Management scenario (**newemp1o.scn**), the **newemp1o.dtd** DTD adds a new structure (**amEmpIDept** structure) to the **InMailMessage** document.

Figure 5.1. E-mail connector - Non-extended **InMailMessage** document type

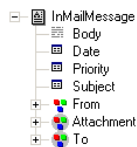
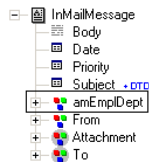


Figure 5.2. E-mail connector - **InMailMessage** document type extended using a DTD



Analysis of the DTD of the E-mail (receiving) connector using MAPI protocol

The **Address** element of the **From** structure and the **To** and **Cc** collections contain the E-Mail addresses of the sender or the recipient of the message. These addresses may be SMTP addresses, but they can also be internal addresses specific to the E-Mail server when the sender and recipient both belong to the same server.

For example, for a Microsoft Exchange server, the addresses will be SMTP addresses or Microsoft Exchange addresses such as:

- SMTP:john.doe@company.com for an external user
- EX:/O=PEREGRINE/CN=RECIPIENTS/CN=JDOE for an internal user

The **SmtAddress** element has been added to the **From** structure and the **To** and **Cc** collections. It contains the SMTP address, regardless of the whether the message was sent internally or not.

Thus, using the previous example, the corresponding values of the **SmtAddress** element will be:

- john.doe@company.com for an external user
- john.doe@company.com for an internal user

Document-types consumed by the E-mail connector (sending)

The e-mail (sending) connector sends e-mail messages. It publishes an **OutMailMessage** document type containing the different components of the e-mail message. The information from this document type is mapped to fields in another document type produced by another connector. (The Asset Management connector in the provided Asset Management to E-mail scenario (**finreque.scn**.)

Document type consumed by the E-mail (sending) connector

The following table presents the different components of a document type consumed by the e-mail (sending) connector.

Table 5.6. Document type consumed by the E-mail (sending) connector

Part of the consumed document-type	Elements
OutMailMessage root node	Message body (Body field) Message priority (Priority field) Subject of the message (Subject field)
From structure (SMTP protocol only)	Electronic address of the author of the message (Address field) Name of the author of the message (Name field) Message author type (Type field)

Part of the consumed document-type	Elements
Attachment collection	<p>Contents of attachments represented by three fields:</p> <ul style="list-style-type: none"> ■ BlobToSend field containing the attached data. ■ FileToSend field containing the file name.
Bcc collection	<p>Electronic addresses of the persons to whom the message was sent as a blind carbon copy (Address field)</p> <p>Names of the persons to whom the message was sent as a blind carbon copy (Name field)</p> <p>Blind carbon copy type (Type field)</p>
Cc collection	<p>Electronic addresses of the persons to whom the message was sent as a carbon copy (Address field)</p> <p>Names of the persons to whom the message was sent as a carbon copy (Name field)</p> <p>Carbon copy type (Type field)</p>
To collection	<p>Electronic addresses of the persons to whom the message was sent (Address field)</p> <p>Names of the persons to whom the message has been addressed (Name field)</p> <p>Sending type of message recipients (Type field)</p>

Processing of attachments

Attachments are files (images, video, sound, executables, etc.) sent with a message. In the document types published by the e-mail connector, these files are represented by fields of the **Attachment** collection.

Mapping attachments

This section describes how to map received and sent attachments.

Mapping received attachments

Received attachments are represented by three fields in the document type produced by the e-mail connector:

- **BlobFromMail** field containing the attached data.
- **FileFromMail** field containing the file name

- **MimeType** indicating the attachment type

To map the fields of the received attachments, map the **BlobFromMail** field to a binary field of the destination connector.

Example: In the sample **E-mail (fetching) to Asset Management** scenario (**newemp1o.scn**), the **BlobFromMail** field is mapped to the **Photo.blobData** field in the **amEmpIDept** document type of the **Asset Management** connector.

amEmpIDept	
bDepartment	[amEmpIDept.bDepartment]
EMail	[amEmpIDept.EMail]
FirstName	[amEmpIDept.FirstName]
FullName	[amEmpIDept.FullName]
IDNo	[amEmpIDept.IDNo]
MxMrs	[amEmpIDept.MxMrs]
Name	[amEmpIDept.Name]
Phone	[amEmpIDept.Phone]
Title	[amEmpIDept.Title]
Photo	IF [Attachment(0).MimeType] <> "ims
blobData	[Attachment(0).BlobFromMail]
Name	[Attachment(0).FileFromMail]

Mapping sent attachments

Sent attachments are represented by two fields in the document type consumed by the e-mail connector:

- **BlobToSend** field containing the attached data.
- **FileToSend** field containing the file name.

To map the fields of the attachments received:

- 1 Map a binary field of the source connector to the **BlobToSend** field.
- 2 Map a text file of the source connector to the **FileToSend** field. This field must contain the name of the attachment.

Example: The **Photo.blobData** and **Photo.Name** fields of the **amAsset** document type of the **Asset Management** connector are mapped to the **BlobToSend** and **FileToSend** fields.

Out-of-the-box scenarios - E-mail connectors

There is no scenario available for this connector.

The following out-of-the-box scenarios use this connector:

- [mail\mailac36\finconfi.scn](#) [page 380]
- [mail\mailac36\finrequc.scn](#) [page 380]
- [mail\mailac36\newemplo.scn](#) [page 380]
- [mail\mailac43\finconfi.scn](#) [page 380]

- mail\mailac43\finreque.scn [page 381]
- mail\mailac43\newemplo.scn [page 381]

LDAP connector

Optional connector

- Production (source)
 - Consumption (destination)
-

The LDAP connector enables you to process data from servers that use the LDAP protocol. It enables you, for example, to access the X500 electronic directories. The actual version of the LDAP connector enables you to read and write the data coming from an LDAP source.

About the LDAP connector

The LDAP connector uses the auto-description capacities of the LDAP v3 protocol and furnishes vast information to the user: Definitions of object classes, naming contexts, supported controls, etc.

Precautions of use for the LDAP connector

Limitations of the LDAP connector depend on the limitations of your LDAP resources (CPU allotted, memory allotted, etc.).

If you pass the limits of your LDAP resources while testing your connector or running a scenario, it is possible that the data processing failed partially or completely.










Configure the LDAP connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

Certain options concerning this connector can be enabled using the **Edit/Options/ Connector** menu. For more information, refer to the **User's Guide**, chapter **Scenario builder**, section **Connector**.

The following table presents all the pages in the connector's configuration wizard.

Table 5.7. Configure the LDAP connector

 Name and describe the connector [page 17]
 Configure the LDAP connector [page 229]
 Advanced configuration (advanced mode) [page 231]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Define document types (advanced mode) [page 42]
 Parallelize consumption (advanced mode) [page 44]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Configure the LDAP connector

The second page of the **Basic configuration of the connector** wizard enables you to configure your LDAP connection.

The fields of this page enable you to specify:

- The name of your LDAP server.
- The connection port of your server.
- A login
- A password
- The type of your server.
- The code page

LDAP server name

Indicate the name of your LDAP server. You can either enter the server name (for example: ldap-server.unil.ch) or its numeric IP address (for example: 207.68.137.42).

Server connection port

Indicate the connection port of your LDAP server. The default value of this field is 389 and should work in most cases. (This value is proposed by default by the protocol.)

Login

Indicate the login that enables you to access your LDAP server.

When you select the **Windows Active Directory** option, this login must respect the following syntax: full name@domain.name. For example, joe.doe@peregrine.com.

Note:

For iPlanet LDAP server, the login is the Distinguished Name of the user.

Password

Enter the password associated with the login. This password is not visible when being entered and is stored in an encrypted format.

Indicate the server type.

The available server types are:

- Novell Directory Service, Netscape Directory Server, etc.
- Microsoft Exchange
- IBM z/OS Security Server
- Microsoft Active Directory
- OpenLDAP

SSL connection

Select this option if you use a secure connection protocol.

According to the operating system used, proceed as follows:

- Solaris: for all versions prior to Solaris 9, you must apply patch 112438 for Solaris 8. This patch is available at the following site: www.sunsolve.com. Corrections are available for Solaris versions 2.6 and 2.7.
- AIX: for all versions prior to AIX 5.2, you must download the prngd package available at the IBM Web site (<http://www.ibm.com/servers/aix/products/aixos/linux/download.html>), as

the peripheral files `/dev/random` and `/dev/urandom` are not installed by default.

Specify the codepage of the server

The drop-down list enables you to indicate the codepage of your LDAP connector.

Test

This button enables you to test your connection.

To test your connection:

- Enter your connection parameters.
- Click **Test**.

The **Test the connection** window appears and tells you whether the connection succeeded or failed. If it failed, there will be messages explaining the cause.

- Click **Close** to return to the configuration wizard.

Advanced configuration (advanced mode)

This page enables you to specify:

- The size of the page.

The default value is 500.

The value of this option is only used by the Microsoft LDAP servers.

- The DN (Distinguished Name) of the schema

In rare cases, a schema's distinguished name is not automatically recovered by the LDAP connector. In this case, you must specify it in this field.

A warning in the Connect-It log will tell you if the distinguished name was not recovered. To obtain this name, refer to the **subschemaSubentry** entry of the DSE root (Directory Service Entries).

- The format of the dates.

To learn more about date formats, refer to this chapter's section [Additional information](#) [page 239] \ [LDAP date formats](#) [page 241].

The default value of this field is `%4Y%2M%2D%2H%2N%2SZ`.

The following symbols are those specified for the date and time:

- Y
Year
- M
Month

- D
Day
- H
Hour
- N
Minute
- S
Second
- Z

Mandatory symbol specifying that the date is in GMT 0 format.

The %[number][symbol] string specifies how many digits are used to express the indicated symbol. Example: %4Y indicates that the year must be expressed in four digits: 2003, 1997, etc.

If your server supports milliseconds, these are expressed by a decimal point. The number after the point expresses how many digits follow the decimal point.

Example: %4Y%2M%2D%2H%2N%.1SZ indicates that milliseconds can be expressed after the decimal point.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

DSE root entry

The **DSE root entry** field enables you to enter the root entry corresponding to the date of the server in your LDAP arborescence.

The default entry is **CurrentTime**.

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Last modification field

Certain document types have a field indicating the last modification date of a record in the database schema. In most cases, the **Modified-date** field is used. However, depending on your database schemas, the name of this field may differ.

The editable zone in this field enables you to indicate the field of the DSE used as the schedule pointer for the LDAP connector. The default value of this field is **modifyTimestamp**.

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Consumption directives

For information about how to enter a connector's consumption directives, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51].

A connector's consumption directives involve:

- Entering reconciliation parameters in the **Reconciliation** and **Advanced reconciliation** tabs.

For more information about these reconciliation tabs, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51], sub-section [Reconciliation](#) [page 52].

- Writing SCOPE and SEARCH DN clauses.

SCOPE clause

This clause determines the scope of your request depending on your entry point, identified by a DN (Distinguished Name) in an LDAP directory tree. Three options are available:

- Base
By selecting this option, your request applies to the data contained under your entry point without exploring the sub-trees of this entry point.
- First level
By selecting this option, your request applies to the data contained under your entry point, as well as under the **direct** sub-nodes of this entry point.
- Recursive
By selecting this option, your request applies to the data contained under your entry point and under **all** the sub-nodes of this entry point.

SEARCH DN clause

This clause enables you to indicate the DN (Distinguished Name) of your entry point in the tree structure of the directory to which you connected your LDAP server. When you launch your connector, the directory proposes several entry points: These are the naming contexts exposed by the LDAP connector.

These naming contexts appear in the drop-down list next to the **SEARCH DN** clause. You can, however, choose another entry point in the directory by editing this field manually. Each DN is made up of RDNs (Relative Distinguished Names), a list of the most frequent is given below.

RDN (Relative Distinguished Name)	Key
CommonName	CN
LocalityName	L
StateOrProvinceName	ST
OrganizationName	O
OrganizationalUnitName	OU
CountryName	C

RDN (Relative Distinguished Name)	Key
StreetAddress	STREET

 Note:

We advise against leaving the **SEARCH DN clause** field empty. This would mean querying the entire directory and would probably overrun the LDAP server's download limitations.

Production directives of the LDAP connector

The production directives of the LDAP connector enable it to extract the data from the LDAP source by taking into account a particular request.

This LDAP request is constructed using three different clauses that you write in the directives tab of your produced document-types:

- The SCOPE clause
- The SEARCH DN clause
- The FILTER clause

SCOPE clause

This clause determines the scope of your request depending on your entry point, identified by a DN (Distinguished Name) in the tree structure of an LDAP directory. Three options are available:

- Base
By selecting this option, your request applies to the data contained under your entry point without exploring the sub-trees of this entry point.
- First level
By selecting this option, your request applies to the data contained under your entry point, as well as under the **direct** sub-nodes of this entry point.
- Recursive
By selecting this option, your request applies to the data contained under your entry point and under **all** the sub-nodes of this entry point.

SEARCH DN clause

This clause enables you to indicate the DN (Distinguished Name) of your entry point in the tree structure of the directory to which you connected your LDAP

server. When you launch your connector, the directory proposes several entry points: These are the naming contexts exposed by the LDAP connector.

These naming contexts appear in the drop-down list next to the **SEARCH DN** clause. You can, however, choose another entry point in the directory by editing this field manually. Each DN is made up of RDNs (Relative Distinguished Names), a list of the most frequent is given below.

RDN (Relative distinguished names)	Key
CommonName	CN
LocalityName	L
StateOrProvinceName	ST
OrganizationName	O
OrganizationalUnitName	OU
CountryName	C
StreetAddress	STREET



Note:

We advise against leaving the **SEARCH DN clause** field empty. This would mean querying the entire directory and would probably overrun the LDAP server's download limitations.

FILTER clause

This clause enables you to filter records under the entry point that you selected. The **FILTER** clause must respect the syntax used by the LDAP requests. This syntax is presented in the RFC 2254.

Example: If you want to obtain the list of all people whose common name starts with 'A', you must enter the following clause:

```
(&(ObjectClass=personn)(cn=A*))
```

If you only want to obtain the list of people beginning with 'Ar', you must enter the following clause:

```
(&(ObjectClass=personn)(&(cn=A*)(!(cn=Ar*))))
```



Tip:

To verify that your LDAP server manages the dates that entries were modified in your directory:

- Create an LDAP document type.
- Enter the following value for the FILTER clause:

```
modifyTimestamp=*
```

This returns you all the entries in the directory whose modifyTimestamp field is populated. If this field is not populated for all the entries, then the connector needs to be configured to recover all entries at each startup.

LDAP filter syntax

The following table presents the syntax used to filter the LDAP data.

Table 5.8. LDAP filters

Filters
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 of section 4.1.5 of the RFC 2251
matchingrule = MatchingRuleId of section 4.1.9 of the RFC 2251
value = AttributeValue of section 4.1.6 of the RFC 2251

The following table indicates how to obtain certain characters in the values of the documents processed by the LDAP connector.

Table 5.9. Mapping table (character - ASCII value)

Desired character	ASCII value
*	0x2a
(0x28
)	0x29
\	0x5c
NULL	0x00

The character must be encoded like the "\" character (ASCII 0x5c), followed by two numbers representing the ASCII value of the encoded character. The case of the two hexadecimal characters is not important. Example: To verify that the **cn** RDN contains the * character, write the following filter:

```
"(cn=*\2a*)"
```

Examples of filters

Objective of the filter	To write
Look through all classes.	(objectclass=*)
Filter the people whose name starts with A .	(& (objectclass=person)(cn=A*))
Filter all types of people in LDAP.	((objectclass=person)(objectclass=organizationalPerson)(objectclass=inetOrgPerson)(objectclass=residentialPerson)(objectclass=newPilotPerson))
Filter the people whose name starts with A while ignoring those whose name starts with Ar .	(& (objectclass=person)(& (cn=A*)(!(cn=Ar*))))
To filter people whose name does not start with H, Y or E, you cannot write:	(& (objectclass=person)(!(cn=H*)(cn=Y*)(cn=E*))) "not" (!) is an unary operator.

Objective of the filter	To write
To filter the people whose name does not start with H, Y or E, you must write:	<code>(&(objectclass=person)(!(cn=H*)(!(cn=Y*))(!(cn=E*)))</code>

Additional information

This section presents additional information about the connector.

Sources of information concerning the LDAP protocol

For more information about the LDAP protocol, refer to the following RFCs (Requests for comments):

- 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 LDAP v3
- 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 - LDAP v2
- RFC 2279: UTF-8, a transformation format of ISO 10646
- RFC 2116: X.500 Implementations Catalog-96
- RFC 2255: The LDAP URL Format

Mapping the sole field of a collection to a field in another application

LDAP-directory entries are often single-field collections (each member of the collection corresponds to a field).

Each attribute represents an information type containing one or more values (multiple-value attribute). Example: A collection for which the value of the attributes is an e-mail address.

Ideally, you would map this field to the field of a collection in the destination application.

For the mapping, refer to **User's guide**, chapter **Mapping document types**, section **Mapping source elements to destination elements**.

If there are no available collections, you have two solutions:

- 1 Map one single member of the collection to the field of the other application.
- 2 Use a Basic script in the mapping. This script is used to concatenate the values of all the fields in a collection. The concatenated value obtained can then be mapped to the single field in the external application.

Map one single member of the collection to the target field.

To map one single field of the collection to the target field:

- 1 In the mapping edit window, map the field of the collection of the LDAP-document type to the target field.
- 2 Enter the number of the field that you want to map to this target field in the **Mapping script** field. (Reminder: Collection members in Connect-It are numbered sequentially from 0; 0 being the first member, 1 the second, etc.)

For example: To indicate the first attribute of a single-field collection, **Person.cn** (common name of a person), enter the following line in the mapping script: `[cn (0) . cn]`.

Using a Basic script

To use this Basic script:

- 1 In the mapping edit window, map the sole field of the collection of the LDAP-document type to the target field.
- 2 Enter the mapping script in the **Mapping script** field.

The first part of the script counts the number of members in the collection whereas the second part concatenates them in order to produce a single string. The syntax is as follows:

```
Dim iCollectionCount As Integer
iCollectionCount = PifGetItemCount("cn")
Dim strCollapse As String
Dim iItem As Integer
For iItem = 0 to iCollectionCount - 1
strCollapse = strCollapse + [cn(iItem)]
```



```
Next iItem
RetVal = strCollapse
```

Identifying the LDAP elements containing the most data


The LDAP connector publishes document types containing the following information:

- Object classes
- Fields of these classes
- Heritage between these classes

The most important information when using an LDAP data source is:

- The object classes containing the data that interests you.
Example: the tables containing records.
- The heritages between the classes enable you to write high-performance requests.
Example: Do you want to use the **Top**, **Person** or **OrganizationalPerson** class?

To obtain this information:

- 1 Select the LDAP connector in the scenario that you created or edited.
- 2 Select the **Document types** tab.
- 3 Create a produced document-type whose root element is **Top**.
All the LDAP directories publish a **Top** object class that serves as a root element for all the server classes.
- 4 Click  to view the data in the LDAP source.
- 5 Search the classes returning the most data in order to use this data in the definition of your document types.

LDAP date formats

The format of **timestamp** fields for the LDAP directories respect the following syntax:

```
[year] [month] [day] [hour] [minute] [second] Z
```

Example: 22 hours 40 minutes 34 seconds, April 5, 2003 is expressed by the following string: **20030405224034Z**.

The final **Z** indicates that the date is on the GMT 0 (Greenwich Meridian Time) time zone.

Certain LDAP servers use other formats for timestamps. Example: Certain servers add a digit corresponding to milliseconds (this digit is expressed by a decimal point.) Example: **20030405224034.5Z**

Out-of-the-box scenarios - LDAP connector

The following out-of-the-box scenarios use this connector:

- [ldap\ldapac36\complete.scn](#) [page 389]
- [ldap\ldapac36\simple.scn](#) [page 390]
- [ldap\ldapnote\names.scn](#) [page 390]

Command-line connector

Base connector

- Production (source)
 - Consumption (destination)
-

This section describes the Command-line connector.

- In consumption mode, the Command-line connector enables you to execute a command line on the computer on which Connect-It is installed.
- In production mode, the Command line connector enables you to recover from a command line executed on a computer where Connect-It is installed:
 - 1 Return values
 - 2 Standard output
 - 3 Error output

The connector produces documents automatically. In a scenario, each time the Command line connector consumes a document (coinciding with the launch of a command line sent by another connector), it automatically produces the corresponding document.

Compatibility of the Command line connector

The Command line connector enables you to launch a command line on any operating system on which Connect-It can be installed.




In Unix, the connector only recovers the return code of commands. The standard output and the error output cannot be recovered.

Configure the Command line connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 5.10. Configure the Command line connector

 Name and describe the connector [page 17]
 Execution frequency [page 243]
 Parallelize consumption (advanced mode) [page 44]

The configuration of a connector in Connect-It is performed using a wizard.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Execution frequency

This page enables you to specify a pause between the processing of a determined number of documents. This prevents your computer, or an involved server, from being overloaded.

This page contains the following fields:

- Number of documents

You enter the number of documents processed by your computer between each pause. Example: If you enter the number **100** and a pause of **30 s**, your computer will process 100 documents followed by a 30-second pause. After this pause, another 100 documents will be processed, and so on.

 **Note:**

The values of the **Pause duration** and **Number of documents** fields depend upon your computer's performance level. You should conduct some tests before setting values for these fields.

- Pause duration

This pause must use the following conventions: **s** for seconds; **m** for minutes; **h** for hours. Example: If you enter **30 s** for 100 documents, your connector

will process the first 100 documents, stop after 30 seconds and then process the following 100 documents.

- **Timeout**

In order for this field to be available, you need to have already selected the **Kill the process if it has not finished before timeout** option. Enter a value in seconds.

 **Warning:**

In Unix, the **Kill the process if it has not finished before timeout** option is not available.

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Published document-types

The Command-line connector publishes two document types:

- The **Command** document type

This document type corresponds to a command line executed on the computer on which Connect-It is installed. This published document-type enables you to define the document types consumed by the Command-line connector.

- The **CommandReturn** (production) document type.

This document type corresponds to the return values of a command line executed on the computer on which Connect-It is installed. This published document-type enables you to define the document types produced by the Command-line connector.

The Command document type

This document type has three fields:

- The **Arg** field

This field corresponds to your command line arguments.

- The **Name** field

This field corresponds to the full path of the executable (the target). For example: `C:\Program Files\Peregrine\AssetCenter\`

If your executable is defined in your operating system's Path environment variable, you can directly specify the name of the executable. Example: `explorer.exe`.

- The **Path** field
This field corresponds to the command-line execution folder. It corresponds to the field **Start in**.

The CommandReturn document type

This document type has three fields:

- The **ReturnCode** field
This field corresponds to the return code of the executed command line.
- The **StdErr** field
This field enables you to obtain data on the standard error output.
- The **StdOut** field
This field enables you to obtain data from the standard output.



Important:

In Unix, the **StdErr** and **StdOut** fields are not available.

BIOS commands

Certain commands are executables and others are BIOS commands.

To use a BIOS command in the command-line connector, you must use the following syntax with the **cmd.exe** executable:

```
cmd /C move d:\HR\Employee.xml d:\HR\done\Employee.xml
```

In Connect-It, the **Name** element is mapped with "cmd /C move" and the **arg** element mapped with "d:\HR\Employee.xml d:\HR\done\Employee.xml".

Consumption directives

Only one consumption directive is available for the Command-line connector. This directive corresponds to the **Synchronous execution** option available in the edit window of the document types consumed by the connector. In synchronous mode (by default), a command line is executed after the previous

command line has finished executing. All messages resulting from this execution (including error messages) will be processed by the connector.

Out-of-the-box scenarios - Command-line connector

There is no out-of-the-box scenario available for this connector.

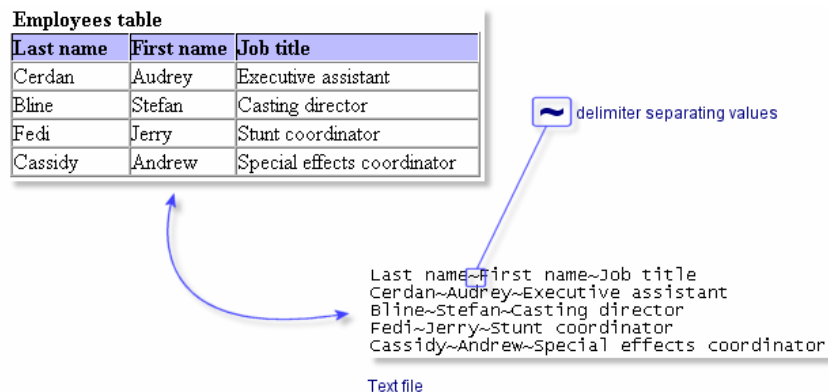
Delimited-text connector

Base connector

- Production (source)
- Consumption (destination)

The Delimited-text connector processes files containing a flat representation of database records. Field values are either aligned in columns of fixed width (expressed as a number of characters) or separated by a delimiter character (a punctuation mark, tab, or any specified character).

Figure 5.3. Delimited-text connector - Representation of data in a text file



The connector can work:

- In read mode

In this mode, data is read from a text file so that it can then be imported into a database via a Connect-It connector (Database, Asset Management, ServiceCenter connectors, etc.)

- In write mode

In this mode, text files are created. The data contained in these file originates from a database via a Connect-It connector (Database, Asset Management, ServiceCenter connectors, etc.).

The text enables the:

- Processing of text files on the computer or the network on which Connect-It is installed.

These files can be compressed using the **gzip** command.

- Processing of text files on FTP sites and Web sites (read-only for the latter).

Known limitations of the Delimited-text connector

The only limitation of the Delimited-text connector concerns the format of **Date** and **Date and Time** type fields. In order to be processed correctly, these fields must be formatted according to the regional settings of the computer on which Connect-It is installed.

This limitation can be bypassed by using a **string** type field and associating it with a user format during the mapping.

To learn more about user formats, refer to the **User's guide** chapter **Writing scripts**, section **User formats**.

Configure the Text connector (read)









A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

Certain options concerning this connector can be enabled using the **Edit/Options/ Connector** menu. For more information, refer to the **User's Guide**, chapter **Scenario builder**, section **Connector**.

The following table presents all the pages in the connector's configuration wizard.

Table 5.11. Configure the Delimited-text connector (read)

 Name and describe the connector [page 17]

 Choosing a processing mode [page 248]		
 Selecting a connection protocol [page 248]		
 Choose a file or folder [page 248]	 Connecting to an HTTP Web site [page 250]	 Connect to the FTP server [page 252]
 Post-processing actions [page 249]	 Define post-processing actions [page 253]	
 Choose a description file [page 253]		

 **Warning:****Path of a file or folder**

Often when configuring a connector, you must indicate the path of a file or folder in the configuration wizard. If this file or folder is in a folder associated with a network drive on your computer, you must never indicate the letter of this drive; only indicate the folder associated with this letter.

Indicating the network drive in the path stops the scenario from working correctly when it is associated with a service. For more information about this, consult the **User's guide**, chapter **Putting a scenario into production**, section **Creating a Connect-It service**, sub-section **Creating a Connect-It service in Windows/Scenario path**.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors \[page 15\]](#), section [Name and describe the connector \[page 17\]](#).

Choosing a processing mode

This page enables you to indicate if the connector is used in read or write mode.

Selecting a connection protocol

The three available options are:

- HTTP Web site
- FTP server
- Local or network file(s)

Choose a file or folder

If you choose to read local or network files, you must:

- 1 Choose a file or folder

- 2 Select an action to perform after the files have been processed.

Choose a file or folder

This page enables you to select the text file or files that your connector must read. You can choose between two options:

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

Read a file

If you select this option, the wizard will ask you to enter the path of the text file on your computer or your network.

Read files in a folder

If you select this option, the wizard will ask you to populate the following fields:

- **Folder name**
Enter the path of the folder that contains your files.
- **Extension**
Indicate the file name extension that the connector must read. For example, 'txt'.

Read the sub-folders

If you select this option, the connector will also read the files in the sub-folders of the selected folder.

Post-processing actions

After the text connector has read a file, Connect-It will give you three options:

- Leave it in the folder
- Delete it from the folder
- Move it to another folder
If you choose this last option, you must indicate the path of the folder to which you want to move the processed file.

You must specify one of these options in the case of failure, and one in the case of success, in processing the text files by your connector.

In order to apply a post-processing action to the documents consumed by the other connectors and mapping boxes of your scenario, you must use the

processing reports that each one produces. For more information about processing reports, refer to the **User's guide**, chapter **Processing reports**.

Connecting to an HTTP Web site

If you have chosen to read the XML files on a Web site, you must indicate the HTTP connection parameters:

- **Address**
Enter an address of the type: **[protocol] ://[address] : [port]/[path]**.
The [address] part is sometimes the only mandatory part.
The usual port number for an HTTP server is 80.
- **Login**
Enter the login that enables you to access the chosen site.
- **Password**
Enter the password corresponding to your login.

Secured connection (HTTPS protocol)

This option enables you to indicate if you want to connect to your site via a secured connection (HTTPS).

Important:

If the [protocol] part of the Web site's address does not corresponds to the HTTPS protocol, the selection of the **Secured connection (HTTPS protocol)** option forces the use of an HTTPS protocol.

The usual port number for an HTTPS server is 443.

Example of an HTTP address

Address	Secured connection (HTTPS protocol)	Connection mode
https://a.b.c.d/	Selected	https port 443
https://a.b.c.d:9000/	Selected	https port 9000
http://a.b.c.d:9000/	Not selected	https port 9000
http://a.b.c.d:9000/	Selected	https port 9000
https://a.b.c.d:9000/	Not selected	https port 9000
http://a.b.c.d:443/	Not selected	https port 443

Address	Secured connection (HTTPS protocol)	Connection mode
https://a.b.c.d:80/	Selected	https port 80

Client certificate

This field enables you to select an HTTPS certificate from among those present on your computer.

Important: If you change or delete a client certificate specified in this field after the connector has already read or written to these documents, you must close and reopen the Scenario builder in order for this modification to be accounted for.

Manage the list of client certificates

The list of client certificates proposed by the configuration wizard corresponds to the list of certificates in your **Root console/ Certificates - current user/ Personal/ Certificates** folder of the Microsoft Management application.

To add certificates to this folder using Windows XP:

- 1 Select **Run** from the Windows **Start** menu.
- 2 Enter **MMX** in the **Open** field.
- 3 Select **File/ Add/Remove Snap-in**.
- 4 Click **Add** in the window that is displayed.
- 5 Select **Certificates** in the window that is displayed.
- 6 Click **Add**.
- 7 Select **My user account** in the dialog box that is displayed.
- 8 Click **Finish**.
- 9 Click **OK**.
- 10 Add or delete the files in the **Certificates - Current user/ Personal/ Certificates folder**.

Do not verify the server identity

Use this option to support a proxy server in a secure connection.

If a proxy server is installed, the address shown in the server certificate does not match the one shown for the proxy server. Selecting this option disables the automatic verification of the server's identity.

FTP server

If you have chosen to read the text files on an FTP site, you must:

- 1 Select the FTP connection parameters
- 2 Select an action to perform after the files have been processed.

Connect to the FTP server

On this page, you must populate three fields, which enable you to connect to your FTP server:

- **Server**
Enter the name of your FTP server
- **Login**
Enter the login that enables you to access the chosen site.
- **Password**
Enter the password corresponding to your login.

You can test the validity of your connection using the **Test** button.

In the next page of the wizard, select either the **Read files** or **Read folders** option according to your needs. If you select the **Read folders** option, you must indicate:

- **Path** of the folder on the FTP site.
- **File extension names**.
By default, the value of this field is **txt**.

Read files

If you select this option, the wizard will ask you to enter the path of one or more text files on your computer or your network.

Read folders

If you select this option, the wizard will ask you to populate the following fields:

- **Path**
Enter the path of the folder that contains your files.
- **Extension**
Enter the file name extension that the connector must read.

Read the sub-folders

If you select this option, the connector will also read the files in the sub-folders of the selected folder.

Note:

During a connection in FTP mode, you might see the following error: `Error: 12015`.

This error means that the directory is not available, usually because of too many simultaneous connections.

Define post-processing actions

After the text connector has read a file, Connect-It will give you three options:

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

If you choose this last option, you must indicate the path of the folder to which you want to move the processed file.

You must specify one of these options in the case of failure, and one in the case of success, in processing the text files by your connector.

For more information on post-processing actions and archival, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Using post-processing actions

To use post-processing actions, you must:


- ◆ Create a mapping between the **UrlFileInfo.Path** structure of the document type produced by the source connector and the **UrlFileInfo.Path** element of the **SuccessReport** document type.

For further information on this structure, refer to section [Published document types](#) [page 268] of this chapter.


Choose a description file

In order to process a text file, the connector must use a description file. Two scenarios are possible:

- The description file already exists.

Indicate its path in the **DSC file** field. Click  to create a location for the file.

- The description file does not exist.

Indicate the path and the name of the description file you want to create in the **DSC file** field and click  to launch the description file creation wizard.

This wizard is presented in the [Creation of the description file \(DSC file\)](#) [page 262] sub-section of this section.



Note:

If you use two delimited text connectors in the same scenario, a backup copy is created for the first DSC file in order not to be overwritten by the second connector.

If you are using a UNICODE version, an additional option enables you to select the encoding type to use.

Options associated with the Delimited-text connector

You can access the options of the Text connector using the menu **Edit/ Options/ Connector/ Delimited text and XML**.

The options associated with the Text connector in FTP protocol are the following:

- ◆ Copy locally the files to read from FTP server.

Select this option to make a local copy of the files on an FTP server and to read the data from the local file.

This option must be selected if the network configuration does not enable a connection to be maintained on the FTP server long enough for a file to be processed.










Configure the Text connector (write)

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

The configuration of the text (write) connector varies depending on the selected connection protocol.

Table 5.12. Configure the Delimited-text connector (write)

 Name and describe the connector [page 17]		
 Processing mode [page 255]		
 Selecting a connection protocol [page 255]		
 Choosing a file or folder [page 257]	 HTTP Web site [page 256]	 FTP connection parameters [page 258]
 Behavior between two sessions [page 259]	 Behavior between two sessions [page 259]	
 Choose a description file [page 259]		

 **Warning:**

Path of a file or folder

Often when configuring a connector, you must indicate the path of a file or folder in the configuration wizard. If this file or folder is in a folder associated with a network drive on your computer, you must never indicate the letter of this drive; only indicate the folder associated with this letter.

Indicating the network drive in the path stops the scenario from working correctly when it is associated with a service. For more information about this, consult the **User's guide**, chapter **Putting a scenario into production**, section **Creating a Connect-It service**, sub-section **Creating a Connect-It service in Windows/Scenario path**.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors \[page 15\]](#), section [Name and describe the connector \[page 17\]](#).

Processing mode

This page enables you to indicate if the connector is used in read or write mode.

Selecting a connection protocol

The two available options are:

- HTTP
- FTP
- Local or network file(s)

HTTP Web site

If you choose to write files on an HTTP site, you must:

- Select the HTTP connection parameters.
- Select an action to perform after the files have been processed.

HTTP connection parameters

On this page, you must populate three fields, which enable you to connect to your HTTP server.

- Address
Specify the connection address to the HTTP server.
- Login
Enter the login that enables you to access the chosen site.
- Password
Enter the password corresponding to your login.

Write command

There are two write commands used to write to an HTTP server:

- POST
Sends data to the program located at the indicated address. It is different from the PUT method in that the sent data must be processed.
- PUT
Sends data so that it can be saved at the indicated address.

Select the command used by your HTTP server.

HTTPS protocol

Select this option if you want to connect to a server using this secure protocol.

The usual port number for an HTTPS server is 443.

The usual port number for an HTTP server is 80.

Client certificate

This field enables you to select an HTTPS certificate from among those present on your computer.

Important: If you change or delete a client certificate specified in this field after the connector has already read or written to these documents, you must close

and reopen the Scenario builder in order for this modification to be accounted for.

Manage the list of client certificates

The list of client certificates proposed by the configuration wizard corresponds to the list of certificates in your **Root console/ Certificates - current user/ Personal/ Certificates** folder of the Microsoft Management application.

To add certificates to this folder using Windows XP:

- 1 Select **Run** from the Windows **Start** menu.
- 2 Enter **MMX** in the **Open** field.
- 3 Select **File/ Add/Remove Snap-in**.
- 4 Click **Add** in the window that is displayed.
- 5 Select **Certificates** in the window that is displayed.
- 6 Click **Add**.
- 7 Select **My user account** in the dialog box that is displayed.
- 8 Click **Finish**.
- 9 Click **OK**.
- 10 Add or delete the files in the **Certificates - Current user/ Personal/ Certificates** folder.

Do not verify the server identity

Use this option to support a proxy server in a secure connection.

If a proxy server is installed, the address shown in the server certificate does not match the one shown for the proxy server. Selecting this option disables the automatic verification of the server's identity.

Choosing a file or folder

If you have chosen to write local or network files, you must:

- 1 Choose a file or folder
- 2 Select an action to perform after the files have been processed.

Choosing a file or folder

This page enables you to select the text file or files that your connector must write. You can choose between two options:

Write documents to one single folder

Enter the path of a text file on your computer or your network. If this file already exists, its contents will be overwritten when you run your connector.

Write to a different file for each document

Indicate the folder in which the files will be written. The names of these files will correspond to the generic name that you have entered in the text-connector consumption directives. (Consult the [Consumption directives of the Delimited-text connector \(write\)](#) [page 268] sub-section of this section.)

FTP server

If you have chosen to write files on an FTP site, you must:

- 1 Select the FTP connection parameters.
- 2 Select an action to perform after the files have been processed.

FTP connection parameters

On this page, you must populate three fields, which enable you to connect to your FTP server.

- Server
Enter the name of your FTP server
- Login
Enter the login that enables you to access the chosen site.
- Password
Enter the password corresponding to your login.

You can test the validity of your connection using the **Test** button.

In the next page of the wizard, select the option **Write the documents to a single file** or **Write to a different file for each document** according to your needs.

Write a file

Enter the name and the path of the file that you want to create with the help of the text connector. If this file already exists, its contents will be overwritten.

Write to a folder

Indicate the folder in which the files will be written. The names of these files will correspond to the generic name that you have entered in the consumption

directives of the text connector. Refer to the [Consumption directives of the Delimited-text connector \(write\)](#) [page 268] sub-section of this section.

Behavior between two sessions

This page enables you to select how documents are processed between two data-writing sessions. The behavioral options are different if you write data to one single file or if you write it to several files in a folder.

Write the documents to a single file

To write to a single file, you have the following options:

- Append to the same file
If you choose this option, the connector starts writing data to the open file again the moment the previous session is interrupted.
- Overwrite the last file
The connector deletes the file in which it wrote data during the previous session.
- Number the different files
The connector creates a file whose name is the same as the previous file, but incremented by 1 unit. Example: `file.xml`, `file1.xml`, `file2.xml`, etc.


Write to a different file for each document


To write in several files, you have the following options:

- Delete all previous files and start numbering them from the beginning
The files in the folder are all deleted (all the files dating from all the previous session; not just from the last one).
The connector starts incrementing all the files it writes.
- Continue numbering files
The connector starts writing files again and incrementing them without deleting the existing files.

Choose a description file

In order to process a text file, the connector must use a description file. Two scenarios are possible:

- The description file already exists.
Indicate its path in the **DSC file** field. Click  to create a location for the file.
- The description file does not exist.

Indicate the path and the name of the description file you want to create in the **DSC file** field and click  to launch the description file creation wizard. This wizard is presented in the [Creation of the description file \(DSC file\)](#) [page 262] sub-section of this section.

 **Note:**

If you use two delimited text connectors in the same scenario, a backup copy is created for the first DSC file in order not to be overwritten by the second connector.

Blind mode

If you select **Activate blind mode**, the Delimited-text connector will create text files without a description file. The values contained in these text files are separated by one of the chosen delimiters.

The delimiters to choose from are:

- **Tab**
- **Comma**
- **Semi-colon**
- **Space**
- **Other**

Enter the symbol that you want to use as a separator in the **Other** field.

Operation of the Delimited-text connector in blind mode

To operate the Delimited-text connector in blind mode:

- Configure your Delimited-text connector by selecting the **Activate blind mode** option.
 - Choose a database-type connector (for example: the Asset Management connector) for which you create produced document-types whose fields contain values that you want to create in the text file or files.
-

 **Note:**

In blind mode, only the values of fields directly under the root node of produced document-types are written in the text files. If the document type includes fields of structures or collections under the root node, their values are not written to the text file or files created.

- Directly link the database-type connector to your Delimited-text connector in your scenario **without passing through a mapping box**. (To create a direct link, just hold down the **Shift** key.)
- Put your database-type connector into production (**Produce now** command in the **Tools** menu).

Text files created in blind mode

If the **Write documents to one single folder** option is selected in the **Basic configuration** wizard, the file created in blind mode:

- Will be composed of lines, all of which correspond to the documents consumed by the Delimited-text connector.
- Will have a name that corresponds to that of a document type produced by the source connector.

If the **Write to a different file for each document** option is selected in the **Basic configuration** wizard, the files created in blind mode:

- Will contain values that are found in one of the documents consumed by the text connector.
- Will have a name composed of:
 - The name of the document type produced by the connector source (example: Test).
 - A number corresponding to the document production order (example: **Test_01**, **Test_02**, **Test_03**, etc.).

Writing the DSC file

The **Write DSC file** option enables you to create a description file in blind mode. If you select **Enable blind mode**, then this option is also selected by default.

If you use the name of an already existing file, Connect-It will automatically save it, and attach to it a number: **_01**. This description file uses the delimiter chosen for blind mode.

Encoding to use

If you are using a UNICODE version, an additional option enables you to select the encoding type to use:

- UTF - 8
- UNICODE

Creation of the description file (DSC file)

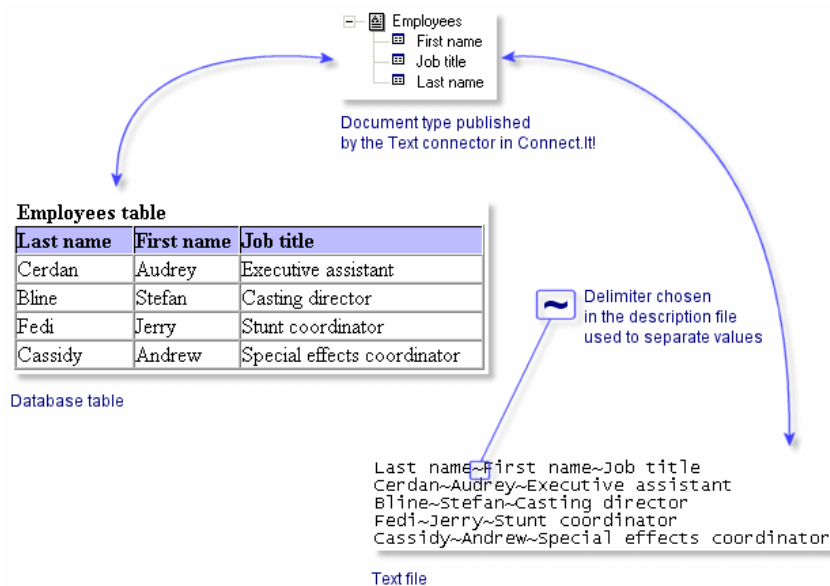
A description file (file name extension **.dsc**) describes the way that the text-file data is organized. This data corresponds to the field values in the database tables.


The Text connector publishes the document types created in the description file.

Example: A description file specifies that:





- The text file contains data from one single table (Employees table).
- The first line of the file contains the column headers (each column corresponds to a field in a table).
- The "~" is used to separate values.

Figure 5.4. Delimited-text connector - Processing data



To access the wizard that enables you to create a description file, click  on the **Choose a description file** page in the **Configure the connector** wizard.


The following table presents all the pages in the description file's configuration wizard.

-  [Select a file for the preview \[page 263\]](#)
-  [Specify the column delimiters \[page 264\]](#)
-  [Specify the data-processing options \[page 264\]](#)
-  [Specify the column header and type \[page 267\]](#)


Select a document type

This page helps you create document types published by the Delimited-text connector. Each published document-type corresponds to a database table.

To create a document type:

- Click .
- Click the highlighted text in the **Document type** column and enter the name of the document type you want to create.
- Click **Next** to continue on the next page.

To delete a document type:

- Select a document type.
- Click .

Select a file for the preview

This page enables you to indicate a file containing delimited text. The wizard uses this file to let you view how the description file is processing data.

File to preview

Indicate a file containing data delimited by spaces, tabs or special characters. This file must correspond to the files processed by the connector in the scenario.

Tip:

To optimize the display of data for previewing, we recommend using as your preview file:

- A local copy of one of the files that the connector will process when you put your scenario into production.
- A small file, although there are no limits pertaining to size.

Number of lines to preview

Indicate the number of lines that the wizard will display in the **Data previewing** zone.

Specify the column delimiters

This page helps you define how your text file data will be separated. Two options are available:

- Fixed width

This option helps you define the fixed width for each of your text-file columns in the **Width** field on the **Columns** page of the wizard.

To create columns, click directly inside the preview zone.

To resize a column, click on this column and pull it until you obtained the desired size.

To delete a column, click it and drag it outside of the preview zone.

- Delimiter

This option helps you define the delimiter that will separate the values in your text file. The available delimiters are:

- **Tab**
- **Comma**
- **Semi-colon**
- **Space**
- **List of characters used as delimiters**

Indicate the number of characters that you want to use as delimiters. Do not use any delimiters to separate the entered characters.

Specify the data-processing options

This page lets you specify the different ways you can process data from your delimited-text files.

The following text delimited by a semi-colon illustrates the different ways data can be processed:

```
Name;Brand;ID
Inspiron;Dell;Comp111
"Inspiron";Dell;Comp112
'Inspiron';Dell;Comp113
"Inspiron" ;Dell ;Comp114
\\"Inspiron";Dell;Comp115
Inspiron\;;Dell;Comp\;116
```

Enter the column names

Select this option to enter the column headers in the produced document.

Import column titles from the first line

Select this option so that the values of the first line of the delimited-text file are used as the column title.

 **Note:**

The first line used as a column header is also processed in the document type produced.

	Name	Brand	ID	Column4	Column5
1	Inspiron	Dell	Comp111		
2	"Inspiron"	Dell	Comp112		
3	Inspiron	Dell	Comp113		
4	"Inspiron"	Dell	Comp114		
5	\ "Inspiron"	Dell	Comp115		
6	Inspiron\		Dell	Comp\	116

 **Warning:**

Once you validate this option, you cannot change it.

Do not generate errors if a line contains a number of columns different than what is indicated in the description

Select this option to ignore lines different from the description of the delimited text.

If the line contains a number of columns different than what is indicated in the description, it is ignored and no error is generated.

Quotation character

Indicate which kind of quotation marks to use in your text:

- Single quotation marks (')
- Double quotation marks (")
- All other symbols except for the chosen delimiter(s).

If you choose double quotation marks, you will obtain the following result:

	Name	Brand	ID	Column4	Column5
1	Inspiron	Dell	Comp111		
2	Inspiron	Dell	Comp112		
3	'Inspiron'	Dell	Comp113		
4	Inspiron	Dell	Comp114		
5	\Inspiron	Dell	Comp115		
6	Inspiron\		Dell	Comp\	116

Select the **Keep quotes around values** option to have quotation marks that you selected appear in the preview zone.

Commentary starting-line

Indicate the character string signaling commentary in your delimited-text files. The default value is //.

In the data preview zone, the commentary is highlighted.

If you choose the value \\, for our example text, you will obtain the following result:

	Name	Brand	ID	Column4	Column5
1	Inspiron	Dell	Comp111		
2	"Inspiron"	Dell	Comp112		
3	'Inspiron'	Dell	Comp113		
4	"Inspiron"	Dell	Comp114		
5	\\ "Inspiron"	Dell; Comp1	15		
6	Inspiron\		Dell	Comp\	116

Escape character

Enter an escape character.


If you choose the character \ for your example text, you will obtain the following result:

	Name	Brand	ID	
1	Inspiron	Dell	Comp111	
2	Inspiron	Dell	Comp112	
3	'Inspiron'	Dell	Comp113	
4	Inspiron	Dell	Comp114	
5	\\\"Inspiron	\"Dell, Com	p115	
6	Inspiron.	Dell	Comp.116	


Specify the column header and type

The values in each column of a text file correspond to the fields in a database table and, consequently, to the created document-type fields for each description file.

This page enables you to:

- Create and name each column of your text files.
 - To create a column:
 - Click .
 - Click the highlighted text in the **Name** column and enter the name of the created-document type.
 - Enter the expected value-type for each created column.
 - To enter a value type, click the **Type** column and select one of the choices from the drop down list that appears. The proposed value types are **Text**, **Number**, **Date**, **Date and time**, **Currency**.
 - Set the width of each column when the **Fixed width** option is selected on the **Delimiters or fixed width** page.
 - To enter a value for the width, click the **Width** column and enter the number of characters required.

To delete a column:

- Select a column.
- Click .

Ordering columns

The order of columns depends of the order of the fields in the text file.

To change the order of the columns:

- Select a column.
 - Replace this column by clicking ↓ or ↑.
-

 Note:

Although possible, there is no real reason to create or rename a column in delimited mode.

Published document types

The delimited text connector publishes the following structure for all document types:

- ◆ `UrlFileInfo`

This structure is mainly used for post-processing actions.

The `UrlFileInfo` structure contains the following fields:

- **creationdate**
This field corresponds to the creation date of the document
- **lastmodificationdate**
This field corresponds to the last modification date of the document
- **line**
This field corresponds to the line number of the document
- **path**
This field corresponds to the path of the document

Consumption directives of the Delimited-text connector (write)

The consumption directives indicate for each document type consumed by the text connector:

- A generic name
By default, the value of this field is the name of the document type consumed by the Text connector.
- A file name extension
Enter this name without the period mark. Example: Enter `txt` instead of `.txt`; `"txt"` is the default value.

These two fields will only be used if you have chosen the **Write to a different file for each document** option during the configuration of your Text connector.

For more information, refer to the section [Configure the Delimited-text connector \(write\)](#) [page 255], sub-section [Choosing a file or folder](#) [page 257].

When you run your scenario, the files that are written will have a composite name, which is created from the generic name, a number corresponding to the file's creation order (_01, _02, _03, etc.) and from the file-name extension indicated.

Example: You enter the values **employee** and **txt**. The created files have the following names: **employee_01.txt**, **employee_02.txt**, **employee_03.txt**, etc.

 **Warning:**

If your Text connector consumes several document types, do not use the same generic name twice: The most recently created files will overwrite the previously created files having the same generic name.

Figure 5.5. Delimited-text connector - Consumption directives

Generic name	anAssetDst
Extension file name	txt

Out-of-the-box scenarios - Delimited-text connector

There is no out-of-the-box scenario available for this connector.

Web Service connector

Optional connector

- Spontaneous production (source)
- Consumption (destination)

The Web Service connector enables you to interact with a Web service. A Web service provides a service or a set of services composed of operations. SOAP, HTTP GET/POST and MIME are the main communication protocols used by these Web services.

The SOAP message protocol (Simple Object Access Protocol) is a data-exchange standard for distributed environments based on the use of communication protocols (transport layers) such as HTTP, SMTP or other Internet protocols, and provides a data-exchange mechanism based on XML messages.

Using the WSDL (Web Service Description Language), the Web Service connector obtains from the service an auto-description of a given Web service. This auto-description contains the protocols used by the service and the list of formats that an operation is capable of processing.

Lists of Web services are available on numerous sites. For example: <http://www.xmethods.com>.

In an integration scenario, the Web Service connector consumes a document that it sends as a request to a Web service. Then it receives a response that it automatically transforms into a produced document type. On receiving a document, this connector sends a query to the Web service and produces a document on receiving the response from the Web service.

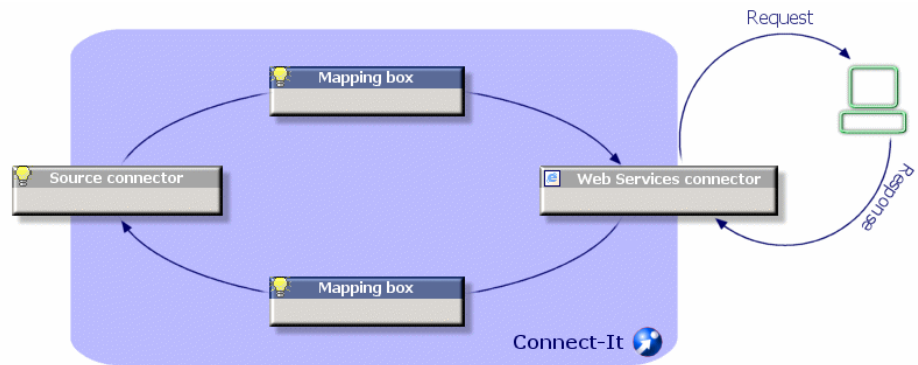
Example of using the Web Service connector

- 1 An external application produces a document containing a zip code. A mapping enables the Web Service connector to consume this document.
- 2 The connector sends a request by sending this zip code as a parameter to a Web service that provides temperatures.
- 3 The Web service sends a response containing the temperature corresponding to the zip code.

In response to this, the Web Service connector spontaneously produces a document in Connect-It.

- 4 The produced document is used in a feedback loop that sends information to the external application that initially provided the zip code.

Figure 5.6. Web Service connector - general principles



 Note:

Certain Web services do not function on the query/response model and cannot provide a response.

Limitations - Web Service connector

The Web Service connector does not support:

- Operations with identical names.
The Web Service connector only supports operations with unique names in the same Web service. If a Web service references two operations with identical names, the connector will only publish the first one.
- Queries with multiple responses.
The Web Service connector associates a response (produced document type) to every request (consumed document type). If a Web service associates more than one response to a request, the connector only publishes the first operation.
- Responses with multiple requests.
If a Web service associates more than one request to a response, the connector only publishes the first operation.

Table 5.13. Supported protocols - Web Service connector






Protocol	Supported
SOAP by HTTP protocol	Yes
HTTP GET	Yes
HTTP POST	Yes
MIME	No

Configure the Web Service connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 5.14. Configure the Web Service connector

 Name and describe the connector [page 17]
 Specify the WSDL [page 272]
 SSL configuration (advanced mode) [page 273]
 Parallelize consumption (advanced mode) [page 44]
 Configure the JVM [page 273]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].


Specify the WSDL

This page enables you to enter the connector's connection parameters to a Web service.

- **WDSL URL address** field: The WDSL URL address enables you to recover for each Web service:
 - the communication protocols used
 - The supported operations.
 - The format of these operations
 - The data consumed and the data produced.
- **Login** field: User identifier

- **Password field**

Example: <http://www.xmethods.net/sd/2001/TemperatureService.wsdl>

When you click the  icon, you can access an UDDI connector (Universal Description Discovery and Integration).

Specify the protocol

Select from the list of protocols displayed the transport protocol to use.

By default, all the protocols are used and operations are then prefixed according to the protocol.

SSL configuration (advanced mode)

This page allows you to define:

- the Java certificates file for the secure connection (**CA certificates file**). This file is required if the Web browser cannot manage the Java certificate.
- the private key file (**Certificates and client keys file (keystore)**). This file is required if the private key is stored in a database.
- the password for the private key file (**Keystore password**).

By default, Connect-It searches for information concerning the Java certificate file in the JRE, in the following files: `/lib/security/jsse cacerts` and `/lib/security/cacerts`.

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Parallelize consumption \(advanced mode\)](#) [page 44].

Configure the JVM

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Consumption directives - Web Service connector

For information about how to enter a connector's consumption directives, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51].

Operations to apply

Select the **query** operation so that the connector can send data to the Web service's server.

If you select **none** in the **Operation to apply** field, the **query** operation will still be used. In this case, a warning appears in the Document log indicating that this operation has been performed.

Published document-types

The Web Service connector publishes document types in production corresponding to what is defined by WSDL.

The document types available in consumption mode correspond to the Web service operations.

For each operation there is a document type available for consumption (sending a request) and for production (receiving the response).

Naming the published document types



Note:

The Web Service connector does not publish operations that use an unsupported protocol. Example: the MIME protocol.

The names you assign to document types published by the Web Service connector must have a prefix corresponding to the communication protocol and name of the operation concerned.

Consumed document types

Table 5.15. Web Service connector - Consumed document types

Protocol	Consumed document types
SOAP Messaging	Soap-[Name of the operation]
HTTP GET	HttpGet-[Name of the operation]
HTTP POST	HttpPost-[Name of the operation]

Out-of-the-box scenarios - Web Service connector

The following out-of-the-box scenarios use this connector:

- ◆ [ws/wsac43/currency.scn](#) [page 425]

XML connector

Optional connector

- Production (source)
- Consumption (destination)

The XML enables the:

- Processing of XML files located on the computer or network on which Connect-It is installed
- Processing of XML files located on FTP or Web sites.

 **Note:**

The XML files processed can be compressed (gzip).

Known limitations of the XML connector

In the document types processed by the XML connector, the following field types are not supported:

- any
- PCDATA (blob)
- CDATA
- Namespaces

















Configure the XML connector (read)

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

Certain options concerning this connector can be enabled using the **Edit/Options/ Connector** menu. For more information, refer to the **User's Guide**, chapter **Scenario builder**, section **Connector**.

The following table presents all the pages in the connector's configuration wizard.

Table 5.16. Configure the XML connector (read)

 Name and describe the connector [page 17]		
 Choosing a processing mode [page 277]		
 Selecting a connection protocol [page 277]		
 Connecting to an HTTP Web site [page 277]		
 Define the connection parameters of the HTTP server [page 288]	 FTP server connection [page 279]	 Local or network file(s) [page 280]
 Choose a DTD/XSD [page 281]	 Select files or folders [page 27]	 Select files or folders [page 280]
	 Define post-processing actions [page 280]	 Define post-processing actions [page 280]
	 Choose a DTD/XSD [page 281]	 Choose a DTD/XSD [page 281]
	 Configure the JVM [page 286]	 Configure the JVM [page 286]

Warning:

Path of a file or folder

Often when configuring a connector, you must indicate the path of a file or folder in the configuration wizard. If this file or folder is in a folder associated with a network drive on your computer, you must never indicate the letter of this drive; only indicate the folder associated with this letter.

Using a network drive in a path will prevent the scenario from working properly when it is associated with a service. For more information, refer to the **User's guide**, chapter **Document type mappings**, section **Edit the mappings**, sub-section **Path of a document's elements**.

Name and describe the connector

To use this page, refer to chapter **Configuring connectors** [page 15], section **Name and describe the connector** [page 17].

Choosing a processing mode

This page enables you to indicate if the connector is used in read or write mode.

Selecting a connection protocol

The three available options are:

- HTTP Web site
- FTP server
- Local or network file(s)

Connecting to an HTTP Web site

If you have chosen to read the XML files on a Web site, you must indicate the HTTP connection parameters:

- Address
Enter an address of the type: **[protocol] ://[address] : [port]/[path]**.
The [address] part is sometimes the only mandatory part.
The usual port number for an HTTP server is 80.
- Login
Enter the login that enables you to access the chosen site.
- Password
Enter the password corresponding to your login.

Secured connection (HTTPS protocol)

This option enables you to indicate if you want to connect to your site via a secured connection (HTTPS).

Important:

If the [protocol] part of the Web site's address does not corresponds to the HTTPS protocol, the selection of the **Secured connection (HTTPS protocol)** option forces the use of an HTTPS protocol.

The usual port number for an HTTPS server is 443.

Example of an HTTP address

Address	Secured connection (HTTPS protocol)	Connection mode
https://a.b.c.d/	Selected	https port 443
https://a.b.c.d:9000/	Selected	https port 9000
http://a.b.c.d:9000/	Not selected	https port 9000
http://a.b.c.d:9000/	Selected	https port 9000
https://a.b.c.d:9000/	Not selected	https port 9000
http://a.b.c.d:443/	Not selected	https port 443
https://a.b.c.d:80/	Selected	https port 80

Client certificate

This field enables you to select an HTTPS certificate from among those present on your computer.

Important: If you change or delete a client certificate specified in this field after the connector has already read or written to these documents, you must close and reopen the Scenario builder in order for this modification to be accounted for.

Manage the list of client certificates

The list of client certificates proposed by the configuration wizard corresponds to the list of certificates in your **Root console/ Certificates - current user/ Personal/ Certificates** folder of the Microsoft Management application.

To add certificates to this folder using Windows XP:

- 1 Select **Run** from the Windows **Start** menu.
- 2 Enter **MMX** in the **Open** field.
- 3 Select **File/ Add/Remove Snap-in**.
- 4 Click **Add** in the window that is displayed.
- 5 Select **Certificates** in the window that is displayed.
- 6 Click **Add**.
- 7 Select **My user account** in the dialog box that is displayed.
- 8 Click **Finish**.
- 9 Click **OK**.
- 10 Add or delete the files in the **Certificates - Current user/ Personal/ Certificates folder**.

Do not verify the server identity

Use this option to support a proxy server in a secure connection.

If a proxy server is installed, the address specified in the server certificate does not match the address specified for the proxy server. Select this option to disable the automatic verification of the server identity.

FTP protocol

If you have chosen to read the XML files on an FTP site, you must:

- 1 Select the FTP connection parameters
 - 2 Select an action to perform after the files have been processed.
-

 Note:

Connection in passive FTP mode is supported.

FTP server connection

On this page, you must populate three fields, which enable you to connect to your FTP server:

- Server
Enter the name of your FTP server
- Login
Enter the login that enables you to access the chosen site.
- Password
Enter the password corresponding to your login.
- Select the option **Connect in passive mode** if your connection is blocked by a firewall and you wish to open an extra port.

You can test the validity of your connection using the **Test** button.

Select files or folders

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select files or folders](#) [page 27] .

Define post-processing actions

After a file is read by the XML connector, Connect-It gives you three options:

- Leave it in the folder
- Delete it from the folder

- Move it to another folder

If you choose this last option, you must indicate the path of the folder to which you want to move the processed file.

You must specify one of these options in the case of failure, and one in the case of success, in processing the XML files by your connector.

In order to apply a post-processing action to the documents consumed by the other connectors and mapping boxes of your scenario, you must use the processing reports that each one produces. For more information about processing reports, refer to the **User's Guide, Processing report**.

For further information on the mappings to be made to take post-processing actions into account, refer to the section [Using post-processing actions](#) [page 280] of this chapter.

Local or network file(s)

If you choose to read local or network files, you must:

- 1 Choose a file or folder
- 2 Select an action to perform after the files have been processed.

Select files or folders

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select files or folders](#) [page 27].

Define post-processing actions

After a file is read by the XML connector, Connect-It gives you three options:

- Leave it in the folder
- Delete it from the folder
- Move it to the folder.

If you choose this last option, you must indicate the path of the folder to which you want to move the processed file.

You must specify one of these options in the case of failure, and one in the case of success, in processing the XML files by your connector.

Using post-processing actions

To use post-processing actions, you must:

- ◆ Create a mapping between the **URLFileInfo.Path** structure of the document type produced by the source connector and the **URLFileInfo.Path** element of the **SuccessReport** document type.


For further information on this structure, refer to section [Published document types](#) [page 299] of this chapter.

Choose a DTD/XSD

To process an XML file, the connector must use a DTD (Document Type Definition) or an XSD (Extended Schema Definition). This page enables you to specify the DTD or the XSD used in the **DTD/XSD** field.

Note:

If you use the XML connector in Learning mode (see below), the name that you indicate in the **DTD** field corresponds to a file in which the contents will be created once the learning process is finished. If the name indicated corresponds to an already existing file, Connect-It automatically saves this file and assigns it a number (Example: **request_01.dtd**). If this connector is relaunched several times in Learning mode, the different DTDs created will all have a number corresponding to their order of creation. (Example: **request_01.dtd**, **request_02.dtd**, **request_03.dtd**, etc. The **request_03.dtd** field corresponds to the DTD file that was most recently saved.)

By clicking , the **File location** page is displayed. This page enables you to enter the path of your DTD or XSD depending on your connection protocol.

Warning:

A partial validation of the document type is performed using the specified DTD. The following elements are validated: elements not declared in the DTD, the integrity of hierarchical (parent-child) relations.

File location

You have the choice of three options:

- **Local/Network**
Enter the full path of your DTD or XSD file located on the computer or network on which Connect-It is installed.
- **FTP**
Enter the FTP parameters in the **Server**, **Login**, **Password**, and **Path** fields. The **Path** field enables you to specify the location of your DTD or XSD file on the server. Example: If your DTD is located in the **myfolder/dtd** folder on

the **http://mycompany.com** server, you must populate the **Server** and **Path** fields with **http://mycompany.com** and **/myfolder/dtd/mydtd.dtd**.

Attention: You cannot write the DTD in learning mode on an FTP site.

 **Note:**

Do not include a slash mark (/) at the end of the server name; instead include it at the start of the path of the DTD.

- **HTTP**

Enter the HTTP parameters in the **Server**, **Login**, **Password**, and **Path** fields. The **Path** field enables you to specify the location of your DTD or XSD file on the server. Example: If your DTD is located in the **myfo1der/DTD** folder on the **http://mycompany.com** server, you must populate the **Server** and **Path** fields with **http://mycompany.com** and **/myfolder/DTD/mydtd.dtd**.

Attention: You cannot write the DTD in learning mode on an HTTP site.

 **Note:**

Do not include a slash mark (/) at the end of the server name; instead include it at the start of the path of the DTD.

Determine the root elements

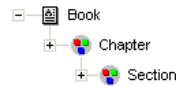
A DTD or XSD is composed of elements that can contain other elements. (For example: In a DTD intended for book publishing, **Section** elements are included in the **Chapter** elements, which are in turn included in the **Book** element.) The only element that cannot be included in another element is a root element. A DTD or XSD can contain any number of root elements or none at all.

To determine the root elements, which correspond to as many document types as published by the XML connector, Connect-It gives you different options:

Publish a document type for each root element found in the DTD/XSD (recommended)

If you choose this option, the XML connector searches all the root elements of the DTD or XSD and publishes a document type for each root element found.

Example: In a DTD intended for book publishing, only one root element (**Book**) is found. In this case, the XML connector will only publish one document type.



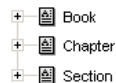
Note:

When no root elements can be found (all the elements that can include other elements), Connect-It will take the first element found in the DTD or XSD as its only root element. Example: A DTD corresponds to the organization of a relational database in which all the elements (each one representing a table) include one another: The Assets table is linked to the Users table, which is linked to the Assets table, etc.

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

If you select this option, you can indicate the root elements of your choice, separating each one with a comma.

Example: In a DTD intended for book publishing, the user decides to choose the elements **Book**, **Chapter**, and **Section**. In this case, the XML connector will publish one document type per root element chosen by the user.



Note:

For a DTD corresponding to the organization of a relational database in which all the elements (each representing a table) include one another, this option enables you to obtain a published document-type for each one of these tables.

Publish a document type for each first-level collection

By choosing this option, the XML connector publishes as many document types as there are first-level collections declared in the selected DTD or XSD.

Example :

A DTD, whose root element is **Company**, contains three first-level collections: **Employees**, **Suppliers**, and **Locations**. If you do not select this option, the connector will publish only one document type having **Company** as its root element.

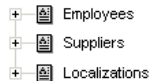
 Note:

The first-level collections are those that appear directly under the root element of a document type published by a connector.



For example, there will be one single "Company" XML document produced containing all employees, suppliers and locations in your database.

Using the **Publish a document type for each first-level collection** option, the XML connector publishes one document type per first-level collection. In this case, the root element of the DTD (the **Company** element) no longer appears in the tab of the document types published by the XML connector.



For example, there will be as many XML documents produced as there are employees, suppliers and locations in your database.

Learning mode (no documents produced)

The learning mode enables the XML connector to create a DTD according to the XML file or files specified on the **Select a file or folder** page.

 Note:

The DTD obtained in learning mode must be saved in a local or network file. In this case, it is not possible to save it to an HTTP or FTP site.

To start learning mode:


- Configure the XML connector by selecting the **Learning mode (no documents produced)** option.
- Select your XML connector in your scenario's diagram.
- Select **Produce now** in the **Tools** menu.
- or**
- Right-click and choose **Produce** from the menu that appears.
- or**
- Press **F5**.

To disable learning mode, reconfigure the XML connector by clearing this option on the **Choose a DTD** page.

In case of conflict between XML and DTD/XSD documents

In principle, the XML elements of the files read by the XML connector must match those defined in your DTD. Nevertheless, it is always possible that new elements appear. This type of conflict is frequently encountered when the DTD used is obtained in learning mode using a limited number of XML files containing a small number of elements.

To handle these conflicts, Connect-It provides two different options:

- **Reject file and continue**
In case of conflicts, the read XML files containing non-defined elements in the DTD or XSD are rejected. (Whether these documents are saved or not depends on the option selected on the **Action after processing** page.) In order that these files be properly processed, you must restart the XML connector in learning mode.
- **Issue warning and continue**
In the case of a conflict, the XML files are processed normally. A warning in the document log, recognizable by the  icon, appears in the tracking lines of the XML connector in read mode.

The types of conflicts handled by the application are:

- element names
- attribute names
- ? and * wildcard characters for occurrences

Ignore the XML files whose root element does not correspond to the current document type (do not apply post-processing actions)

This option enables you to process XML documents corresponding to different document types, and therefore linked to different DTDs.

Selecting this option enables you to take current DTD into account alone, and as a consequence, avoid processing errors for XML documents linked to other document types.

For example, if you have specified a folder in which there are several XML documents, select this option to only take into account the DTD defined in the **DTD/XSD** field.

Configure the JVM

This page is only visible if you selected an XSD in the [Choose a DTD/XSD](#) [page 94] page of the configuration wizard.

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Options associated with the XML connector

You can access the options of the XML connector using the menu **Edit/ Options/ Connector/ Delimited text and XML**.

The following options are associated with the XML connector in FTP protocol:

- Display the URL being processed in the Connect-It log
- Copy locally the files to read from FTP server.

Select this option to make a local copy of the files on an FTP server and to read the data from the local file.














This option must be selected if the network configuration does not enable a connection to be maintained on the FTP server long enough for a file to be processed.

Configure the XML connector (write)

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 5.17. Configure the XML connector (write)

 Name and describe the connector [page 17]		
 Select a processing mode [page 287]		
 Selecting a connection protocol [page 287]		
 Define the connection parameters of the HTTP server [page 288]	 FTP connection [page 290]	 Select files or folders [page 288]
 Choose a DTD/XSD [page 292]	 Select files or folders [page 288]	 Behavior between two sessions [page 291]
	 Behavior between two sessions [page 291]	 Choose a DTD/XSD [page 292]
	 Configure the JVM [page 295]	 Configure the JVM [page 295]

 **Warning:**

Path of a file or folder

Often when configuring a connector, you must indicate the path of a file or folder in the configuration wizard. If this file or folder is in a folder associated with a network drive on your computer, you must never indicate the letter of this drive; only indicate the folder associated with this letter.

Indicating the network drive in the path stops the scenario from working correctly when it is associated with a service. For more information about this, consult the **User's guide**, section **Edit the mappings**, sub-section **Path of a document's elements**.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors \[page 15\]](#), section [Name and describe the connector \[page 17\]](#).

Select a processing mode

This page enables you to indicate if the connector is used in read or write mode.

Selecting a connection protocol

The available options are:

- FTP
- Local or network file(s)
- HTTP

Define the connection parameters of the HTTP server

On this page, you must populate three fields enabling you to connect to the HTTP server on which you want to write files.

- Address
This field is mandatory. Enter an address of the type:
[protocol]://[address]:[port]/[path].
- Login
Enter the login that enables you to access the chosen site.
- Password
Enter the password corresponding to your login.

Secured connection (HTTPS protocol)

To use this option, refer to the section [Secured connection \(HTTPS protocol\)](#) [page 277].

Write command

There are two write commands used to write to an HTTP server:

- POST
Sends data to the program located at the indicated address. It is different from the PUT method in that the sent data must be processed. **Example:** sending data from a CGI form.
- PUT
Sends data so that it can be saved to the indicated address. **Example:** updating pages of a Web site.

Select the command used by your HTTP server.

Local or network file(s)

If you have chosen to write local or network files, you must select a file or folder.

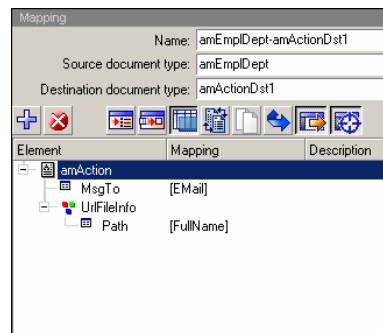
Select files or folders

This page enables you to select the XML file or files that your connector must read.

Before defining the connector's behavior, specify the folder for writing files (**Folder name** field). The path for this folder is used in a mapping in combination with the **UrFileInfo.Path** element.

You can choose between the following behaviors:

- Write documents to one single file
Enter the path of a folder on your computer or your network (**Folder name** field). The document will be overwritten or left alone depending on the parameters defined in the next page of the wizard.
- Write to a different file for each document
Specify the folder in which the files will be written (**Folder name** field). The name of these files will correspond to the generic name that you have entered in the consumption directives of the XML connector. Refer to the [Consumption directive of the XML connector \(write\) - FTP mode](#) [page 296] sub-section of this section.
You can also define the names of these files by mapping the element **UrlFileInfo.Path**. For example, to create a file with the name of each employee in AssetCenter. In this case, the path defined in the wizard page is concatenated with the path defined in the mapping.



In order for the path defined in the mapping to be used as the default path, and as a result to avoid using the path defined in the wizard, use the '/' character in the mapping.

 **Note:**

This behavior does not prevent you from specifying a valid path in the connector configuration wizard.

If the path specified for the **UrlFileInfo.Path** element does not exist, it is created automatically provided that:

- the path uses part of the path specified in the configuration wizard.

- the path does not add more than one level to the specified tree. For example, for a path **a/b/c** specified in the configuration wizard, only a path **a/b/c/d/filename**, specified in the mapping, is possible.

FTP protocol

If you have chosen to write files on an FTP site, you must:

- 1 Select the FTP connection parameters.
- 2 Select an action to perform after the files have been processed.

FTP connection

On this page, you must populate three fields, which enable you to connect to your FTP server:

- Server
Enter the name of your FTP server
- Login
Enter the login that enables you to access the chosen site.
The login is case-sensitive.
- Password
Enter the password corresponding to your login.

You can test the validity of your connection using the **Test** button.

Select files or folders

Before defining a connector's behavior, specify:

- ♦ the write files folder (**Path** field). The specified path is used in mapping in combination with the **UrlFileInfo.Path** element.

Select the option **Write the documents to a single folder** or **Write to a different file for each document** according to your needs.

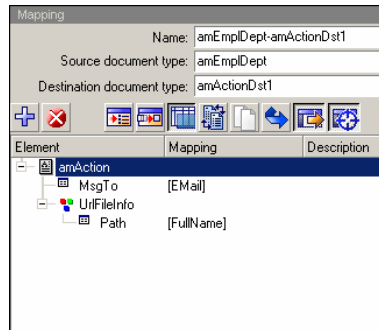
Write documents to one single folder

Enter the name and the path of the file (**Path** field) that you want to create with the help of the XML connector. If this file already exists, its contents will be overwritten.

Write to a different file for each document

Indicate the folder (**Path** field) in which the files will be written. The names of these files will correspond to the generic name that you have entered in the

consumption directives of the XML connector. Refer to the [Consumption directive of the XML connector \(write\) - FTP mode](#) [page 296] sub-section of this section. You can also define the names of these files by mapping the element **UrlFileInfo.Path**. For example, to create a file with the name of each employee in AssetCenter. In this case, the path defined in the wizard page is concatenated with the path defined in the mapping.



In order for the path defined in the mapping to be used as the default path, and as a result to avoid using the path defined in the wizard, use the '/' character in the mapping.

Note:

This behavior does not prevent you from specifying a valid path in the connector configuration wizard.

If the path specified for the **UrlFileInfo.Path** element does not exist, it is created automatically provided that:

- the path uses part of the path specified in the configuration wizard.
- the path does not add more than one level to the specified tree. For example, for a path **a/b/c** specified in the configuration wizard, only a path **a/b/c/d/filename**, specified in the mapping, is possible.

Behavior between two sessions

This page enables you to select how documents are processed between two data-writing sessions. The behavioral options are different if you write data to one single file or if you write it to several files in a folder.

Write the documents to a single file

To write to a single file, you have the following options:

- **Append to the same file**
If you choose this option, the connector starts writing data to the open file again the moment the previous session is interrupted.
- **Overwrite the last file**
The connector deletes the file in which it wrote data during the previous session.
- **Number the different files**
The connector creates a file whose name is the same as the previous file, but incremented by 1 unit. Example: **file.xml**, **file1.xml**, **file2.xml**, etc.


Write to a different file for each document

To write in several files, you have the following options:

- **Delete all previous files and start numbering them from the beginning**
The files in the folder are all deleted (all the files dating from all the previous session; not just from the last one).
The connector starts incrementing all the files it writes.
- **Continue numbering files**
The connector starts writing files again and incrementing them without deleting the existing files.

Choose a DTD/XSD

To create an XML file, the connector must use a DTD (Document Type Definition) or XSD (Extended Schema Definition). On the **Select a DTD/XSD** page, indicate the DTD or XSD used in the **DTD/XSD** field. The document types published by the XML connector correspond to this DTD or XSD.

By clicking , the **File location** page is displayed. This page enables you to enter the path of your DTD or XSD depending on your connection protocol.

File location

You have the choice of three options:

- **Local/Network**
Enter the full path of your DTD or XSD file located on the computer or network on which Connect-It is installed.
- **FTP**

Enter the FTP parameters in the **Server**, **Login**, **Password**, and **Path** fields. The **Path** field enables you to specify the location of your DTD or XSD file on the server. Example: If your DTD is located in the **myfolder/dtd** folder on the **http://mycompany.com** server, you must populate the **Server** and **Path** fields with **http://mycompany.com** and **/myfolder/dtd/mydtd.dtd**.

 **Note:**

Do not include a slash mark (/) at the end of the server name; instead include it at the start of the path of the DTD or XSD.

■ HTTP

Enter the HTTP parameters in the **Server**, **Login**, **Password**, and **Path** fields. The **Path** field enables you to specify the location of your DTD or XSD file on the server. Example: If your DTD is located in the **myfolder/dtd** folder on the **http://mycompany.com** server, you must populate the **Server** and **Path** fields with **http://mycompany.com** and **/myfolder/dtd/mydtd.dtd**.

 **Note:**

Do not include a slash mark (/) at the end of the server name; instead include it at the start of the path of the DTD or XSD.

Determine the root elements

A DTD or XSD is composed of elements that can contain other elements. (For example: In a DTD intended for book publishing, **Section** elements are included in the **Chapter** elements, which are in turn included in the **Book** element.) The only element that cannot be included in another element is a root element. A DTD can contain any number of root elements.

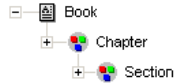
To determine the root elements, which correspond to as many document types as published by the XML connector, Connect-It gives you different options:

- Publish a document type for each root element found in the DTD/XSD (recommended)
- Publish a document type for each root element selected by the user (values are separated by commas)

Publish a document type for each root element found in the DTD/XSD (recommended)

If you choose this option, the XML connector searches all the root elements of the DTD or XSD and publishes a document type for each root element found.

Example: In a DTD intended for book publishing, only one root element (**Book**) is found. In this case, the XML connector will only publish one document type.



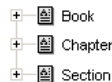
Note:

When no root elements can be found (all the elements that can include other elements), Connect-It will take the first element found in the DTD or XSD as its only root element. Example: A DTD corresponds to the organization of a relational database in which all the elements (each one representing a table) include one another: The Assets table is linked to the Users table, which is linked to the Assets table, etc.

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

If you select this option, you can indicate the root elements of your choice, separating each one with a comma.

Example: In a DTD intended for book publishing, the user decides to choose the elements **Book**, **Chapter**, and **Section**. In this case, the XML connector will publish one document type per root element chosen by the user.



 **Note:**

For a DTD corresponding to the organization of a relational database in which all the elements (each representing a table) include one another, this option enables you to obtain a published document-type for each one of these tables.

Do not reference the resulting DTD in the XML file

This option is selected by default.

Write resulting DTD/XSD inside the XML file

By selecting this option, the XML files created by the XML connector contain a DTD or XSD in their header.

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

If you choose this option, you must specify in the available field the location to which you want to save the DTD or XSD of the created XML documents. In this case, the DTD or XSD must be saved to a local or network file.

 **Warning:**

Starting a scenario will open several sessions: a session corresponding to each source connector started in your scenario by a scheduler (Example: The source connector produces documents every week.)

A DTD or XSD is only written in the external file the first time that you use the scenario. This means that for all the other sessions of that scenario, the DTD or XSD in the external files always corresponds to the one obtained in the first session.

In order for a new DTD or XSD to be written to the external file, you must stop the scenario and then restart it.

Configure the JVM

This page is only visible if you selected an XSD in the [Choose a DTD/XSD \[page 94\]](#) page of the configuration wizard.

To use this page, refer to chapter [Configuring connectors \[page 15\]](#), section [Configure the JVM \[page 43\]](#).

Consumption directive of the XML connector (write) - FTP mode

The consumption directives indicate for each document type consumed by the XML connector:

- A generic name
By default, the value of this field is the name of the document type consumed by the XML connector.
- A file name extension
Enter this name without the period mark. Example: Enter **xml** instead of **.xml**; "**xml**" is the default value.

These two fields will only be used if you have chosen the **Write to a different file for each document** option during the configuration of your XML connector. (See the sub-section [Configure the XML connector \(write\)](#) [page 286] of this section.)

When you run your scenario, the files that are written will have a composite name, which is created from the generic name, a number corresponding to the file's creation order (_01, _02, _03, etc.) and from the file-name extension indicated.

Example: You enter the values **ebizz** and **xml**. The files created have the following names: **ebizz_01.xml**, **ebizz_02.xml**, **ebizz_03.xml**, etc.

Warning:

If your XML connector consumes several document types, do not use the same generic name twice: The most recently created files will overwrite the previously created files having the same generic name.

Figure 5.7. XML connector - Consumption directives

Generic name	janAssetDat
Extension file name	xml

Additional information - XML connector

This section presents additional information about the connector.

Writing to one or more files

The XML connector in write mode enables you to write data to one or more files.

Write the documents to a single file

This mode enables you to save each document consumed by the connector in one single file.

You can, for example, use this mode in a single file so that a single XML file contains an inventory of assets recorded in AssetCenter. You can use the following DTD:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT AssetInventory (amAsset*) >
<!ELEMENT amAsset (AssetTag?) >
<!ELEMENT AssetTag (#PCDATA) >
```

In this DTD, the root element (AssetInventory) has the amAsset collection as its sub-element. This collection's members correspond to asset records in the AssetCenter database. Each asset is identified by its asset tag.

In this case, the XML connector writes a file. The following example shows the file's contents:

```
<?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>
```



Mapping between source and destination

The mapping between the **amAsset** and the DTD used by the XML connector presents the following aspects:

- 1 The asset tag field is directly mapped to the AssetTag of the DTD.
- 2 Only the child nodes (amAsset) of the root element (AssetInventory) are displayed in the destination document type.

With the following DTD:

```
<?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>
```

The amSoftware element and the ID field will be ignored and will not be displayed in the document type consumed by the XML connector. It is as though they are not declared in the DTD.

- 3 The **Publish a document type for each first-level collection** option must not be selected.

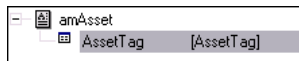
Write to a different file for each document

This mode enables you to save each document consumed by the XML connector in different folders.

In the previous example, each asset record gives rise to the creation of a file by the XML connector.

You can use the following DTD:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT amAsset (AssetTag?)>
<!ELEMENT AssetTag (#PCDATA)>
```



In this case, the XML connector writes several files. The following examples show their contents.

File 1:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<amAsset>
<AssetTag>DEMO-OFF1</AssetTag>
</amAsset>
```

File 1:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<amAsset>
<AssetTag>DEMO-NTR1</AssetTag>
</amAsset>
```

Mapping between source and destination

The mapping between the **amAsset** and the DTD used by the XML connector presents the following aspects:

- 1 All the elements declared in the DTD are displayed in the destination: collections, structures, fields.
- 2 The **Publish a document type for each first-level collection** option must not be selected.

Transformation rules for end-of-line markers

Depending on whether the XML connector is used in read or write mode, the rules for the end-of-line markers (CRLF or LF) differs.

Utilization in read mode

By default

- In Windows, all end-of-line characters are transformed to CRLF (Carriage Return and Line Feed).
- In Unix, all end-of-line characters are transformed to LF (Line Feed).

This behavior can be modified using a script, as shown in the example below:

```
RetVal = Replace([Record.cc.dd.ee], Chr(13) & Chr(10), Chr(10), 0)
```

Utilization in write mode

By default:

- In Windows, all end-of-line characters are transformed to CRLF (Carriage Return and Line Feed).
- In Unix, all end-of-line characters are transformed to LF (Line Feed).

This behavior may not be modified using a script.

Published document types

The XML connector publishes the following structure for all document types:

- ◆ **UrlFileInfo**

This structure is mainly used for post-processing actions.

The **UrlFileInfo** structure contains the following fields:

- **creationdate**

This field corresponds to the creation date of the document

- **lastmodificationdate**
This field corresponds to the last modification date of the document
 - **path**
This field corresponds to the path of the file
-

 **Note:**

If you wish to use multiple XML file types, you must deploy as many connectors as there are different files. Since each connector has a unique behavior (production or consumption), you must create as many connectors as there are behaviors.



6 Inventory connectors

CHAPTER

Inventory connectors enable you to process data from applications allowing you to inventory computer-based repositories.

Altiris connector

Optional connector

◆ Production (source)

This section describes the Altiris connector.

The Altiris connector enables you to process an Altiris for Inventory solution database.











An Altiris database contains data related to inventoried assets (desktop and portable computers, servers, etc.) and their dependencies (printers).

Configure the Altiris connector

You can configure the connector using a wizard. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.1. Configure the Altiris connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

- ◆ **Executing an initial import:** This option enables you to avoid SELECT queries in reconciliation scripts for initial imports, for example, when the database is empty or the items to be inserted do not yet exist. This shortens the processing time.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Additional information

Unique reconciliation key

- ◆ The **Guid** field of the **vComputer** view is used as the unique identifier for the mapping to an AssetCenter table.

Schedule pointers

The **Collection Time** field in the **AeXInv_AeX_AC_Inventory_Results** view can be used as the schedule pointer.

Only computers entered into the inventory table are taken into account. Computers that have not been entered into the inventory table must be scheduled separately.

Out-of-the-box scenarios - Altiris connector

The following out-of-the-box scenarios use this connector:

- ◆ Altiris scenarios [page 376]

Asset Insight connector

Optional connector

◆ Production (source)

The Asset Insight 4.0 gateway connector enables you to process Asset Insight databases.

Asset Insight databases contain information relating to a company's IT equipment. For each computer registered in an Asset Insight database, the scenarios enable you to create records in AssetCenter.

Known limitations of the Asset Insight connector

We highly recommend using the Asset Insight connector with an ODBC driver and ODBC Administrator using identical version levels of DLL files.

Example :

Your Asset Insight connector will only work if you use ODBC Administrator version 3.0 and a Microsoft Access 4.00 ODBC driver.

Mandatory initialization of the status pointer

In a scenario involving an Asset Insight connector, you must assign its pointer the value '0' in the scheduler's edit window if it's a text field. If it's a date field, you must assign it an old enough date so that it will work. By default, an Asset Insight connector can assign a value to the pointer before the first launch of a scenario in scheduled mode.

 **Warning:**

Without having completed this step, the Asset Insight connector will not produce any documents.

To assign this value to the pointer:

- 1 Select **Edit/Scheduling**











- 2 Associate the documents produced by the Asset Insight connector to the previously created schedulers.
- 3 For every produced document appearing in the list, double-click the corresponding cell in the Pointer-status column so that a text box appears.
- 4 Enter '0' in the text box.

Configure the Asset Insight connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.2. Configure the Asset Insight connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Production directives of the Asset Insight connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information

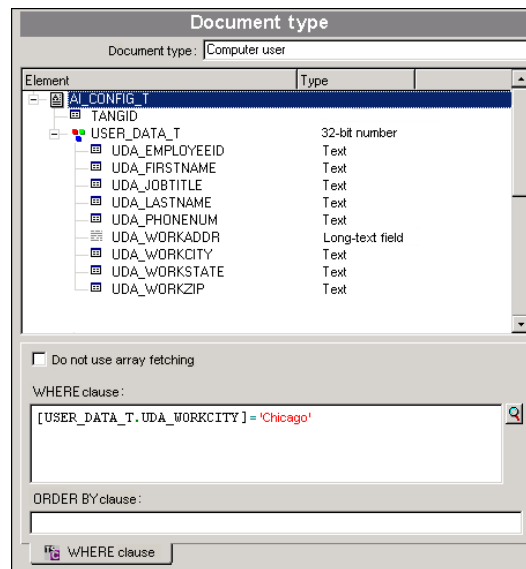
about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Example of a WHERE clause for the Asset Insight connector

To obtain all the employees working in Chicago, you must enter the following function in the WHERE clause field:

```
[USER_DATA_T.UDA_WORKCITY]='Chicago'
```

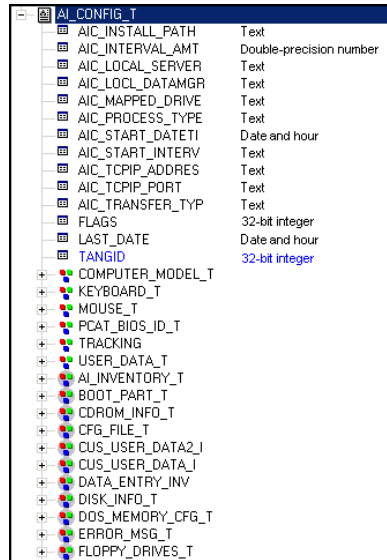
Figure 6.1. Asset Insight connector - Production directive example



Published document-type

The Asset Insight connector only publishes one document type: **AI_CONFIG_T**. This document type corresponds to the Asset Insight scan performed on a computer. The elements in this document type depend on the scanning configurations. For example: Collections representing tables in a database might appear in some scans while not in others.

Figure 6.2. Asset Insight connector - AI_CONFIG_T document type



Field Name	Data Type
AIC_INSTALL_PATH	Text
AIC_INTERVAL_AMT	Double-precision number
AIC_LOCAL_SERVER	Text
AIC_LOCL_DATAMGR	Text
AIC_MAPPED_DRIVE	Text
AIC_PROCESS_TYPE	Text
AIC_START_DATETI	Date and hour
AIC_START_INTERV	Text
AIC_TCPIP_ADDRES	Text
AIC_TCPIP_PORT	Text
AIC_TRANSFER_TYP	Text
FLAGS	32-bit integer
LAST_DATE	Date and hour
TANGID	32-bit integer
COMPUTER_MODEL_T	
KEYBOARD_T	
MOUSE_T	
PCAT_BIOS_ID_T	
TRACKING	
USER_DATA_T	
AI_INVENTORY_T	
BOOT_PART_T	
CDROM_INFO_T	
CFG_FILE_T	
CUS_USER_DATA2_I	
CUS_USER_DATA_I	
DATA_ENTRY_INV	
DISK_INFO_T	
DOS_MEMORY_CFG_T	
ERROR_MSG_T	
FLOPPY_DRIVES_T	

Warning:

The elements in this document type depend on your Asset Insight database structure. Some of the tables represented by collections in an AI_CONFIG_T document type might not even be present in the database, and if the scenario mapping tries to reference the fields from these absent tables, this scenario will not run properly. In order to solve this problem, you need to delete the referenced fields:

- 1 In the scenario mappings.
- 2 In the document types produced by the Asset Insight connector.

Out-of-the-box scenarios - Asset Insight connector

The following out-of-the-box scenarios use this connector:

- ◆ [Asset Insight scenarios](#) [page 378]

LANDesk for Inventory 8 Connectors

Optional connector

◆ Production (source)

The LANDesk inventory connectors enable you to process data from databases retrieved using the LANDesk inventory tool.

This inventory tool gathers extensive information about a computer population.

Known limitations

We highly recommend using the LANDesk connector with an ODBC driver and ODBC Administrator that use the same DLL versions.

Example :











The LANDesk connector will not work if you use ODBC Administrator version 3.0 and a Microsoft Access 4.00 ODBC driver.

Configure the LANDesk connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.3. Configure the LANDesk for Inventory 8 Connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]

Prerequisites

Before configuring your connector, make sure a valid connection to your **ODBC** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and password.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Production directives of the LANDesk connector

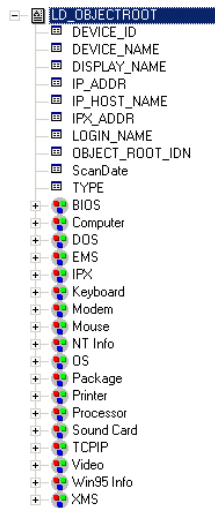
For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Published document-type

The LANDesk connector publishes a single document type **LD_OBJECTROOT**. This document type corresponds to the scan that LANDesk has performed on a computer.

Figure 6.3. LANDesk for Inventory 8 connector - COMPUTER_SYSTEM document type



Out-of-the-box scenarios - LANDesk for Inventory 8 connector

The following out-of-the-box scenarios use this connector:

- ◆ [LANDesk scenarios](#) [page 386]

LANDesk Software Distribution connector

Additional connector

- 1 Consumption (destination)
- 2 Consumption - spontaneous production (request - response)

The LANDesk Software Distribution connector enables interaction with the LANDesk Web service.

In an integration scenario, the LANDesk Software Distribution connector consumes a document and sends it as a query to a Web service, then receives a reply which it automatically transforms into a produced document.

- ▶ [AssetCenter Software Distribution Guide](#).

Limitations of the LANDesk Software Distribution connector

Only the NTLM authentication protocol is used.

You must use LANDesk Management Suite version 8.6 or later and the MBSDK must be installed on the LANDesk server.

Prerequisites

- A LANDesk account with administrative rights must exist. This account is populated when the connector is configured and saved in the LANDesk Software Distribution to AssetCenter scenarios.
As the account that is used is a LANDesk administrator account and since the tasks modify information contained in LANDesk, you should restrict access to these tasks to persons authorized to create scheduled tasks in AssetCenter.
- LANDesk Software Distribution scenarios use inventory data that is retrieved from an Enterprise Discovery to AssetCenter scenario.
- Enterprise Discovery and LANDesk agents must be installed.

Configure the LANDesk Software Distribution connector

Table 6.4.

 [Name and describe the connector \[page 17\]](#)

 [Specify the WSDL \[page 313\]](#)

 [SSL configuration \(advanced mode\) \[page 314\]](#)

 [Configure the JVM \[page 314\]](#)

 [Parallelize consumption \[page 314\]](#)

Name and describe the connector

To use this page, refer to chapter [Configuring connectors \[page 15\]](#), section [Name and describe the connector \[page 17\]](#).

Specify the WSDL

This page enables you to enter the connector's connection parameters to a Web service.

- **WSDL address** field: The WSDL URL address enables you to recover for each Web service:
 - The communication protocols used
 - The supported operations.
 - The format of these operations
 - The data consumed and the data produced
- **Domain\login** field: User ID
- **Password** field

SSL configuration (advanced mode)

This page allows you to define:

- the Java certificates file for the secure connection (**CA certificates file**).
By default, Connect-It searches for information concerning the Java certificate file in the JRE, in the following files: /lib/security/jsse.cacerts and /lib/security/cacerts.
- the private key file (**Certificates and client keys file (keystore)**). This file is required if the private key is stored in a database.
- the password for the private key file (**Keystore password**).

Configure the JVM

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Parallelize consumption

To use this page, refer to chapter [Configuring connectors](#) [page 15].

Out-of-the-box scenarios

The following out-of-the-box scenarios use this connector:

- ◆ [LANDesk scenarios](#) [page 386]

System Management System (SMS) connectors

Optional connector

◆ Production (source)

The SMS 2.x, and SMS 2003 (Microsoft Systems Management Service) gateway connectors enable you to process data from SMS data sources.

The SMS connectors only enable read-access to data in SMS databases. They do not enable you to write data to an SMS database.

Information to read before using the connectors

Warnings relating to the gateway connectors

Before going to production phase with a scenario containing a gateway connector, we highly recommend testing the scenario on a copy of the database into which the data is to be imported (for example: a copy of your AssetCenter database).

This test phase makes it possible to:

- Validate the reconciliation keys used in the mappings.
An invalid reconciliation key can bring about the creation of duplicate records in the target database. Removing these duplicate records manually is extremely difficult.
- Tailor the maptables, character string tables, and global functions.
The standard maptables, string tables, and global functions provided with Connect-It or the sample scenarios may not match your needs exactly.
For example: You may need to add entries in the **Brand** mactable. This table, which contains the brands of products (contained in the **Genmaps.mpt** file), may not contain all the brands you need.

Warnings related to the SMS connectors

All SMS connectors have been tested on original SMS databases. The out-of-the-box scenarios might not properly work or might not open in the following two cases:

- The database to which your SMS connector is connected has been customized (tables added or removed, field names modified, etc.). The document types

produced by the connector may not include the elements mapped in the scenario.











- The database to which your SMS connector is connected is not an SMS database. The produced document-types will not include the elements mapped in the scenario.

Configure the SMS connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.5. Configure the SMS 2.x and SMS 2003 connectors

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]

Prerequisites

Before configuring your connector, make sure a valid connection to your **SMS** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and password.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Production directives of the SMS connectors

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

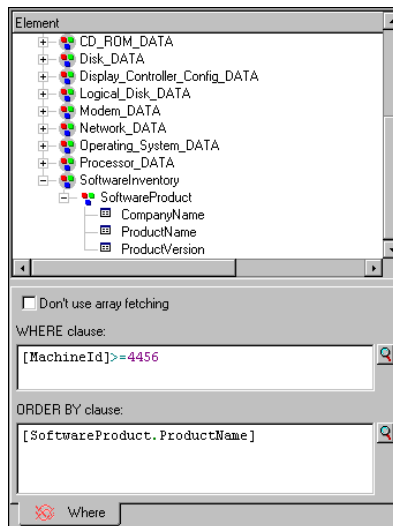
For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Example of a WHERE clause for the Asset Insight connector

To recover the record corresponding to the computers whose ID (dwMachineID) is 4575, you need to enter the following function in the WHERE clause field:

```
[dwMachineID] = 4575
```

Figure 6.4. SMS connectors - Production directive example



Out-of-the-box scenarios - SMS 2.x, and SMS 2003 connector

The following out-of-the-box scenarios use this connector:

- ◆ [SMS scenarios \[page 405\]](#)

ilInventory connector

Optional connector

◆ [Production \(source\)](#)

The ilInventory connector enables you to process data from an ODBC data source.

 **Warning:**

To use the ODBC connections, the ODBC Administrator must have been previously installed.

Known limitations of the ilInventory connector

We highly recommend using the ilInventory connector with an ODBC driver and ODBC administrator using identical version levels of DLL files.

Example :

The Inventory does not function if you are using ODBC Administrator 3.0 and a Microsoft Access 4.00 ODBC driver.

Type of supported fields

The document types published by the ilInventory connector are made of fields. Each of these fields has a particular type: text field, integer field, Blob field, etc.












The field types supported by the ilInventory connector are the Microsoft Access field types.

Configure the ilInventory connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors \[page 15\]](#), section [Configuring a connector \[page 16\]](#).

The following table presents all the pages in the connector's configuration wizard.

Table 6.6. Configure the inventory connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Manage transactions (advanced mode) [page 38]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]

Prerequisites

Before configuring your connector, make sure a valid connection to your **ODBC** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and passwords.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Manage transactions (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Manage transactions \(advanced mode\)](#) [page 38].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Production directives of the inventory connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information

about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

 **Note:**

The WHERE and ORDERBY clauses apply at the root level of the document type produced and at the level of the collection.

For more information about the AQL syntax, refer to the appendix [AQL queries](#) [page 427].

Sysdate field

In the WHERE clauses written in AQL, the use of the **sysdate** field must be replaced with the function **getdate()**.

Example: The SQL query: **SELECT * from AmAsset where dinstall>sysdate** must be replaced with the AQL query **SELECT * from AmAsset where dinstall>getdate()**.

Additional information

This chapter presents the following additional information:

- ◆ Additional information about declaring your ODBC data source.

Additional information about declaring your ODBC data source.

When a scenario using a iInventory connector is launched as a service in 32-bit Windows, the ODBC data source must be declared as a system data source (system DNS) and not as a user data source (user DNS).

Verifying that an ODBC data source is a system data source

To verify that a data source is a system data source:

- 1 Start ODBC Administrator.
- 2 Check that the ODBC data source used by your connector is located in the **System DNS** tab.

If it is not, delete it, and then create a new ODBC data source.

Out-of-the-box scenarios - Inventory connector

The following out-of-the-box scenarios use this connector:

- ◆ [Inventory scenarios](#) [page 383]

Tivoli Inventory connector (version 4.0)

Optional connector

◆ Production (source)

The Tivoli Inventory Management 4.0 connector enables you to process databases whose data was obtained using Tivoli Inventory 4.0. This application belongs to the Tivoli Inventory Management application suite.

Tivoli Inventory gathers information from computer scans. With this information, the dedicated out-of-the-box scenario lets you create records in the destination database.

Known limitations

Proper use of the Tivoli Inventory Management 4.0 connector requires that you use an ODBC driver and an ODBC administration application whose DLL versions are identical.

Example :

Your connector will not work if you use ODBC Administrator version 3.0 and an Access 4.00 ODBC driver.











Configure the Tivoli Inventory Management 4.0 connector

The configuration enables the connector to connect to your Tivoli Inventory Management 4.0 datasource.

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.7. Configure the Tivoli Inventory Management 4.0 connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]

Prerequisites

Before configuring your connector, make sure a valid connection to your **ODBC** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and password.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Production directives of the Tivoli Inventory Management connector 4.0

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Consumption directives of the Tivoli Inventory Management connector 4.0

For information about how to enter a connector's consumption directives, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51].

To set a connector's consumption directives, you must enter reconciliation parameters in the **Reconciliation** and **Advanced reconciliation** tabs.

For more information about these reconciliation tabs, refer to the chapter [Connector directives](#) [page 47], section [Consumption directives](#) [page 51], sub-section [Reconciliation](#) [page 52].

Out-of-the-box scenarios - Tivoli Inventory (version 4.0) connector

The following out-of-the-box scenarios use this connector:

- ◆ [Tivoli scenarios](#) [page 420]

Tivoli CM for Software Distribution 4.2 connector

This connector reads Tivoli Region Manager data, in particular data from the SD_PACKAGE table.

This connector is documented separately in the **TCM Integration** guide provided with Connect-It.

Tivoli CM for Inventory 4.2 connector

Optional connector

- Production (source)
- Consumption (destination)

The Tivoli CM Inventory 4.2 connector enables you to process the databases whose data was obtained using the Tivoli Configuration Management version 4.2 application.

This inventory function recovers complete information on an IT portfolio. From this information, the out-of-the-box scenarios migrate the data to an AssetCenter or ServiceCenter database.

Known limitations of the Tivoli CM - Inventory 4.2 connector

Proper use of the Tivoli CM - Inventory 4.2 connector requires that you use an ODBC driver and an ODBC administration application whose DLL versions are identical.

Example :

Your Tivoli CM - Inventory 4.2 does not work unless you use a version 3.0 of the ODBC administration software and an ODBC Microsoft Access 4.00 driver.

Mandatory initialization of the status pointer

In a scenario that includes the Tivoli CM Inventory 4.2 connector, the value of the connector pointer must be '0'. By default, the Tivoli CM Inventory 4.2 connector is unable to give a value to this pointer before the first launch of a scenario in scheduled mode.

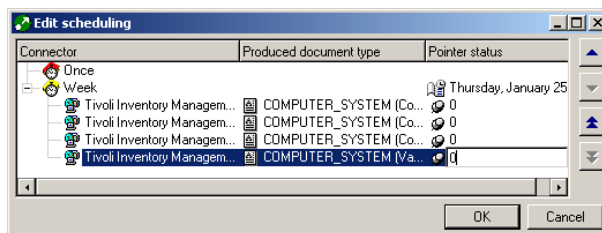
 **Warning:**

Without this operation the Tivoli CM Inventory 4.2 connector cannot produce any documents.

To assign this value to the pointer:

- 1 Select the **Scenario/ Scheduling** menu.

- 2 Associate the documents produced by the Tivoli CM Inventory 4.2 connector to the schedulers that you have already created.
- 3 For each produced document, double-click in the **Pointer status** column.
- 4 Enter '0' in the text box.



Configure the Tivoli CM Inventory 4.2 connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.8. Configure the Tivoli CM Inventory 4.2 connector

Name and describe the connector [page 17]
Select a connection type [page 18]
Configuring the connection [page 19]
Advanced configuration (advanced mode) [page 29]
Configure the reconnection parameters (advanced mode) [page 31]
Determine server delay (advanced mode) [page 33]
Configure schedule pointers (advanced mode) [page 36]
Configure the cache (advanced mode) [page 39]
Advanced options (advanced mode) [page 40]
Define document types (advanced mode) [page 42]

Prerequisites

Before configuring your connector, make sure a valid connection to your **ODBC** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and password.

The configuration of a connector in Connect-It is performed using a wizard.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

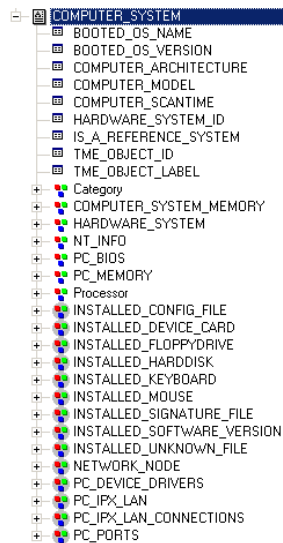
Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Published document-type

The Tivoli CM Inventory 4.2 connector publishes only one **COMPUTER_SYSTEM** document type. This document type corresponds to the scan that Tivoli Inventory launched on a given computer. The elements of this document type vary according to the configuration of your scanner generator. For example: Collections corresponding to database tables can occur or disappear from one scan to another one.

Figure 6.5. Tivoli CM - Inventory 4.2 connector - COMPUTER_SYSTEM document type



 **Warning:**

The elements of this document type can vary according to the structure of your Tivoli Configuration Management 4.2 databases. In addition, the tables represented by collections in the COMPUTER_SYSTEM document type can be absent from the database. In this case, if the scenario mapping calls on a field from one of these absent tables, this scenario cannot run. Solution: You must remove the implicated fields:

- 1 In the scenario mappings.
- 2 In the document types produced by the Tivoli CM Inventory 4.2 connector.

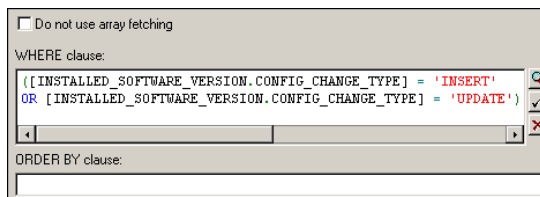
Production directives of the Tivoli CM Inventory 4.2 connector

The production directives of the Tivoli CM Inventory 4.2 connector enable it to filter and sort data from the ODBC data source.

To filter and sort the data that will appear in the documents produced by the connector, you must write clauses similar to those used in an SQL query:

- The WHERE clause
- The ORDER BY clause

Figure 6.6. Tivoli CM Inventory 4.2 connector - production directives



Do not use array fetching

WHERE clause:

```
((INSTALLED_SOFTWARE_VERSION.CONFIG_CHANGE_TYPE) = 'INSERT'
OR [INSTALLED_SOFTWARE_VERSION.CONFIG_CHANGE_TYPE] = 'UPDATE')
```

ORDER BY clause:

WHERE clause

This clause enables you to filter records in the ODBC database. You can, for example, filter records extracted from the database created on or after January 23, 2001. Simply use the following query:

```
[COMPUTER_SCANTIME] >= '2001-01-23'
```

ORDER BY clause

This clause enables you to indicate the field used to define the order in which documents are produced by the Tivoli CM Inventory 4.2 connector. You can enter a list of fields separating them with commas each time.

Do not use array fetching

This option must be selected, for example, when recovering Blob-type fields or **memo**-type fields when the table does not have a main field. If array fetching is used, Blob-type fields may not be processed correctly (they are clipped). This option may reduce performance in particular.

Out-of-the-box scenarios - Tivoli CM Inventory 4.2 connector

The following out-of-the-box scenarios use this connector:

- ◆ [Tivoli scenarios](#) [page 420]

Tivoli CM for Software Distribution Status 4.2 connector

This connector enables you to read data from the DIST_STATE table.



Scheduling pointers are not supported for this connector.

This connector is documented separately in the **TCM Integration** guide provided with Connect-It.

TS.Census 2 connector

Optional connector

◆ Production (source)

The TS.Census 2 gateway connector enables you to process data obtained from Tally System TS.Census 2. This application belongs to the Tally System TS.Census 2 software suite.

Tally System TS.Census 2 gathers information on IT portfolios. Using this information, the out-of-the-box scenario lets you create records in the destination database.

Known limitations of the TS.Census 2 connector

We highly recommend using the TS.Census 2 connector with an ODBC driver and ODBC Administrator using identical version levels of DLL files.

Example :








The TS.Census 2 connector will not work if you use ODBC Administrator version 3.0 and a Microsoft Access 4.00 ODBC driver.

Configure the TS.Census 2 connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.9. Configure the TS.Census 2 connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]

 [Configure the cache \(advanced mode\) \[page 39\]](#)

 [Advanced options \(advanced mode\) \[page 40\]](#)

 [Define document types \(advanced mode\) \[page 42\]](#)

Prerequisites

Before configuring your connector, make sure a valid connection to your **ODBC** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and password.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)



To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Published document-type

The TS.Census 2 connector publishes two types of documents: the **NC_Component** document and the **NC_Workstation** document.

The **NC_Component** document type corresponds to the computer-component scans (both hardware and/or software) carried out by TS.Census 2.

The **NC_Workstation** document type corresponds to the computer-related information gathered from a scan.

Element	Type
+  NC_Component	
+  NC_Workstation	

Note:

For more information about document types, refer to the Tally System TS.Census 2 documentation. All these tables and fields are described in the appendices.

Production directives of the TS.Census 2 connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

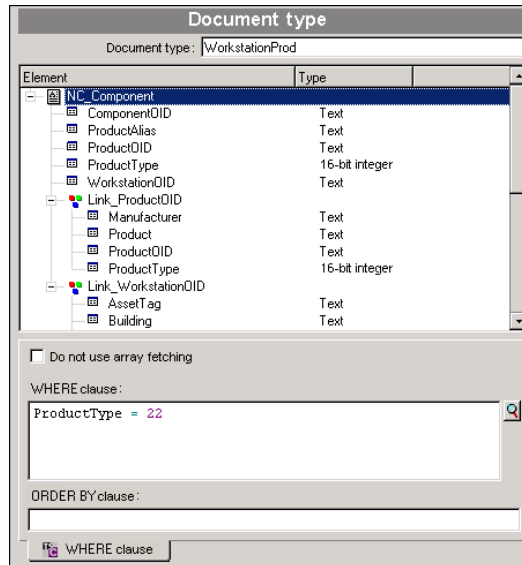
For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Example of production directives for the TS.Census 2 connector

If you want to recover the record from the NC_Component table whose product type is equal to 22, write the following WHERE clause:

```
[ProductType] =22
```

Figure 6.7. Production directives of the TS.Census 2 connector



Out-of-the-box scenarios - TS.Census 2 connector

The following out-of-the-box scenarios use this connector:

- ◆ [TS Census scenarios](#) [page 418]

TS.Census 3 connector

Optional connector

◆ Production (source)

The TS.Census 3 gateway connector enables you to process data obtained from Tally System TS.Census 3. This application belongs to the Tally System TS.Census 3 software suite.

Tally System TS.Census 3 gathers information about an IT portfolio. Using this information, the out-of-the-box scenario lets you create records in the target database.

Known limitations of the TS.Census 3 connector

We highly recommend using the TS.Census 3 connector with an ODBC driver and ODBC Administrator using identical version levels of DLL files.

Example :








The TS.Census 3 connector will not work if you use ODBC Administrator version 3.0 and a Microsoft Access 4.00 ODBC driver.

Configure the TS.Census 3 connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.10. Configure the TS.Census 3 connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]

 [Configure the cache \(advanced mode\) \[page 39\]](#)

 [Advanced options \(advanced mode\) \[page 40\]](#)

 [Define document types \(advanced mode\) \[page 42\]](#)

Prerequisites

Before configuring your connector, make sure a valid connection to your **ODBC** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and password.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)



To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Published document-type

The TS.Census 3 connector publishes two types of documents: the **NC_Component** document and the **NC_Workstation** document.

The **NC_Component** document type corresponds to the computer-component scans (both hardware and/or software) carried out by TS.Census 3.

The **NC_Workstation** document type corresponds to the computer-related information gathered from a scan.

Element	Type
+  NC_Component	
+  NC_Workstation	

 **Note:**

For more information about document types, refer to the Tally System TS.Census 3 documentation. All these tables and fields are described in the appendices.

Production directives of the TS.Census 3 connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

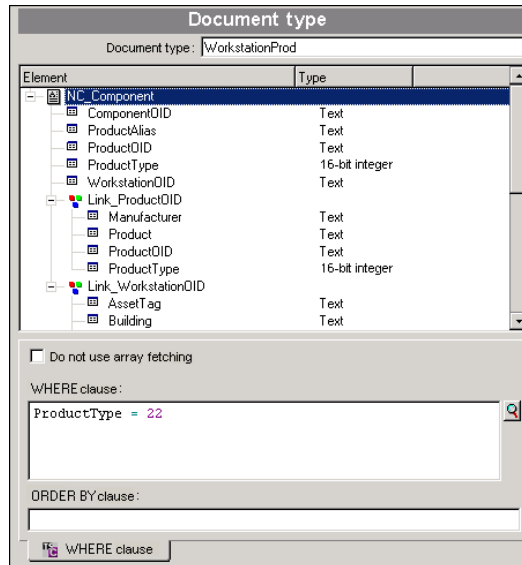
For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Example of production directives for the TS.Census 3 connector

If you want to recover the record from the NC_Component table whose product type is equal to 22, write the following WHERE clause:

```
[ProductType] =22
```

Figure 6.8. Production directives of the TS.Census 3 connector



Out-of-the-box scenarios - TS.Census 3 connector

The following out-of-the-box scenarios use this connector:

- ◆ [TS Census scenarios](#) [page 418]

CA Unicenter AMO 3 connector

Optional connector

◆ Production (source)

The CA Unicenter AMO gateway connector enables you to process databases that have been obtained using the Unicenter AMO network inventory tool. This inventory tool gathers extensive information about a computer population. Using this information, the CA Unicenter AMO to Asset Management scenario (**amoac.scn**) enables you to create records in the Assets table of an Asset Management application.

Known limitations of the CA Unicenter AMO 3 connector

We highly recommend using the CA Unicenter AMO connector with an ODBC driver and ODBC Administrator using identical version levels of DLL files.

Example :








Your AC Unicenter AMO connector will not function unless you use an ODBC Administrator version 3.0 and a Microsoft Access ODBC driver 4.00.

Configure the CA Unicenter AMO 3 connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.11. Configure the Unicenter AMO 3 connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]

 [Configure the cache \(advanced mode\) \[page 39\]](#)

 [Advanced options \(advanced mode\) \[page 40\]](#)

 [Define document types \(advanced mode\) \[page 42\]](#)

Prerequisites

Before configuring your connector, make sure a valid connection to your **ODBC** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and password.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Production directives of the CA Unicenter AMO connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Published document-type

The CA Unicenter AMO connector publishes the following document types:

- Computer
- Motor
- Group
- User

Out-of-the-box scenarios - CA Unicenter AMO connector

The following out-of-the-box scenarios use this connector:

- ◆ [Unicenter AMO scenarios](#) [page 424]

CA Unicenter AMO 4 connector

Optional connector

◆ Production (source)

The CA Unicenter AMO gateway connector enables you to process databases that have been obtained using the Unicenter AMO network inventory tool.

This inventory tool gathers extensive information about a computer population. Using this information, the CA Unicenter AMO to Asset Management scenario (**amoac.scn**) enables you to create records in the Assets table of an Asset Management application.

Known limitations of the CA Unicenter AMO 4 connector

We highly recommend using the CA Unicenter AMO connector with an ODBC driver and ODBC Administrator using identical version levels of DLL files.

Example :








Your AC Unicenter AMO connector will not function unless you use an ODBC Administrator version 3.0 and a Microsoft Access ODBC driver 4.00.

Configure the CA Unicenter AMO 4 connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.12. Configure the CA Unicenter AMO 4 connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]

 [Configure the cache \(advanced mode\) \[page 39\]](#)

 [Advanced options \(advanced mode\) \[page 40\]](#)

 [Define document types \(advanced mode\) \[page 42\]](#)

Prerequisites

Before configuring your connector, make sure a valid connection to your **ODBC** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and password.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Production directives of the CA Unicenter AMO 4 connector

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Published document-type

The CA Unicenter AMO 4 connector publishes the following document types:

- Application
- Computer
- Motor
- Group
- User
- DirectAccessTables

This document type contains a collection of all the other document types published by the connector. You can directly map this collection to a document type consumed by a destination connector.

For more information on Collection-Document mappings, refer to the **User's guide**, chapter **Mapping document types**, section **Mapping source elements to destination elements/Collection-to-document mapping**.

Out-of-the-box scenarios - CA Unicenter AMO 4 connector

The following out-of-the-box scenarios use this connector:

- ◆ [Unicenter AMO scenarios](#) [page 424]

Winpark Actima connector

Optional connector

◆ Production (source)

The Winpark Actima connector enables you to process databases that have been obtained using an network inventory tool from the Winpark Actima software suite .

This inventory tool gathers extensive information about a computer population. Using this information, the provided Winpark Actima to Asset Management scenario (**wpkam.scn**) enables you to create records in the Assets table of an Asset Management application.

Known limitations

We highly recommend using the Winpark Actima connector with an ODBC driver and ODBC Administrator using identical version levels of DLL files.

Example :

Your Winpark Actima connector will not function unless you use an ODBC Administrator version 3.0 and a Microsoft Access ODBC driver 4.00.











If you loose your connection using Paradox 5.0, disconnect and reconnect. This generally solves the problem.

Configure the Winpark Actima connector

Using a wizard, you can configure this connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 6.13. Configure the Winpark Actima connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]

Prerequisites

Before configuring your connector, make sure a valid connection to your **ODBC** database has been declared in ODBC Administrator.

A valid ODBC declaration must use an appropriate login and password.

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Configure schedule pointers (advanced mode)

To use this page, refer to the chapter [Configuring connectors](#) [page 15], section [Configure schedule pointers \(advanced mode\)](#) [page 36].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Production directives of the Winpark Actima

For information about how to enter a connector's production directives, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48].

For this connector, these directives involve writing WHERE and ORDERBY clauses, which enable you to filter records from the source database. For more information

about such clauses, refer to the chapter [Connector directives](#) [page 47], section [Production directives](#) [page 48], sub-section [WHERE and ORDERBY clauses](#) [page 49].

Document types published by the Winpark Actima connector

The Winpark Actima connector publishes three document types:

- Purchases
- Budget
- Contracts
- Requests
- DirectAccessTables
- Entity
- Invoices
- Supplier
- Work order
- Software
- Peripheral
- Post
- Network
- Site
- User

Out-of-the-box scenarios - Winpark Actima connector

The following out-of-the-box scenarios use this connector:

- ◆ [Winpark Actima scenarios](#) [page 425]



7 ERP connectors

CHAPTER

The ERP (Enterprise Resource Planning) connectors enable you to process data from Enterprise Resource Planning applications.

SAP connectors

Connect-It proposes the following SAP connectors:

- [SAP BAPI connector](#) [page 353]
This connector publishes document types corresponding to the BAPI methods exposed by an SAP R/3 server. This connector enables the synchronous processing of this data.
- [SAP IDoc connector](#) [page 357]
This connector publishes document types corresponding to flat text files in the IDOC format. This IDOC format enables you to send and receive data to and from an SAP R/3 server.
- [SAP ALE connector](#) [page 360]
This connector uses the ALE communication layer to publish document types corresponding to the flat text files in IDOC.

RFC (Remote Function Calls)

The RFCs are used to the SAP BAPI connectors. They enables the SAP or non-SAP applications to access the functions of an SAP R/3 server in a network architecture.

The RFCs:

- Enable you to call on and process procedures predefined on an SAP server.
- Manage communications, parameter transfers, and error messages.
- Are controlled by the standard SAP authentication procedures. These authentication procedures verify that the application data is read and edited in a secured and consistent way.

An SAP R/3 server provides thousands of functions, most of which can be remotely accessed. These are usually written using the **ABAP/4** (Advanced Business Application Programming) language.

The RFCs are bi-directional, which enables an SAP client application to access the functions in the non-SAP applications.

RFC libraries and Java archives

To use the SAP connectors, you must use Java archives (**.jar**) and native Java libraries (**.dll**, **.so** or **.o**) that you obtain using the following procedure:

- 1 Launch your Internet browser.
- 2 Enter the following address: <http://service.sap.com/connectors>

 **Note:**

To access this site, you must obtain an authorization certificate from SAP support.

-
- 3 Select the **SAP Java Connector** section.
 - 4 Select the **Tools & Services** section.
 - 5 Click **Download SAP JCo Release 2.1.2**.

This download enables you to obtain a compressed file (**.zip** or **.tgz**) that contains the files necessary for the SAP connectors to work.

 **Note:**

Other versions of SAP JCo are supported.

We recommend consulting the compatibility matrix of this version of Connect-It on Peregrine System's site: <http://www.peregrine.com>

- 6 Uncompress the compressed file on your disk.
- 7 Copy the **sapjco.jar** archive to the **lib** sub-folder of the Connect-It installation folder.
- 8 Install the native Java libraries. This step depends on your operating system. Copy the following libraries to the **bin32** or **bin** sub-folder of the Connect-It installation folder:
 - **librfc32.dll** and **sapjcorfp.dll** for Microsoft Windows operating systems
 - **librfccm.so** and **libsapjcorfc.so** for Linux and Sun Solaris operating systems
 - **librfccm.o** and **libsapjcorfc.o** for IBM AIX operating systems

SAP BAPI connector

Optional connector

- Consumption (destination)
 - Consumption - spontaneous production (request - response)
-

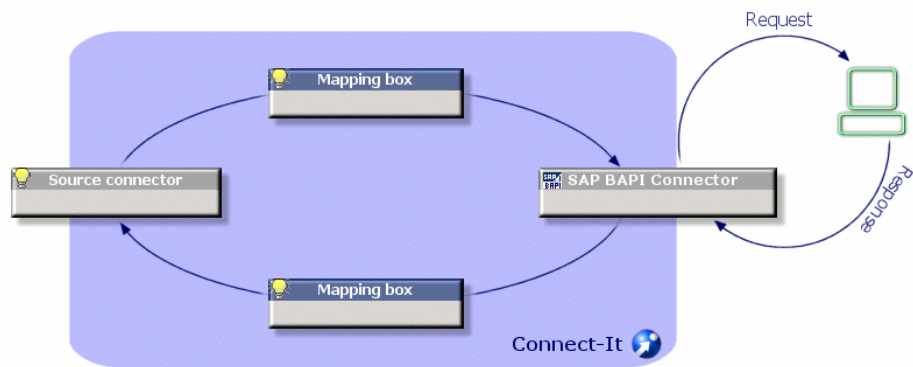
The SAP BAPI connector connects to an SAP R/3 server and obtains a list of (BAPI) services sorted by business object.

The document types published by the SAP BAPI connector are organized as follows:

- Consumed document types
These document types let the connector send data to the SAP server.
- Consumed-Produced document types
These document types let the connector:
 - Send a request to the SAP server (consumption)
 - Receive a response from this request (production)

The way an SAP connector works when using consumed-produced document types is illustrated by the following diagram.

Figure 7.1. SAP connectors - Consumed-produced document-type



Limitations of the SAP BAPI connector

This connector does not work with versions earlier than 3.5 of SAP R/3, which introduced the use of BAPIs.

ABAP data types

The following chart shows how the ABAP data and the Connect-It data are equivalent.

Table 7.1. Correspondence between ABAP data types and Connect-It data types





ABAP type	JCO data type	Connect-It data type
C	TYPE_CHAR	Text
D	TYPE_DATE	Date
P	TYPE_BCD	Double integer
T	TYPE_TIME	Date
X	TYPE_BYTE	Blob
N	TYPE_NUM	Integer
F	TYPE_FLOAT	Double integer
I	TYPE_INT	Integer
b	TYPE_INT1	Integer
s	TYPE_INT2	Integer
g	TYPE_STRING	Text

Configure the SAP BAPI connector

A wizard enables you to configure the connector.

The following table presents all the pages in the connector's configuration wizard.

Table 7.2. Configure the SAP BAPI connector

 Name and describe the connector [page 17]
 Configure server connection [page 355]
 Configure the JVM [page 356]
 Parallelize consumption (advanced mode) [page 44]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Configure server connection

This page enables you to indicate the connection parameters of the connector to the SAP server.

SAP server

Indicate the name of your SAP server or the routing string to access the server.
For example:

```
/H/saprouter1/H/saprouter2/H/sapserver
```

System number

Indicate the system number that you want to use with your SAP server.

Client

Indicate the client port.

Example: **800**

Login

Indicate your user name.

Password

Indicate the password corresponding to the user name, specified in the previous field.

Language

Indicate the language in which the information from the server should appear.

Example: **en** for English, **de** for German, etc.

Enable Auto Commit/RollBack mode after each document

This option enables you to automatically enable the **BAPI_TRANSACTION_COMMIT** BAPI call (or **BAPI_TRANSACTION_ROLLBACK**) after the BAPI user call in order to validate the changes.

This option can only be used in write mode.

Configure the JVM

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

Parallelize consumption (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15].

Out-of-the-box scenarios - SAP BAPI connector

The following out-of-the-box scenarios use this connector:

- ◆ [SAP BAPI scenarios](#) [page 402]

SAP IDoc connector

Optional connector

- Production (source)
- Consumption (destination)

Services and events for SAP IDoc use a flat file format (located in a folder shared by the network or on the FTP) to pass information from and to SAP R/3. The IDoc is a data structure defined by SAP that supports data exchange between any two applications in a common format.

The SAP R/3 system provides hundreds of IDocs, but it also allows you to create your own IDoc types or expand existing ones.

An IDoc instance consists of three, distinct parts:

- A header
 - This header contains a string that lets the SAP R/3 server to uniquely identify an Idco file.
- Data
 - This data is organized in segments.
- A series of statuses.
 - The file went through all of these statuses on its path to or from the SAP R/3 server.

Warning:

For the IDoc files containing multiple documents, each document must be separated by a header (**EDI_DC** segment).

Configure the SAP IDoc connector

A wizard enables you to configure the connector.

A consumed document type corresponds the sending of the data to an IDoc file.











A produced document type corresponds reading of the data in an IDoc file.

The following table presents all the pages in the connector's configuration wizard.

 Note:

If you wish to use multiple IDoc file types, you must deploy as many connectors as there are different IDoc files. Since each connector has a unique behavior (production or consumption), you must create as many connectors as there are behaviors.

Table 7.3. Configure the SAP IDoc connector

 Name and describe the connector [page 17]		
 Choosing a processing mode [page 277]		
 Selecting a connection protocol [page 277]		
 Local or network file(s) [page 280]	 Connecting to an HTTP Web site [page 277]	 FTP server connection [page 279]
 Define post-processing actions [page 280]	 Define post-processing actions [page 279]	
 IDoc type definition (read and write) [page 358]		
 Configure the JVM [page 360]		

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

IDoc type definition (read and write)

This page enables you to choose the IDoc document types and the way you want to get it:

- IDoc types in local
 - The **Local file** option lets you use the IDoc types already present on your computer.
- IDoc types of the SAP server
 - The **On the SAP server** option enables to dynamically obtain the IDoc types on your SAP server.

The available fields change depending on which of these options you select.

IDOC type file

If you selected the option **Local IDoc types**, click **Browse** to find the location of the IDoc type on your computer.

SAP version (write)

Indicate the version of the SAP you use.

SAP server connection settings

The following fields enable you to indicate the following SAP server connection parameters:

- SAP server
- User
- Password
- Language
- Client
- System identifier

IDoc settings

Basic type

Name of the SAP document you want to produce.

For example, **MATMAS01**.

Extension

This field enables you to enter:

- A CIM type for the IDoc types
Version **2** or **3**.
- An SAP server CIM type
Available versions: **3.1 G** to **4.6 C**.

Segment release

SAP version on which the segment depends. The most recent version of the application is taken into account and is adapted to the oldest version of the segment.

Version of IDoc record type

Version 2 record types correspond to SAP 3.x.

Version 3 record types correspond to SAP 4.x.

Configure the JVM

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].

SAP ALE connector

The SAP ALE connector processes the IDoc files like the SAP IDoc connector. However, it uses the ALE (Application Link Enabling) communication layer to send and receive these files to and from the SAP R/3 server. The events that it receives (corresponding to produced document types) can also use the SAP R/3 server RFCs.

Unlike the SAP IDoc, it supports both synchronous and asynchronous data processing (event-driven mode).

Prerequisites





The following elements must be defined for the SAP server:

- 1 logical system
- 2 RFC destination
- 3 tRFC port
- 4 distribution model
- 5 partner profile

Configure the SAP ALE connector

A wizard enables you to configure the connector.

The following table presents all the pages in the connector's configuration wizard.

 Nommer et décrire le connecteur [page 361]
 Select a processing mode [page 361]
 Configure SAP server configuration [page 361]
 Configure the reconnection parameters (advanced mode) [page 362]

Nommer et décrire le connecteur

To use this page, refer to chapter [Configuring connectors \[page 15\]](#), section [Configuring a connector \[page 16\]](#) .

Select a processing mode

This page enables you to define whether the connector works in read or write mode.

Configure SAP server configuration

This page enables you to define the connection parameters for the SAP server.

SAP server

Name of the SAP server

System number

Indicate the system number that you want to use with your SAP server.

Client

Indicate the client port.

Login

Indicate your user name.

Password

Indicate the password corresponding to the user name, specified in the previous field.

Language

Indicate the language in which the information from the server should appear.

Example: **en** for English, **de** for German, etc.

Logical system

Name of the logical system.

Gateway host

Name of the host machine of the gateway.

Populate this field if the SAP server is not the host of the gateway.

Gateway service

Name of the TCP/IP socket of the connection gateway of the SAP R/3 system. All systems connecting to the SAP R/3 system in the background use this socket at the entry point.

Program ID

Name of the identifier of the program.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Configure the JVM

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the JVM](#) [page 43].



8 | Hewlett-Packard connectors

CHAPTER

The Hewlett-Packard connectors are used to process data from OpenView Service Desk.

OpenView Service Desk (Outbound)

Optional connector

◆ Consumption

The OpenView Service Desk connector (Inbound) is used to:











- ◆ Process data from an OpenView Service Desk database.

Configure the OpenView Service Desk (Outbound) connector

A wizard enables you to configure the connector. Consult chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 8.1. Configure the OpenView Service Desk (Outbound) connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

- **SQL92 supported:** if you use the **AssetCenter Database** ODBC driver, disable this option.
- **Perform an initial import:** this option avoids using a reconciliation script in a SELECT query for an initial import, i.e. when the database does not contain any records or when the elements to be inserted do not exist yet. This allows record processing time to be reduced.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Manage transactions (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Manage transactions \(advanced mode\)](#) [page 38].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Out-of-the-box scenarios - OpenView Service Desk (Outbound) connector

The following out-of-the-box scenarios use this connector:

- ◆ **AC2SD.scn**

This scenario is documented separately. Please consult the [SDAC_Integration_Userguide](#).

OpenView Service Desk (Inbound)

Optional connector

◆ Production

The OpenView Service Desk connector (Inbound) is used to:

- ◆ Process `.xml` files from OpenView Service Desk.

Configure the OpenView Service Desk (Inbound) connector

You can configure the connector using a wizard.

The following table presents all the pages in the connector's configuration wizard.

Table 8.2. Configure the OpenView Service Desk (Inbound) connector

 Name and describe the connector [page 17]

 Select a connection protocol [page 366]

 Select files or folders [page 366]

 Behavior between two sessions [page 368]

 Choose a DTD/XSD [page 368]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection protocol

This page lets you define the connection protocol used for the connector.

The protocol used for this connector is defined by default.

Select files or folders

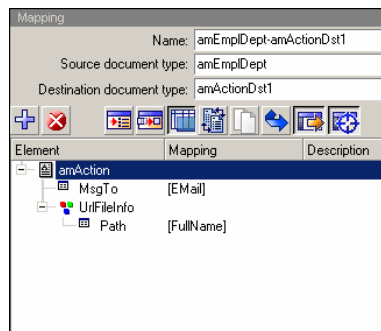
This page enables you to select the XML file or files that your connector must read.

Before defining the connector's behavior, specify the folder for writing files (**Folder name** field). The path for this folder is used in a mapping in combination with the `UrlFileInfo.Path` element.

You can choose between the following behaviors:

- Write documents to one single file
Enter the path of a folder on your computer or your network (**Folder name** field). The document will be overwritten or left alone depending on the parameters defined in the next page of the wizard.
- Write to a different file for each document
Specify the folder in which the files will be written (**Folder name** field). The name of these files will correspond to the generic name that you have entered in the consumption directives of the XML connector. Refer to the [Consumption directive of the XML connector \(write\) - FTP mode](#) [page 296] sub-section of this section.

You can also define the names of these files by mapping the element **UrlFileInfo.Path**. For example, to create a file with the name of each employee in AssetCenter. In this case, the path defined in the wizard page is concatenated with the path defined in the mapping.



In order for the path defined in the mapping to be used as the default path, and as a result to avoid using the path defined in the wizard, use the '/' character in the mapping.

 **Note:**

This behavior does not prevent you from specifying a valid path in the connector configuration wizard.

If the path specified for the **UrlFileInfo.Path** element does not exist, it is created automatically provided that:

- the path uses part of the path specified in the configuration wizard.

- the path does not add more than one level to the specified tree. For example, for a path **a/b/c** specified in the configuration wizard, only a path **a/b/c/d/filename**, specified in the mapping, is possible.

Behavior between two sessions

This page enables you to select how documents are processed between two data-writing sessions.

Append to the same file

The connector starts writing data to the open file again from where the previous session was interrupted.

Overwrite the last file

The connector deletes the file in which it wrote data during the previous session.


Number the different files

The connector creates a file whose name is the same as the previous file, but incremented by 1 unit. Example: **file.xml**, **file1.xml**, **file2.xml**, etc.

Choose a DTD/XSD

To create an XML file, the connector must use a DTD (Document Type Definition) or XSD (Extended Schema Definition). On the **Select a DTD/XSD** page, indicate the DTD or XSD used in the **DTD/XSD** field. The document types published by the XML connector correspond to this DTD or XSD.

A DTD is provided and populated with default values for this connector.

By clicking , the **File location** page is displayed. This page enables you to enter the path of your DTD or XSD depending on your connection protocol.

File location

You have the choice of three options:

- **Local/Network**
Enter the full path of your DTD or XSD file located on the computer or network on which Connect-It is installed.
- **FTP**
Enter the FTP parameters in the **Server**, **Login**, **Password**, and **Path** fields.

The **Path** field enables you to specify the location of your DTD or XSD file on the server. Example: If your DTD is located in the **myfolder/dtd** folder on the **http://mycompany.com** server, you must populate the **Server** and **Path** fields with **http://mycompany.com** and **/myfolder/dtd/mydtd.dtd**.

 **Note:**

Do not include a slash mark (/) at the end of the server name; instead include it at the start of the path of the DTD or XSD file.

■ HTTP

Enter the HTTP parameters in the **Server**, **Login**, **Password**, and **Path** fields. The **Path** field enables you to specify the location of your DTD or XSD file on the server. Example: If your DTD is located in the **myfolder/dtd** folder on the **http://mycompany.com** server, you must populate the **Server** and **Path** fields with **http://mycompany.com** and **/myfolder/dtd/mydtd.dtd**.

 **Note:**

Do not include a slash mark (/) at the end of the server name; instead include it at the start of the path of the DTD or XSD file.

Determine the root elements

A DTD or XSD is composed of elements that can contain other elements. (For example: In a DTD intended for book publishing, **Section** elements are included in the **Chapter** elements, which are in turn included in the **Book** element.) The only element that cannot be included in another element is a root element. A DTD can contain any number of root elements.

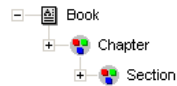
To determine the root elements, which correspond to as many document types as published by the XML connector, Connect-It gives you different options:

- Publish a document type for each root element found in the DTD/XSD (recommended)
- Publish a document type for each root element selected by the user (values are separated by commas)

Publish a document type for each root element found in the DTD/XSD (recommended)

If you choose this option, the XML connector searches all the root elements of the DTD or XSD and publishes a document type for each root element found.

Example: In a DTD intended for book publishing, only one root element (**Book**) is found. In this case, the XML connector will only publish one document type.



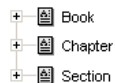
 Note:

When no root elements can be found (all the elements that can include other elements), Connect-It will take the first element found in the DTD or XSD as its only root element. Example: A DTD corresponds to the organization of a relational database in which all the elements (each one representing a table) include one another: The Assets table is linked to the Users table, which is linked to the Assets table, etc.

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

If you select this option, you can indicate the root elements of your choice, separating each one with a comma.

Example: In a DTD intended for book publishing, the user decides to choose the elements **Book**, **Chapter**, and **Section**. In this case, the XML connector will publish one document type per root element chosen by the user.



 Note:

For a DTD corresponding to the organization of a relational database in which all the elements (each representing a table) include one another, this option enables you to obtain a published document-type for each one of these tables.

OpenView Inventory Manager Connector

Optional connector

◆ Production (source)

The OpenView Inventory Manager connector enables you to:











- ◆ Process data from an Inventory Manager database.

Configure the OpenView Inventory Manager connector

A wizard allows you to configure the connector. Refer to chapter [Configuring connectors](#) [page 15], section [Configuring a connector](#) [page 16].

The following table presents all the pages in the connector's configuration wizard.

Table 8.3. Configure the OpenView Inventory Manager connector

 Name and describe the connector [page 17]
 Select a connection type [page 18]
 Configuring the connection [page 19]
 Advanced configuration (advanced mode) [page 29]
 Configure the reconnection parameters (advanced mode) [page 31]
 Determine server delay (advanced mode) [page 33]
 Configure schedule pointers (advanced mode) [page 36]
 Configure the cache (advanced mode) [page 39]
 Advanced options (advanced mode) [page 40]
 Define document types (advanced mode) [page 42]

Name and describe the connector

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Name and describe the connector](#) [page 17].

Select a connection type

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Select a connection type](#) [page 18].

Configuring the connection

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configuring the connection](#) [page 19].

Advanced configuration (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced configuration \(advanced mode\)](#) [page 29].

- **SQL92 supported:** if you use the **AssetCenter Database** ODBC driver, disable this option.
- **Executing an initial import:** This option enables you to avoid SELECT queries in reconciliation scripts for initial imports, for example, when the database is empty or the items to be inserted do not yet exist. This shortens the processing time.

Configure the reconnection parameters (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the reconnection parameters \(advanced mode\)](#) [page 31].

Determine server delay (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Determine server delay \(advanced mode\)](#) [page 33].

Manage transactions (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Manage transactions \(advanced mode\)](#) [page 38].

Configure the cache (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Configure the cache \(advanced mode\)](#) [page 39].

Advanced options (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Advanced options \(advanced mode\)](#) [page 40].

Define document types (advanced mode)

To use this page, refer to chapter [Configuring connectors](#) [page 15], section [Define document types \(advanced mode\)](#) [page 42].

Out-of-the-box scenarios - OpenView Inventory Manager connector

The following out-of-the-box scenarios use this connector:

- ◆ [OpenView Inventory Manager Scenarios](#) [page 385]

9 Out-of-the-box scenarios

CHAPTER

This chapter presents the scenarios provided with Connect-It.

The scenarios presented in this chapter are classified according to their source connector and can be divided into three categories:

- **Sample scenarios**

These scenarios only enable you to view how one or more connectors work. They have no other value and should not **under any circumstances** be used in a production environment.

The sample scenarios are designated in this chapter with the following icon:



- **Business scenarios to adapt**

These scenarios correspond to integration processes that actually occur in real situations. However, to be used in production environments, they must be adapted by the user.

These adaptations notably concern the mappings between source connectors and destination connectors.

The business scenarios are designated in this chapter with the following icon:



- **Business scenarios**

These scenarios correspond to integration processes that actually happen in real situations. They can be used directly in production environments.

The business scenarios are designated in this chapter with the following icon:



Altiris scenarios

This section briefly describes the out-of-the-box scenarios that use the Altiris connector.

altiris\altiris61ac44\altirisac.scn

This scenario enables you to transfer data from an Altiris database to an AssetCenter database.

The following data is transferred:

- computers
- installed software
- monitors
- installed cards
- logical hard drives
- physical hard drives

altiris\altiris61ac44\altirisac_swnorm.scn

This scenario enables you to transfer data from an Altiris database to an AssetCenter database.

This scenario is used for software normalization.

Action Request System scenarios

This section briefly describes the out-of-the-box scenarios using the Action Request System connector.

ars\rsm\rsm4ac36\cmpo.scn



This scenario enables you to transfer data from Action Request System to an AssetCenter 3.6 database.

The data items transferred by this scenario are the following:

- Purchase request
- Category
- Models

The following table lists the mapped source and destination document types.

Source	Destination
CHG:ChangeSrc (Request creation)	amRequestDst1
CHG:ChangeSrc (Category creation)	amCategoryDst1
CHG:ChangeSrc (Model creation)	amProductDst

ars\rsm\rsm4ac36\sharedat.scn



This scenario enables you to transfer data from Action Request System to an AssetCenter 3.6 database.

The data items transferred by this scenario are the following:

- Employees
- Locations
- Assets

The following table lists the mapped source and destination document types.

Source	Destination
SHR:PeopleSrc1	amEmplDeptDst1
SHR:LocationSrc	amLocationDst

Source	Destination
AST:AssetSrc	amAssetDst

Asset Insight scenarios

This section briefly describes the out-of-the-box scenarios using the Asset Insight connector.

ist\ist4ac36\istac.scn



This scenario enables you to transfer data from an Asset Insight 4 database to an AssetCenter 3.6 database.

The following data is transferred:

- Computers
- Devices
- Software

ist\ist4ac43\istac.scn



This scenario enables you to transfer data from an Asset Insight 4 database to an AssetCenter 4.3 database.

ist\ist4ac44\istac.scn



This scenario has the same functions as the [ist\ist4ac36\istac.scn](#) [page 378] scenario.

ist\ist4ac50\istac.scn



This scenario has the same functions as the [ist\ist4ac36\istac.scn](#) [page 378] scenario.

Asset Management scenarios

This section briefly describes the out-of-the-box scenarios using the Asset Management connector.

ac\ac43\catalog.scn



This scenario enables you to import catalog-related data into an AssetCenter database.

For further information, refer to the AssetCenter **Reconciliation Guide**, chapter **Sample scenarios**.

ac\ac44\catalog.scn

This scenario has the same functions as the [ac\ac43\catalog.scn](#) [page 379] scenario.

ac\ac50\catalog.scn

This scenario has the same functions as the [ac\ac43\catalog.scn](#) [page 379] scenario.

E-mail scenarios

This section briefly describes the out-of-the-box scenarios using the e-mail connectors.

mail\mailac36\finreque.scn



This scenario enables you to send a purchase request (corresponding to a record in the Requests table) to an e-mail address (that of the purchase approver).

The following table lists the mapped source and destination document types.

Source	Destination
amRequestSrc	OutMailMessageDst

mail\mailac36\finconfi.scn



This scenario enables you to send a message corresponding to a purchase approval. The status of the request is changed in the Request table in the Asset Management application.

The following table lists the mapped source and destination document types.

Source	Destination
InMailMessageSrc	amRequestDst

mail\mailac36\newemplo.scn



This scenario enables you to create a record in the Departments and Employees table in the Asset Management application from an e-mail message.

The following table lists the mapped source and destination document types.

Source	Destination
InMailMessageSrc	amEmplDeptDst

mail\mailac43\finconfi.scn



This scenario enables you to send a message corresponding to a purchase approval. The status of the request is changed in the Purchase Requests table in the Asset Management application.

The following table lists the mapped source and destination document types.

Source	Destination
InMailMessageSrc	amRequestDst

mail\mailac43\finreque.scn



This scenario enables you to send a purchase request (corresponding to a record in the Requests table) to an e-mail address (that of the procurement manager).

The following table lists the mapped source and destination document types.

Source	Destination
amRequestSrc	OutMailMessageDst

mail\mailac43\newemplo.scn



This scenario enables you to create a record in the Departments and Employees table of the Asset Management application from an e-mail message.

The following table lists the mapped source and destination document types.

Source	Destination
InMailMessageSrc	amEmplDeptDst

Enterprise Discovery scenarios

This section briefly describes the out-of-the-box scenarios using the Enterprise Discovery connector.

These scenarios are described in a separate document. Refer to the **Enterprise Discovery - Customization** manual, in the **Getting your data in AssetCenter** chapter.

- ▶ Connect-It - Integration with an AssetCenter database.

ed\ed2ac44\edac.scn



This scenario is used to transfer inventory data to an AssetCenter application.

ed\ed2ac44\edac-reconc.scn



This scenario is used to reconcile inventory data for an AssetCenter application.

ed\ed2ac44\edac-swnorm.scn



This scenario is used to access the software normalization module available in AssetCenter.

ed\ed2ac50\edac.scn



This scenario is used to transfer inventory data to an AssetCenter application.

ed\ed2ac50\edac-reconc.scn



This scenario is used to reconcile inventory data for an AssetCenter application.

ed\ed2ac50\edac-swnorm.scn



This scenario is used to access the software normalization module available in AssetCenter.

ed\ed2Sc60\edsc.scn



This scenario transfers inventory data to a ServiceCenter application.

ed\ed2Sc61\edsc.scn



This scenario has the same functions as the scenario [ed\ed2Sc60\edsc.scn](#) [page 383].

Inventory scenarios

This section briefly describes the out-of-the-box scenarios using the Network Discovery connector.

iinv5ac43.scn\iinvac.scn



This scenario enables you to transfer the data from a Microsoft Access (hardware and software information) database to the Computers table in ServiceCenter 4.3.

iinv5ac44.scn\iinvac.scn



This scenario performs the same functions as the `iinv5ac43.scn\iinvac.scn` [page 383] scenario

`iinv5ac50.scn\iinvac.scn`



This scenario has the same functions as the `iinv5ac43.scn\iinvac.scn` [page 383] scenario.

`iinv5sc51.scn\iinvsc.scn`



Before executing this scenario, you must import the `pcsoft-uninst.unl` and `pcsoftware.unl` files into ServiceCenter. These files are provided with the installation of Connect-It (`datakit\sc` folder of your installation folder).

This scenario enables you to update the information relating to the computers (and their software) in a ServiceCenter 5.1 database using data obtained with iInventory LANAuditor.

`iinv5sc6.scn\iinvsc.scn`



This scenario enables you to update the information relating to the computers (and their software) in a ServiceCenter 6 database using data obtained with iInventory LANAuditor.

`iinv5sc61.scn\iinvsc.scn`



This scenario has the same functions as the scenario `iinv5sc6.scn\iinvsc.scn` [page 384].

OpenView Inventory Manager Scenarios

This section briefly describes the out-of-the-box scenarios using the OpenView Inventory Manager connector.

hpovcm\im42ac44\imac.scn

This scenario enables you to transfer data from an OpenView Inventory Manager database to an AssetCenter database.

The following data is transferred:

- computers
- logical hard drives
- physical hard drives
- network cards
- internal devices

hpovcm\im42ac44\imac_swnorm

This scenario enables you to transfer data from an OpenView Inventory Manager database to an AssetCenter database.

This scenario is used for software normalization.

hpovcm\im42ac44\imsc.scn

Before executing this scenario, you must import the **pcsoft-uninst.un1** and **pcsoftware.un1** files into ServiceCenter. These files are provided with the installation of Connect-It (datakit\sc folder of your installation folder).

This scenario enables you to transfer data from an OpenView Inventory Manager database to a ServiceCenter database.

LANDesk scenarios

This section briefly describes the out-of-the-box scenarios using the LANDesk connector.

ldsk\ld8ac43\ldskac.scn



This scenario enables you to transfer data from a LANDesk 8 database to an AssetCenter 4.3 database.

- Computers
- Devices
- Software

ldsk\ld8ac44\ldskac.scn



This scenario has the same functions as the scenario [ldsk\ld8ac43\ldskac.scn](#) [page 386].

ldsk\ld8ac50\ldskac.scn



This scenario has the same functions as the scenario [ldsk\ld8ac43\ldskac.scn](#) [page 386].

ldsk\ld8sc51\ldsksc.scn



Before executing this scenario, you must import the **pcsoft-uninst.un1** and **pcsoftware.un1** files into ServiceCenter. These files are provided with the installation of Connect-It (**datakit\sc** folder of your installation folder).

This scenario enables you to update the information relating to the computers (and their software) in a ServiceCenter 5.1 database using data obtained with LANDesk8.

ldsk\ld8sc6\ldsksc.scn



This scenario has the same functions as the scenario [ldsk\ld8sc51\ldsksc.scn](#) [page 386].

ldsk\ld8sc61\ldsksc.scn



This scenario has the same functions as the scenario [ldsk\ld8sc51\ldsksc.scn](#) [page 386].

ldskws\ld8ac50\LDStartTasks.scn



This scenario updates the following tables depending on the LANDesk information that is received:

- amESDPackage
- amESDDelivMethod

ldskws\ld8ac50\LDUpdateRepository.scn



This scenario creates scheduled tasks in LANDesk Software Distribution from information that is in the AssetCenter amESDTask scheduled tasks table.

ldskws\ld8ac50\LDUpdateTask.scn



This scenario updates scheduled tasks in LANDesk Software Distribution from information that is in the AssetCenter amESDTask scheduled tasks table.

Mercury Scenarios

This section briefly describes the out-of-the-box scenarios that use the Mercury connectors.

These scenarios are documented separately. Refer to the **ServiceCenter-MAM/BAC Integration Solution** guide.

merq\mam242sc61



Scenarios:

- **scmam-ci.scn**

This scenario is used to transfer data from ServiceCenter to MAM.

The scenario exports Configuration Items (devices) from ServiceCenter and imports them into MAM.

- **mamsc-ci-service.scn**

This scenario is used to transfer data from ServiceCenter to MAM.

This scenario exports Business Services and relationships between Business Services from MAM and imports them into ServiceCenter as devices

- **mamsc-ci.scn**

This scenario is used to transfer data from MAM to ServiceCenter.

This scenario exports Configuration Items (devices) and relationships between Configuration Items from MAM and imports them into ServiceCenter. Reconciliation for Configuration Items is done using the MAC address.

merq\bac51sc61



Scenarios:

- **scbac-ci.scn**

This scenario is used to transfer data from ServiceCenter to BAC.

This scenario exports Configuration Items (devices) from ServiceCenter and imports them into BAC.

- **scbac-changes.scn**

This scenario is used to transfer data from ServiceCenter to BAC.

This scenario exports changes and associated Configuration Items from ServiceCenter and imports them into BAC. This scenario exports the ServiceCenter URL that is associated with the change to allow the ServiceCenter Web page to be viewed in BAC.

- **bacsc-incident.scn**

This scenario is used to transfer data from BAC to ServiceCenter.

This scenario automatically opens a support ticket in ServiceCenter.

- **bacsc-ci.scn**

This scenario is used to transfer data from BAC to ServiceCenter.

This scenario exports Configuration Items (devices) and relationships between Configuration Items from BAC and imports them into ServiceCenter.

LDAP scenarios

This section briefly describes the out-of-the-box scenarios using the LDAP connector.

ldap\ldapac36\complete.scn



This scenario enables you to transfer data from an LDAP directory to the Departments and Employees table in an Asset Management application.

The following table lists the mapped source and destination document types.

Source	Destination
inetOrgPersonSrc	amEmplDeptDst

ldap\ldapac36\simple.scn



This scenario enables you to transfer data from an LDAP directory to the Departments and Employees table in an Asset Management application.

The following table lists the mapped source and destination document types.

Source	Destination
organizationalPersonSrc	amEmplDeptDst

ldap\ldapnote\names.scn



This scenario enables you to transfer data from an LDAP directory to a Lotus Notes database.

The following table lists the mapped source and destination document types.

Source	Destination
organizationalPersonSrc1	PersonDst

MQSeries scenarios

This section briefly describes the out-of-the-box scenarios using the MQSeries connector.

mqseries\acmq\mqtoprgn.scn



This scenario enables you to create, from MQSeries messages, records in the following tables of an Asset Management application:

- Cost centers table (amCostCenter)
- Companies table (amCompany)
- Purchase orders table (amPOOrder)
- Receiving slips table (amDeliv)

The following table lists the mapped source and destination document types.

Source	Destination
CostCenterSrc	amCostCenterDst
VendorSrc	amCompanyDst
ExtPOAckSrc	amPOOrderDst
ExtReceiptAckSrc	amDelivDst

mqseries\acmq\prgntomq.scn



This scenario enables you to:

- Create MQSeries messages from records in the following tables of an Asset Management application:
 - Purchase orders table (amPOOrder)
 - Receiving slips table (amDeliv)
- Update these same tables from the:
 - PEREGRINE.OUT queue
 - PEREGRINE.IN queue

The following table lists the mapped source and destination document types.

Source	Destination
amPOOrderSrc	PO
amDeliv	Receipt

mqseries\sc51mq\mqsc.scn



This scenario enables you to:

- Create records in ServiceCenter's **contact** file from the MQSeries messages in the PEREGRINE.IN queue.

- Create MQSeries messages in the PEREGRINE.OUT.PERSON queue from records in ServiceCenter's **contact** file.

The following table lists the mapped source and destination document types.

Source	Destination
External Contacts	Contacts from Exterior
Contacts of ServiceCenter	Contacts from ServiceCenter

mqseries\sc60mq\mqsc.scn



This scenario has the same functions as the scenario [mqseries\sc51mq\mqsc.scn](#) [page 391].

mqseries\sc61mq\mqsc.scn



This scenario has the same functions as the scenario [mqseries\sc51mq\mqsc.scn](#) [page 391].

NT Security scenarios

This section briefly describes the out-of-the-box scenarios using the NT Security connector.

ntsec\ntac36\addcpu.scn



This scenario enables you to import NT information on computers in NT domains specified in AssetCenter 3.6 database.

ntsec\ntac36\adduser.scn



This scenario enables you to import NT information obtained from departments and employees to NT domains specified in AssetCenter 3.6 database.

ntsec\ntac42\addcpu.scn



This scenario enables you to import NT information on computers in NT domains specified in AssetCenter 4.2 database.

ntsec\ntac42\adduser.scn



This scenario enables you to import NT information obtained from departments and employees to NT domains specified in an AssetCenter 4.2 database.

ntsec\ntac43\addcpu.scn



This scenario enables you to import NT information on computers in NT domains specified in AssetCenter 4.3 database.

ntsec\ntac43\adduser.scn



This scenario enables you to import NT information obtained from departments and employees to NT domains specified in an AssetCenter 4.3 database.

ntsec\ntac44\addcpu.scn



This scenario has the same functions as the scenario [ntsec\ntac43\addcpu.scn](#) [page 393].

ntsec\ntac44\adduser.scn



This scenario has the same functions as the scenario [ntsec\ntac43\adduser.scn](#) [page 393].

ntsec\ntac50\addcpu.scn



This scenario has the same functions as the [ntsec\ntac43\addcpu.scn](#) [page 393] scenario.

ntsec\ntac50\adduser.scn



This scenario has the same functions as the [ntsec\ntac43\adduser.scn](#) [page 393] scenario.

Peregrine Network Discovery scenarios

This section briefly describes the out-of-the-box scenarios using the Network Discovery connector.

pnd\pnd51ac43\pndac.scn



This scenario enables you to transfer the data relating to a Peregrine Network Discovery database to the Portfolio Items table in AssetCenter 4.3.

pnd\pnd51sc51\pndsc.scn



This scenario enables you to transfer the data relating to a Peregrine Network Discovery database to the Computers table in ServiceCenter 5.1.

pnd\pnd51sc60\pndsc.scn



This scenario has the same functions as the scenario [pnd\pnd51sc51\pndsc.scn](#) [page 395].

pnd\pn52dac43\pndac.scn

This scenario has the same functions as the scenario [pnd\pnd51ac43\pndac.scn](#) [page 394].



pnd\pn52dac44\pndac.scn



This scenario has the same functions as the scenario [pnd\pnd51ac43\pndac.scn](#) [page 394].

pnd\pnd52ac50\pndsc.scn



This scenario has the same functions as the [pnd\pnd51ac43\pndac.scn](#) [page 394] scenario.

pnd\pnd52sc51\pndsc.scn



This scenario has the same functions as the scenario [pnd\pnd51sc51\pndsc.scn](#) [page 395].

pnd\pnd52sc60\pndsc.scn



This scenario has the same functions as the scenario [pnd\pnd51sc51\pndsc.scn](#) [page 395].

pnd\pnd52sc61\pndsc.scn



This scenario has the same functions as the scenario [pnd\pnd51sc51\pndsc.scn](#) [page 395].

Peregrine Desktop Inventory scenarios

This section describes in brief the out-of-the-box scenarios using the Peregrine Desktop Inventory connector.

pdi\pdi7ac36\pdia.scn



This scenario enables you to update the information relating to the computers in the Computers table (acComputers) in an AssetCenter 3.6 database using data obtained with Peregrine Desktop Inventory.

pdi\pdi7ac42\pdia.scn



This scenario enables you to update the information relating to the computers in the Computers table (acComputers) in an AssetCenter 4.2 database using data obtained with Peregrine Desktop Inventory.

pdi\pdi7ac43\pdiaac.scn



This scenario enables you to update the information relating to the computers in the Computers table (acComputers) in an AssetCenter 4.3 database using data obtained with Peregrine Desktop Inventory.

pdi\pdi7sc51\pdisc.scn



This scenario enables you to update the information relating to the computers (and their software) in a ServiceCenter 5.1 database using data obtained with Peregrine Desktop Inventory.

Before executing this scenario, you must import the **pcsoft-uninst.un1** and **pcsoftware.un1** files into ServiceCenter. These files are provided with the installation of Connect-It (**datakit\sc** folder of your installation folder).

pdi\pdi7sc60\pdisc.scn

Ce scénario a les mêmes fonctions que le scénario [pdi\pdi7sc51\pdisc.scn](#) [page 397].



pdi\pdi8ac43\pdiaac.scn

Ce scénario a les mêmes fonctions que le scénario [pdi\pdi7ac43\pdiaac.scn](#) [page 397].



pdi\pdi8ac44\pdiaac.scn



This scenario has the same functions as the scenario [pdi\pdi7ac43\pdiaac.scn](#) [page 397].

Refer to section [Asset Management connector](#) [page 63], sub-section [Document type produced by the AssetManagement connector](#) [page 72] for the specific mapping issues in the scenario.

- ▶ Connect-It - Integration with an AssetCenter database.

pdi\pdi8ac44\pdiac-swnorm.scn

This scenario is dedicated to users of the Software Licenses module for AssetCenter and allows to reconcile software models.

For further information, refer to section [Document type produced by the AssetManagement connector](#) [page 72].

pdi\pdi8ac50\pdiac.scn



This scenario has the same functions as the [pdi\pdi8ac44\pdiac.scn](#) [page 397] scenario.

pdi\pdi8ac50\pdiac-swnorm.scn



This scenario has the same functions as the [pdi\pdi8ac44\pdiac-swnorm.scn](#) [page 398] scenario.

pdi\pdi8sc51\pdisc.scn



Ce scénario a les mêmes fonctions que le scénario [pdi\pdi7sc51\pdisc.scn](#) [page 397].

pdi\pdi8sc60\pdisc.scn



Ce scénario a les mêmes fonctions que le scénario [pdi\pdi7sc51\pdisc.scn](#) [page 397].

pdi\pdi8sc61\pdisc.scn



This scenario performs the same functions as the [pdi\pdi7sc51\pdisc.scn](#) [page 397].

RDS scenarios

This section briefly describes the out-of-the-box scenarios using the RDS connector.

rds\rds50sc51\rdssc.scn



This scenario enables you to prepare data from a ServiceCenter data in order to be used by BiPortal.

rds\rds52ac4\rdsac.scn



This scenario enables you to prepare data from an AssetCenter database in order to be used by BiPortal.

SAP IDOC scenarios

This section briefly describes the out-of-the-box scenarios using the SAP IDOC connector.

sap\sapac43\masterdata\lGetCompany.scn



This scenario enables you to transfer data from an SAP database to an AssetCenter database.

The data items transferred by this scenario are the following:

- Companies
- Locations

The following table lists the mapped source and destination document types.

Source	Destination
GetSupplier	amCompanyDst

sap\sapac43\masterdata\lGetCostcenter.scn



This scenario enables you to transfer data from an SAP database to an AssetCenter database.

The data items transferred by this scenario are the following:

- ◆ Cost centers

The following table lists the mapped source and destination document types.

Source	Destination
GetCostCenterSrc	amCostCenterDst

sap\sapac43\masterdata\lGetPeople.scn



This scenario enables you to transfer data from an SAP database to an AssetCenter database.

The data items transferred by this scenario are the following:

- Departments and employees
- Cost centers

The following table lists the mapped source and destination document types.

Source	Destination
HRMD_A01	amEmplDeptDst

sap\sapac43\purchaseprocess\IGetInvoice.scn



This scenario enables you to transfer data from an SAP database to an AssetCenter database.

The data items transferred by this scenario are the following:

- ◆ Invoices

The following table lists the mapped source and destination document types.

Source	Destination
GetInvoiceSrc	amInvoiceDst

sap\sapac43\purchaseprocess\IGetOrderNbr.scn



This scenario enables you to transfer data from an SAP database to an AssetCenter database.

sap\sapac44\masterdata



The following scenarios have the same functions as the scenarios for AssetCenter version 4.3.

- IGetCompany.scn
- IGetCostcenter.scn
- IGetPeople.scn

sap\sapac44\purchaseprocess



The following scenarios have the same functions as the scenarios for AssetCenter version 4.3.

- IGetOrderNbr.scn
- IGetInvoice.scn

sap\sapac50\masterdata



The following scenarios have the same functions as the scenarios for AssetCenter version 4.3.

- IGetCompany.scn
- IGetCostcenter.scn
- IGetPeople.scn

sap\sapac50\purchaseprocess



The following scenarios have the same functions as the scenarios for AssetCenter version 4.4.

- IGetOrderNbr.scn
- IGetInvoice.scn

SAP BAPI scenarios

This section briefly describes the out-of-the-box scenarios using the SAP BAPI connector.

sap\sapac43\purchaseprocess\BAPI_CREATE_REQUEST.scn



This scenario enables you to transfer data from an SAP database to an AssetCenter database.

The data items transferred by this scenario are the following:

- Requests
- Suppliers
- Employees

The following table lists the mapped source and destination document types.

Source	Destination
amPOrder	NewRequestDst

sap\sapac43\purchaseprocess\BAPI_CREATE_PO.scn



This scenario enables you to transfer data from an SAP database to an AssetCenter database and vice-versa.

The data items transferred by this scenario are the following:

- Purchase orders
- Purchase order lines

The following table lists the mapped source and destination document types.

Source	Destination
amPOrderSrc	NewPurchaseOrderDst
ProcessReportNewPurchaseOrderSrc	amPOrderDst

sap\sapac43\purchaseprocess\BAPI_RECEIVE.scn



This scenario enables you to transfer data from an SAP database to an AssetCenter database and vice-versa.

The data items transferred by this scenario are the following:

- Receiving slips
- Receiving lines

The following table lists the mapped source and destination document types.

Source	Destination
amReceiptSrc2	BAPI_GOODSMVT_CREATEDst2

Source	Destination
ProcessReportBAPI_GOODSMVT_CREATE	Src1 amReceiptDst2

sap\sapac44\purchaseprocess\BAPI_CREATE_REQUEST.scn



This scenario has the same functions as the scenario [sap\sapac43\purchaseprocess\BAPI_CREATE_REQUEST.scn](#) [page 402].

sap\sapac44\purchaseprocess\BAPI_CREATE_PO.scn



This scenario has the same functions as the scenario [sap\sapac43\purchaseprocess\BAPI_CREATE_PO.scn](#) [page 403].

sap\sapac44\purchaseprocess\BAPI_RECEIVE.scn



This scenario has the same functions as the scenario [sap\sapac43\purchaseprocess\BAPI_RECEIVE.scn](#) [page 403].

sap\sapac44\purchaseprocess\BAPI_CREATE_REQUEST.scn



This scenario has the same functions as the [sap\sapac43\purchaseprocess\BAPI_CREATE_REQUEST.scn](#) [page 402] scenario.

sap\sapac44\purchaseprocess\BAPI_CREATE_PO.scn



This scenario has the same functions as the [sap\sapac43\purchaseprocess\BAPI_CREATE_PO.scn](#) [page 403] scenario.

sap\sapac44\purchaseprocess\BAPI_RECEIVE.scn



This scenario has the same functions as the [sap\sapac43\purchaseprocess\BAPI_RECEIVE.scn \[page 403\]](#) scenario.

SMS scenarios

This section briefly describes the out-of-the-box scenarios using the SMS 2.x and 2003 connectors.

sms\sms2ac43\smsac.scn



This scenario enables you to transfer information from an SMS database (IT portfolio) to the amComputer table of an AssetCenter 4.3 database.

Source	Destination
System_DATASrc	amComputerDst

sms\sms2003ac43\smsac.scn



This scenario enables you to transfer information from an SMS database (IT portfolio) to the amComputer table of an AssetCenter 4.3 database.

sms\sms2003ac44\smsac.scn



This scenario has the same functions as the [sms\sms2003ac43\smsac.scn \[page 405\]](#) scenario.

sms\sms2003ac50\smsac.scn



This scenario has the same functions as the [sms\sms2003ac43\smsac.scn](#) [page 405] scenario.

sms\sms2003sc51\smssc.scn



Before executing this scenario, you must import the **pcsoft-uninst.un1** and **pcsoftware.un1** files into ServiceCenter. These files are provided with the installation of Connect-It (**datakit\sc** folder of your installation folder).

This scenario enables you to update the information relating to the computers (and their software) in a ServiceCenter 5.1 database using data obtained with SMS2003.

sms\sms2003sc6\smssc.scn



This scenario has the same functions as the scenario [sms\sms2003sc51\smssc.scn](#) [page 406].

sms\sms2003sc61\smssc.scn



This scenario has the same functions as the scenario [sms\sms2003sc51\smssc.scn](#) [page 406].

ServiceCenter scenarios

This section briefly describes the out-of-the-box scenarios using the ServiceCenter connector.

scac\sc3ac36\scac.scn



This replication scenario enables you to create or update the data from the databases of an Asset Management 3.6 application to a ServiceCenter 3.0.

Configuration of the Asset Management application

This section enables you to configure your Asset Management application in order to use the out-of-the-box scenario.

Amdb.ini file

This file contains the list of database connections created on a computer on which the Asset Management application is installed. This file is created automatically when the Asset Management application is installed. It is located in the Windows system folder.

Example:

```
C:\WINNT\amdb.ini.
```

When you perform the configuration of your database, the Asset Management connector reads this file and proposes the list of available connections. If your computer does not have access to an Asset Management application, you cannot use the Asset Management connector.

Features tables

Features allow you to extend the Asset Management data-model.

In Connect-It, features appear as fields or collections in the document types published by the Asset Management connector.

Example :

For example: In the **amAsset** document type, the version of DOS used by an asset (a computer, in this case) appears in the **fv_DOSVersion** field.

For information on the import process in AssetCenter, refer to the "Reference Guide: Administration and advanced use" in the "Importing data" chapter.

Mandatory fields in an Asset Management database

In an Asset Management application, a given field or link may be mandatory by default or have been customized this way by the administrator of the Asset Management application.

In the case of reconciliation, each structure published by the Asset Management application corresponds to a record. If an element in this structure is a mandatory field and is not populated, the structure is rejected.

Example :

For example: In the Assets table, the **AssetTag** field is mandatory. If the element that represents it in a document type corresponding to the Assets table is not populated, the entire structure is rejected.

API

To be able to use the Asset Management connector correctly, you must perform a full installation of AssetCenter, AssetCenter Cable and Circuit, or InfraCenter or a customized installation in which the "API" package is selected.

Configuring ServiceCenter

This section enables you to configure ServiceCenter in order to use the out-of-the-box scenario.

Creation of input events

In order for your scenario to function correctly, you must import one of the following UNL files into ServiceCenter, which can be found in the **datakit\sc** folder of your Connect-It installation folder:

- **cmdb.unl**,
- **pcsoft-uninstall.unl**
- **pcsoftware.unl**

These files enable you to update events linked to the Configuration Items (C.I.), such as ICM-type events. They also modify the validity rules applied to the model and modelvendor files, as well as clean the standard ServiceCenter datakit..

Warning:

These files must not be imported to the ServiceCenter database that you are running. You can request these files from Peregrine Systems client support.

Input event synchronization

Input events in ServiceCenter are processed either synchronously or asynchronously. In asynchronous mode, ServiceCenter defines how the event queue is managed. In synchronous mode, the events are processed as soon as they are inserted in the ServiceCenter event queue.

When developing your scenarios, use the synchronous mode if you want to verify immediately that your scenario is working.

Since Connect-It works in synchronous mode, selecting the asynchronous processing mode prevents you from immediately verifying in ServiceCenter that the data has been processed correctly: A warning message is issued to indicate that the document is pending processing.

In synchronous mode, it indicates whether a document has been processed successfully or rejected.

Context of application

The out-of-the-box scenario is a demonstration scenario replicates data between the data kits provided with the Asset Management application and ServiceCenter.

Warning:

The current scenario will not, under any circumstances, be applicable to a ServiceCenter application installed in production. It is only provided and described in this document as an example.

This scenario is restricted to the mapping of data linked to the inventory: assets, products, companies, locations, employees, and departments.

Because these data kits were not designed especially for data replication between ServiceCenter and an Asset Management application, they contain entirely different data.

In an actual company, the list of employees recorded in the ServiceCenter database must correspond to the list recorded in the Asset Management database.

The main preliminary task, then, is to clean and fit the differing data so that it has the same naming policy, keys, etc.

The first step in the out-of-the-box scenario, however, involves a massive import of data from ServiceCenter to an Asset Management application and vice versa.

It describes some practices and solves some issues that would apply to a real replication between ServiceCenter and an Asset Management application.

These simple rules should be applied whenever possible:

- Define a unique repository.
To avoid conflicts, the data should be created, modified and deleted in a unique repository, whether that be in ServiceCenter or in an Asset Management application. The replication process would, thus, only perform simple updates in the other system. The unique repository need not be defined globally, but rather by specific functions. For instance, AssetCenter may be defined as the repository for the assets and ServiceCenter for the tickets. As a rule of thumb, the repository should be defined in the system where the data is the most accurate and is usually processed. If having a unique repository appears to be a drastic restriction, then one of the systems should be limited to applying minor updates to the data, but not to create new records or modify IDs.
- Keep track of the system where the record has been created.
It is useful to keep track of such information in order to verify that a replication has been properly performed or to apply a specific mapping, depending on the system where the data has been created.
- Unify data between an Asset Management application and ServiceCenter.
The data should be processed as closely as possible between the two systems to simplify the mapping. For instance, certain categories associated by default to an Asset Management application are completely different from those associated to the **model** in ServiceCenter. A proper mapping would require that the same categories are defined.

The difference between the reconciliation and replication of data.

There are some major differences between the reconciliation of data and the replication of data.

Reconciliation

Reconciliation is the integration of input data coming from a source application that is considered more up-to-date than the already existing data in the destination application.

This mechanism is based on the following question:

"Does the information that I would like to reconcile already exist in the target system?"

- If the answer is "no", the input data is inserted.
- If the answer is "yes", the existing data is updated according to the information contained in the input data.

When the destination system is empty, this process is no longer a reconciliation, but an import.

Replication

In a replication process, the input data is no longer considered to be the most up-to-date.

The last modification date of both input and existing data must now be compared with the last replication date in order to figure out which data source is the most up-to-date.

A replication conflict is then raised when both the input and the existing data have been modified since the last replication.

There is no sure-fire way to determine which data source contains the most up-to-date information. In this case, the scenario designer must set up the behavior of the replication engine.

This might be:

- Rejecting the input data.
- Choosing to consider the input data as more up-to-date.
- Trusting the existing data.

Managing the replication of data in Connect-It

When the ServiceCenter connector consumes a document, it doesn't create a record directly in the ServiceCenter database. Instead, it creates an input event and adds it in the **event queue entry**.

As a result, the reconciliation (as well as the replication) process is internally handled by the ServiceCenter server when the input event is computed.

The person in charge of running scenarios must solve the replication issues in the ServiceCenter server.

Unlike the ServiceCenter connector, the Asset Management connector can interact directly with the underlying database. In this way, the reconciliation (as well as the replication) process is fully handled by Connect-It.

The following sections explain:

- How a replication conflict is detected on the Asset Management side.
- What actions can be performed in the case of a conflict.
- How dates enabling data replication are compared.

Detecting a conflict

In Asset Management, the replication mechanism is triggered as soon as the **modification date** (Connect-It SQL name: **dtLastModif**) is detected in the document to consume.

The replication mechanism will run automatically any time this field is found in a structure or in a collection.

In the scenario, this field must be mapped to the ServiceCenter **sysmodtime** field.

In this scenario, if a document is produced by ServiceCenter, it is because it has been modified since the last replication (or because it did not exist at that time). As a result, the next question to consider is:

"Have the corresponding record(s) been modified on the Asset Management side since the last replication?"

- If the answer is "yes", there is a replication conflict.
- If the answer is "no" (this may be the case if the data doesn't yet exist in the Asset Management database), the data will be updated (or inserted).

Taking action in the case of a conflict

An identified replication conflict encompasses the structure or the collection item that is being processed.

There are several possible actions for the records in question:

- Reject it.
In this case, a field error is generated for the **dtLastModif** field, and the data in the Asset Management application is not modified.
- Warning concerning field.
In this case, a field warning is generated for the **dtLastModif** field, and the data in Asset Management is updated.
- Overwrite it.
In this case, the data in Asset Management is overwritten with the data from ServiceCenter, and no warnings are logged.

This parameter can be set in the consumption directive tabs of the Asset Management connector.

Comparing the dates

The comparison of dates in the replication process demands a high level of accuracy which depends on:

- Time zones
- Server time differences

Time zones

A date based on GMT+1 time zone, and another based on GMT-7 cannot be directly compared: A transformation needs to be performed.

Inside the Connect-It engine, dates are always stored and manipulated as GMT. It is each connector's responsibility to provide the Connect-It engine with GMT dates.

Of course, dates are displayed according to the local settings of the computer on which Connect-It is run.

Server time difference

In an "ideal" world, every server would be perfectly synchronized with a single time reference. In the real world, some servers run behind, others run ahead.

With Connect-It, the time-based data provided by a connector can be corrected according to a given delay (which can be negative).

As a result, the server's delay is hidden in >Connect-It.

scac\sc51ac43\scac.scn - scac\sc51ac43\acsc.scn



Before executing this scenario, you must import the file **acsc.un1**, which creates the required events on importing the data in ServiceCenter.

The two scenarios (**scac.scn** and **acsc.scn**) replicate the inventory data and the associated data (contacts, locations, models, etc.) between a ServiceCenter 5.1 database and an AssetCenter 4.3 database.



Note:

We recommend against replicating data in both directions because of differences in the database schemas. We recommend managing (authorization to update, insert and/or delete) each functional category in one or the other applications and to enable only the corresponding mappings for each scenario. For example, updating Employees and Locations in AssetCenter, and updating inventory data in ServiceCenter, implies disabling the replication of Employees/Locations in the **scac.scn** scenario and disabling the replication of the inventory data in the **acsc.scn** scenario.

scac\sc51ac44\scac.scn - scac\sc51ac44\acsc.scn



This scenario has the same function as the scenario [scac\sc51ac43\scac.scn - scac\sc51ac43\acsc.scn](#) [page 413].

scac\sc6ac43\scac.scn - scac\sc6ac43\acsc.scn



This scenario has the same function as the scenario [scac\sc51ac43\scac.scn - scac\sc51ac43\acsc.scn](#) [page 413].

In this case, the replication is made between a version 6.0 of ServiceCenter and a version 4.3 of the Asset Management application.

scac\sc6ac44\scac.scn - scac\sc6ac44\acsc.scn



This scenario has the same function as the scenario [scac\sc51ac43\scac.scn - scac\sc51ac43\acsc.scn](#) [page 413].

scac\sc61ac44\



- acsc-ci.scn

This scenario enables the integration of AssetCenter Configuration Items into ServiceCenter.

- `acsc-incident.scn`

This scenario is used to create an incident in ServiceCenter when the reconciliation proposal is done in AssetCenter.

- `scac-ci.scn`

This scenario enables the integration of ServiceCenter Configuration Items into AssetCenter.

- `scac-wo.scn`

This scenario is used to create a work order in AssetCenter that corresponds to a change request in ServiceCenter.

`scac\sc61ac50\`

Scenarios in this folder have the same functions as scenarios in the `scac\sc61ac44\` [page 414] folder.

`scauto\scacfg\scacfg.scn`



The following table lists the mapped source and destination document types.

Source	Destination
Output Event Type	InputEventTypes
Input Event Type	OutputEventTypes
EventTypes	SCAutoConfiguration

`scauto\sca-sc\sca-sc.scn`



This scenario enables you to create a configuration file (`scautoconfiguration.xml`) for the SCAuto listening connector. This file contains the definition of a ServiceCenter database's elements.

sc\sc51mail\scincident-mail.scn



This scenario sends an e-mail when an incident ticket is created. The data in the **probsummary** table is read and an e-mail is automatically sent to the assignee, using the e-mail address contained in the **operator** record.

The e-mail contains the incident description and the file or files linked to the incident ticket are sent as attachments.

This scenario is scheduled to be executed every hour from 8 AM through 6 PM, and inserts or updates the records.

sc\sc60mail\scincident-mail.scn



This scenario has the same function as the scenario [sc\sc51mail\scincident-mail.scn](#) [page 416].

sc\sc61mail\scincident-mail.scn



This scenario has the same function as the scenario [sc\sc51mail\scincident-mail.scn](#) [page 416].

ServiceCenter Web Service scenarios

This section briefly describes the out-of-the-box scenarios using the ServiceCenter Web Service connector.

scws\scws61\scws-im.scn




This scenario creates or updates incident management records.

Mandatory fields

Some fields for produced or consumed document types are mandatory. They are defined as mandatory according to the exposed WSDL.

For this scenario in creation mode, the following complex elements are mandatory:

- createIncidentRequest
- model
- instance
- dkeys

The  icon indicates that they are mandatory.

If a field is not populated with a value, a warning is generated in the document log.

 **Note:**

This scenario does not perform any updates.

Strict error handling

When the Web service sends a reply with a Failure status, this option enables to record an error for the consumed document and stop production.

This option is not activated by default.

scws\scws61\scws-cm.scn



This scenario updates configuration management records.

This scenario performs two mappings. The first performs insertions (RetrieveComputer-CreateComputerDst), and the second performs updates (RetrieveComputer-UpdateComputerDst1). First the scenario attempts to recover an ID of an existing computer, and creates one if not found. Then it performs the update.

In the RetrieveComputer-CreateComputerDst mapping, the insertion or the update is defined using the following mapping:

```
If [RetrieveComputerResponse.returnValue] = 0 Then
PifIgnoreDocumentMapping
End If
```

In this mapping, no insertion is performed if the value returned by the Web service indicates that a computer ID exists (value equal to zero). If the value returned by the Web service equals 1, indicating a failure to return a computer ID value, an insertion is made and a computer ID is created with an empty value. In the RetrieveComputer-UpdateComputerDst1 mapping, the insertion or the update is defined via the following mapping:

```
If [RetrieveComputerResponse.returnValue] <> 0 Then
  PifIgnoreDocumentMapping
End If
```

In this mapping, no update is performed if the value returned by the Web service indicates a failure to return a computer ID, i.e. a computer does not exist. Otherwise the update is performed.

Mandatory fields

Some fields, for produced or consumed document types, are mandatory. Those fields were defined as mandatory according to the exposed WSDL. The mandatory nature of those fields changes according to the mode used (creation or update). If a field is not populated with a value, a warning is generated in the document log.

Strict error handling

When a the Web service sends reply with a Failure status, this option enables to save an error for the consumed document and stop production.

TS Census scenarios

This section briefly describes the out-of-the-box scenarios using the TS Census connectors.

tsc\tsc2ac36\tscac.scn



This scenario enables you to transfer data from a TSCensus 2.x databases to an AssetCenter 3.6 database.

The following data is transferred:

- Computers
- Software
- Devices

tsc\tsc2ac43\tscac.scn



This scenario enables you to transfer data a TSCensus 2.x database to an AssetCenter 4.3 database.

tsc\tsc3ac36\tscac.scn



This scenario enables you to transfer data from a TSCensus 3.0 database to an AssetCenter 3.6 database.

tsc\tsc3ac43\tscac.scn



This scenario enables you to transfer data from a TSCensus 3.0 database to an AssetCenter 4.3 database.

tsc\tsc3ac44\tscac.scn



This scenario has the same functions as the [tsc\tsc3ac43\tscac.scn](#) [page 419] scenario.

tsc\tsc3ac50\tscac.scn



This scenario has the same functions as the [tsc\tsc3ac43\tscac.scn](#) [page 419] scenario.

tsc\tsc3sc51\tscsc.scn



Before executing this scenario, you must import the **pcsoft-uninst.un1** and **pcsoftware.un1** files into ServiceCenter. These files are provided with the installation of Connect-It (**datakit\sc** folder of your installation folder).

This scenario enables you to update the information relating to the computers (and their software) in a ServiceCenter 5.1 database using data obtained with TS Census.

tsc\tsc3sc6\tscsc.scn



This scenario has the same functions as the scenario [tsc\tsc3sc51\tscsc.scn](#) [page 420].

tsc\tsc3sc61\tscsc.scn



This scenario has the same functions as the scenario [tsc\tsc3sc51\tscsc.scn](#) [page 420].

Tivoli scenarios

This section briefly describes the scenarios using the Tivoli Inventory Management connectors.

tivoli\tim\tim4ac42\timac.scn



This scenario enables you to transfer data from a Tivoli Inventory Management 4.0 database to an AssetCenter 4.2 database.

The following data is transferred:

- Computers
- Software

tivoli\tim\tim4ac43\timac.scn



This scenario enables you to transfer data from a Tivoli Inventory Management 4.0 database to an AssetCenter 4.3 database.

The following data is transferred:

- Computers
- Software

tivoli\tim\tim4ac44\timac.scn



This scenario has the same functions as the [tivoli\tim\tim4ac43\timac.scn](#) [page 421] scenario.

tivoli\tim\tim4ac50\timac.scn



This scenario has the same functions as the [tivoli\tim\tim4ac43\timac.scn](#) [page 421] scenario.

tivoli\tcm\tcm42sc51\tcm.scn



Before executing this scenario, you must import the **pcsoft-uninst.unl** and **pcsoftware.unl** files into ServiceCenter. These files are provided with the installation of Connect-It (**datakit\sc** folder of your installation folder).

This scenario enables you to transfer data from a Tivoli Inventory Management 4.2 database to an ServiceCenter 5.1 application.

The following data is transferred:

- Computers

- Software

tivoli\tcm\tcm42sc60\tcmisc.scn



This scenario has the same functions as the scenario [tivoli\tcm\tcm42sc51\tcmisc.scn](#) [page 421].

tivoli\tcm\tcm42sc61\tcmisc.scn



This scenario has the same functions as the scenario [tivoli\tcm\tcm42sc51\tcmisc.scn](#) [page 421].

tivoli\tcm\tcm42ac43\tcmac.scn



This scenario enables you to transfer data from a Tivoli Configuration Manager 4.2 database to an AssetCenter 4.3 database.

The following data is transferred:

- Computers
- Software

tivoli\tcm\tcm42ac44\tcmac.scn



This scenario has the same functions as the [tivoli\tcm\tcm42ac43\tcmac.scn](#) [page 422] scenario.

tivoli\tcm\tcm42ac44\tcmac-swnorm.scn



This scenario is dedicated to users of the Software Licenses module for AssetCenter and allows to reconcile software models.

For further information, refer to section [Document type produced by the AssetManagement connector](#) [page 72].

tivoli\tcm\tcm42ac50\tcmac.scn



This scenario has the same functions as the [tivoli\tcm\tcm42ac44\tcmac.scn](#) [page 422] scenario.

tivoli\tcm\tcm42ac50\tcmac-swnorm.scn



This scenario has the same functions as the [tivoli\tcm\tcm42ac44\tcmac-swnorm.scn](#) [page 422] scenario.

tivoli\tcm_sd\tcmsd42ac43

The scenarios contained in this folder are documented separately.

tivoli\tcm_sd\tcmsd42ac44

The scenarios contained in this folder are documented separately.

tivoli\tcm_sd\tcmsd421ac44

The scenarios contained in this folder are documented separately.

tivoli\tcm_sd\tcmsd421ac50

The scenarios contained in this folder are documented separately.

Tivoli Enterprise Data Warehouse scenarios

The out-of-the-box scenarios are the following:

- Tivoli\tedw_sc51tedw\sc_cm3r_tedw.scn
- Tivoli\tedw_sc51tedw\sc_incidents_tedw.scn
- Tivoli\tedw_sc51tedw\sc_probsummary_tedw.scn

These scenarios are documented separately.

Unicenter AMO scenarios

This section briefly describes the out-of-the-box scenarios using the UniCenter AMO connector.

amo\amo3ac43\amoac.scn



This scenario enables you to transfer data from a Unicenter AMO 3 database to an Asset Management application.

The following data is transferred:

- Computers
- Software
- Associated devices

amo\amo3ac44\amoac.scn



This scenario has the same functions as the scenario [amo\amo3ac43\amoac.scn](#) [page 424].

amo\amo4ac44\amoac.scn



This scenario has the same functions as the scenario [amo\amo3ac43\amoac.scn](#) [page 424].

amo\amo4ac50\amoac.scn



This scenario has the same functions as the scenario [amo\amo3ac43\amoac.scn](#) [page 424].

Web Service scenarios

This section briefly describes the out-of-the-box scenarios using the Web Service connector.

ws/wsac43/currency.scn



This scenario enables you to update the exchange rate already defined in the **amCurRate** table of your Asset Management application.

The WSDL address of the Web service is

<http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl>.

Winpark Actima scenarios

This section briefly describes the out-of-the-box scenarios using the Winpark Actima connector.

wpk\wpkac43\wpkac.scn



This scenario enables you to transfer data from a Winpark Actima database to an AssetCenter 4.3 database.

The following data is transferred:

- Departments and employees
- Companies
- Contracts
- Cost centers
- Locations
- Computers
- Devices
- Models
- Telephones
- Network information and other non-specific assets.

wpk\wpkac44\wpkac.scn



This scenario has the same functions as the [wpk\wpkac43\wpkac.scn](#) [page 426] scenario.

wpk\wpkac50\wpkac.scn



This scenario has the same functions as the [wpk\wpkac43\wpkac.scn](#) [page 426] scenario.



A | AQL queries

APPENDIX

This chapter explains how to write AQL queries:

Presentation

This section presents AQL and lists the places in the software where you can use queries.

AQL

AQL ("Advanced Query Language") is the query language used by AssetCenter to access the data in the AssetCenter database. It is comparable to SQL. AQL is automatically translated into SQL by the database engine when it's being used for queries.

 Note:

We recommend that you have some knowledge of SQL and a good understanding of databases before using AQL.

Advantages of AQL

AQL is better suited than SQL for writing queries involving the AssetCenter database. This is due to the following reasons:

Independence from database

The various database engines supported by AssetCenter all use differing versions of SQL, which are incompatible with each other. AQL is independent of the database engine used.

Consequently, if you write AQL queries and migrate from one database engine to another, the queries will still work with the new database engine.

For example, AQL uses the same set of functions, regardless of the database engine used.

Thus, the **Substring** function in AQL is equivalent to **Substr** under Oracle for WorkGroupsSQL and **Substring** under Microsoft SQL Server SQL.

Generation of optimized SQL code

AQL generates SQL code optimized according to the database engine used.

This is especially visible when using indexes. For example, to search the full name of models by forcing the use of the indexes **Model ID(Model_IDModelId)** and **Full name (FullName)** you will write the following AQL query:

```
SELECT FIRST_ROWS lModelId, FullName FROM amModel
```

The SQL code that is generated will be different depending on the target database engine and optimized according to this engine. The equivalent Oracle SQL code will be:

```
SELECT /*+ FIRST_ROWS INDEX_ASC(M1 Model_lModelId) */ M1.lModelId, M1.FullName FROM amModel M1
```

The Microsoft SQL Server or Sybase SQL Server code will be:

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

The IBM DB2 code will be:

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

Simplified access to the AssetCenter database

AQL simplifies the management of links and joins. This greatly facilitates access to the database when writing queries as compared to using SQL directly.

In addition, AQL simplifies access to features, allowing you to use them as direct fields in their related tables.

AQL also facilitates the utilization of calculated fields.

Specifics of AQL compared to SQL

AQL does not support DDL ("Data Definition Language") orders.

AQL comprises extensions that simplify the handling of joins, features and calculated fields.

Warning:

You should never write to the AssetCenter database directly using SQL statements.

Queries in AssetCenter

Queries enable you to combine criteria concerning information in a table or in linked tables.

You can use queries to:

- Create filters on record lists. In this case, the queries are, in general, simple and based on the "Where" clause.
- Define views.
- Define export conditions in the export module.
- Create Crystal Reports.
- Create wizards.
- When you use AssetCenter APIs.
- If AssetCenter is used as a DDE server.

The queries are written in AQL ("Advanced Query Language"): It is an internal language used by AssetCenter to access the data in the AssetCenter database.

AssetCenter has a query editor that helps write queries:

- Either indirectly by using the graphic interface.
- Or by writing the query directly in AQL.

 **Warning:**

In order to be illustrative of the capabilities of AQL, the examples given later use all of the AQL syntax. The SELECT, WHERE, and FROM clauses, in particular, are explicitly shown. Certain functions, such as query filters (where the user only defines the WHERE clause in the AQL query) or the expression builder greatly simplify the creation of queries for the user (certain clauses are not visible). These examples cannot be used for these functions.

Recommendations for writing AQL queries

We recommend reading this section before starting to write queries in AQL.

This section deals with:

- Notation specific to AQL.
- Particularities specific to AQL and the AssetCenter database that condition the best way to write queries.

The sections entitled [AQL syntax](#) [page 447] and [AQL function references](#) [page 460] complement this section.

 **Warning:**

SQL names of fields, links and tables of the AssetCenter database are used to write AQL queries. Refer to the **Database.txt** file describing the database structure for an exhaustive list of these names.

This file is located in the following folder: **[AssetCenter installation folder]/doc/infos**

Presentation of AQL joins.

Definition

A join consolidates multiple data tables in a single query.

AQL joins

The AssetCenter data model defines not only fields and tables, but it also defines the links between tables. This enables you to automate the generation of join clauses in AQL.

AQL links are expressed as:

```
Link[.Link[.Field]]
```

By simplifying the joins in such a manner, AQL also simplifies the creation of the majority of queries used for the database.

Example

The following query, written in AQL, returns for each model:

- Its ID (**lModelId**).
- Its **Full name (FullName)**.
- The **Name** of the table linked to the brands (**amBrand**).

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

Here is the same query written in SQL Oracle or Microsoft SQL Server:

```
SELECT M1.lModelId, M1.FullName, B2.Name FROM amModel M1, amBrand B2 WHERE M1.lBrandId=B2.lBrandId
```

The two joins between the **Models** table and the **Brands** table (**amModel** and **amBrand**, respectively) are automatically generated in AQL. Using AssetCenter's interface query editor, you just need to click the hierarchic list on the fields of the selected table - or the fields of the linked tables - to generate the corresponding AQL code.

 **Note:**

On all systems except Oracle and DB2, the number of outer joins is limited to 1. Under Microsoft SQL Server 7 and MSSQL 2000, you can modify the **amdb.ini** file to get round possible query-execution problems. Use the following instruction to modify this file in the detail of your connection:

```
useSQL92Join=1
```

Reason for and usefulness of primary key 0 records.

Primary key "0" records

The data model of the AssetCenter database has certain particularities:

- The primary and foreign keys of each table are numeric (32-bit integer).
- A foreign key that does not point to a record is set to "0" (and not "NULL").
- Each table has an empty record whose primary key is set to "0".

Usefulness

With these primary key "0" records, the results of a query using a non outer join between two given tables, A and B, can include records from table A which are not linked to any real record in table B (link not populated). These are records in table A, which are linked to the primary key "0" record in table B.

Example:

The following query, written in AQL, returns for each portfolio item's asset tag, the name of its user and its supervisor:

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

A portfolio item that is not assigned to a user and/or supervisor appears in the results of the query. At database level, such an asset is linked to the primary key "0" record in the Departments and employees table.

Reason for these specificities

This section explains why primary key "0" records are used, whereas a query using an external SQL join can select records in table A that are not linked to a record in table B.

Primary key "0" records enable you to compensate for the fact that certain RDBMS do not handle multiple outer-joins: Using primary key "0" records, SQL queries generated from AQL queries do not use outer joins.

Example:

The AQL query below searches for each portfolio item, its asset tag and name of location of its user. The results will include assets that do not have a user and assets whose users do not have a location.

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

If the generated SQL code used the outer joins of the DBMS, the SQL code generated for Sybase SQL Server would be:


```
SELECT a.AssetTag, l.name FROM amPortfolio a, amEmplDept e, amLocation l W
HERE a.lUserId *= e.lEmplDeptId AND e.lLocaId *= l.lLocaId
```

This code is not supported by Sybase SQL Server since it uses several outer joins one after another.

However, because there is a primary-key ("0") record in the Departments and Employees table and in the Locations table, you do not need to call on external SQL joins. AssetCenter therefore generates an SQL query using normal (non external) joins:

```
SELECT l.name FROM amPortfolio a, amEmplDept e, amLocation l WHERE a.lUser
Id = e.lEmplDeptId AND e.lLocaId = l.lLocaId
```

This query gives the expected results, since the **User** and **Location** links still point to a record in the table of departments and employees or locations (they point to the primary key "0" record if the link is not populated).

Consequences

- ◆ It is important to take these records into account in the queries that you write, especially when using aggregate functions.

Example:

```
SELECT count(AssetTag) FROM amPortfolio
```

If you execute the above query, which counts the number of assets in the table of assets, the primary key "0" record is taken into account in the results. You therefore need to decrease the results by 1 in order to obtain the real number of assets in the database.

- ◆ It is rarely necessary to have to generate outer joins at DBMS level.

 Note:

Note: If you really want to manage outer joins at DBMS level, use the "=" and "*=" AQL operators.

Usage of NULL.

AssetCenter uses the NULL value of the DBMS in only two instances:

- For an empty "text" type field.
- For a non populated "date" or "date+time" type field.

AQL allows you to use several equivalent syntaxes, as shown below. It converts them to the equivalent valid SQL code of the database engine.

For empty "Text" type fields, you can use any of the following syntaxes, since the NULL value will be stored in the database:

WHERE <text field> = NULL

WHERE <text field> IS NULL

WHERE <text field> = "

For non-populated "date" or "date+time" type fields, you can use any of the following syntaxes, since the NULL value will be stored in the database:

WHERE <date or date+time field> = NULL

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

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

 Note:

When a "Numeric" type field is not populated (its value is NULL), Connect-It sets its value to "0". Similarly, the absence of a link is comes out as "Link = 0" or "foreign key = 0". Example: "Location=0" or "lLocald=0".

Self

"Self" is an expression that is equivalent to the description string on the table to which it is applied.

Using "Self" enables you to simplify the queries while taking into account the customizations made to the AssetCenter database.

Example:

If the description string of the table of departments and employees is:

```
[Name], [FirstName], ([Phone])
```

The AQL query:

```
SELECT self FROM amEmplDept
```

Is equivalent to:

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

CurrentUser

"CurrentUser" enables you to write queries that depend on the person connected to the database.

"CurrentUser" can be used as an expression, for example in a query, or as a link. You have to enter this expression manually as it is not offered by the query editor.

Used as "expression"

Example: We are looking for all the portfolio items used by the employee connected to the database.

```
SELECT lPortfolioItemId FROM amPortfolio WHERE User = CurrentUser
```

Used as "link"

"CurrentUser" may be considered as a link starting in all tables and pointing to the record in the table of departments and employees corresponding to the current user.

- With the form "CurrentUser", this function points to the record corresponding to the current user.
- With the form "CurrentUser.Field", this function returns the value of the field for the current user.

Example: When an action is triggered by the connected user, it is possible to contextually trigger another messaging type action, which automatically sends a warning message to the connected user. You just need to populate the detail of the action as follows:

The screenshot shows a configuration window for a messaging action. It has four tabs: Description (selected), Messaging, Features, and History. The fields are as follows:

- Referenced object: [Empty dropdown]
- Priority: Normal [Dropdown]
- Acknowledgment
- To: [CurrentUser.E Mail] [Search icon]
- Cc: [Empty field] [Search icon]
- Bcc: [Empty field] [Search icon]
- Subject: Confirmation message [Search icon]
- Message: You have executed an action [Search icon]

System itemized lists

If an AQL query uses a system itemized list, you must use the values that are stored in the database and not those which are displayed on screen.

Example:

The following query selects those contracts whose **Type** field (SQL name: seType) is set to **Master lease**:

```
SELECT Self FROM amContract WHERE seType = 1
```

The **Type** field (SQL name: seType) is a system itemized list. The values stored in the database are:

- 0 for an **Other** type contract
- 1 for a **Master lease** type contract
- 2 for a **Lease schedule** type contract
- 3 for an **Insurance** type contract
- 4 for a **Maintenance** type contract



Note:

The values of system itemized list can be found via AssetCenter Database Administrator, or the **database.txt** file, which describes the structure of the database.

This file is located in the following folder: **[AssetCenter installation folder]/doc/infos**

Hierarchic tables

All hierarchic tables contain:

- A "FullName" field.
- A "sLvl" field.

"FullName" fields

For each record in a hierarchic table, the "FullName" field stores the value of a field of the record, preceded by a tree structure constituted by the values of the fields of parent records until root level.

Values are separated by the "/" character without spaces. This character also appears at the start and at the end of the tree-structure.

Examples:

- ◆ For the table of assets, the "FullName" field stores the Asset Tag of the asset preceded by the Asset Tag of its parent asset, that in turn preceded by the Asset Tag of its parent asset, and so on.

```
FullName = '/PC118/DD054/CR012/'
```

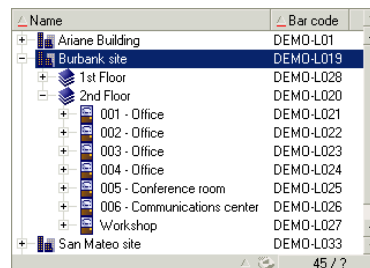
- ◆ In the table of locations, the "FullName" field stores the name of the location preceded by the names of parent locations.

```
FullName = '/Milwaukee/Water St. Site/Building A/5th floor/'
```

"sLvl" fields

For each record in a hierarchic table, the "sLvl" indicates its level in the tree-structure.

Root level is level 0.



The following query selects the "Sales Head Office" record and its sub-components:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Sales Head Office/Sales /%') AND (sLvl >= 1)
```

The following query selects the "Sales Head Office" record but not its sub-components:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Sales Head Office/Sales /%') AND (sLvl = 1)
```

The following query selects the sub-components of the "Sales Head Office" record but not the "Sales Head Office" record itself:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Sales Head Office/Sales /%') AND (sLvl > 1)
```

Simplified AQL notation

This section lists the notation that can be used to simplify the writing of AQL queries:

Foreign keys

In clauses other than the SELECT and ORDER BY clauses, the SQL name of a link that is not followed by a period is equivalent to the SQL name of the associated foreign key.

Example: the clause:

```
WHERE location = 0
```

Is equivalent to:

```
WHERE lLocaId = 0
```

Where "location" is the SQL name of the "Location" link from the table of departments and employees to the table of locations, and "lLocaId" is the SQL name of the associated foreign key in the table of assets.

Description strings

In SELECT and ORDER BY clauses, the SQL name of a link that is not followed by a period is equivalent to the join <SQL name of link>.self, which is in turn equivalent to <SQL name of link>.<Description string>.

Example:

If the description string of the table of departments and employees is:

```
[Name], [FirstName] ([Phone])
```

The AQL query:

```
SELECT user FROM amPortfolio
```

Is equivalent to:

```
SELECT user.self FROM amPortfolio
```

That is itself equivalent to:

```
SELECT (((((User.Name + ',') + User.FirstName) + '(') + User.Phone) + ')')
FROM amPortfolio
```

Features

AQL gives you direct access to the features of a table, as if they were direct fields in the table. To search feature values for a given table, you just need to write the SQL name of the feature (**fv_** prefix).

Example: The following query searches the values of the feature with SQL name **fv_WorkUnit** for the **Departments and Employees** table (**amEmplDept**):

```
SELECT fv_WorkUnit FROM amEmplDept
```

Calculated fields

AQL facilitates the use of calculated fields associated with a table.

You just need to write the SQL name of the calculated field (**cf_** prefix).

Sorts and index

AQL allows two strategies for queries that use sorts (ORDER BY clause):

- A mode where AssetCenter forces the use of the indexes indicated in the query, when such indexes exist, and displays the results throughout the search.
- A mode where AssetCenter does not force the use of the indexes indicated in the queries. In this case, the database engine determines how the data is sorted.

 **Note:**

The choice between these two different methods is not available under SQL Anywhere. The database engine automatically selects the most suitable method.

Example

In the case of the following query:

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

- Access without **Forcing the indexes**: the database engine scans the table in full without using the "Brand" index indicated in the query. It searches all data items satisfying the query, sorts them according to the "Brand" and

sends them to the user. The results will only be displayed after a certain period of time.

- In the other case: The database engine uses the "Brand" index, and displays the results as and when they are found. The first data items are thus shown rapidly, but the overall processing time is longer.

How to force the indexes

The way in which you force the utilization of the indexes depends on the way in which you create the query.

Via the Configure list menu

You can configure the type of data access for each AssetCenter list, whether it be a main list or a tab list. To do this:

- 1 Go to the list you want to configure.
- 2 Right-click.
- 3 Select **Configure list** in the shortcut menu that appears.
- 4 In the **Columns and sort** tab, check the **Force indexes** box in order to use the indexes indicated in the query and display the results as and when they are generated. Uncheck this box in order to select the other type of access.

Using AQL

If you write a query directly in AQL, you can force the use of indexes via the "FIRST_ROWS" clause.

Example:

```
SELECT FIRST_ROWS AssetTag FROM amAsset ORDER BY AssetTag
```

 **Note:**

If the sort focuses on the system itemized lists (for example, on the Features table in the **seDataType** field), there is a possibility that the sort is not valid when an index is forced.

Sort order

The sort order depends on:

- The database engine.
- The use of indexes.

Under Oracle for WorkGroups

With indexes forced

- NULL records do not appear.
- The sort is performed according to the value of the ASCII codes thus differentiating between upper and lower case characters (binary sort).

Without indexes forced

- NULL records appear.
- Oracle for WorkGroups is not case sensitive.

Example

Sort

Starting list	A B C D a b NULL NULL
List with indexes forced	A B C D a b
List without indexes forced	NULL NULL A a B b C D

Microsoft SQL Server or Sybase SQL Server

The sort order depends on a parameter set when the database is created. These engines can be configured in order to be case sensitive or to take accented characters into account, etc.

Under Sybase SQL Anywhere

Under Sybase SQL Anywhere, the indexes cannot be forced via an AQL query. The database engine determines the optimal method used to access the data and to sort it.

Precautions

With complex queries it often difficult to determine whether it is more "advantageous" to force the indexes or not. In practice, we recommend performing tests before making a final decision.

In particular, we recommend testing with and without the indexes forced in the case of a list that is filtered, be it explicitly (via a simple filter, query) or implicitly (via access restrictions).

The query editor

AssetCenter has a query editor. This is a to finalize, perfect and view queries. It is intended particularly for database administrators and advanced users.

How it works

The query editor makes it possible to design queries:

- Either indirectly by using the graphic interface.
- Or by writing the query directly in AQL.

Whether you use the graphic method or prefer to write directly in AQL (both approaches are frequently combined), you get to see a real-time transcription of your query in SQL. However, you cannot write your queries directly in SQL.

Figure A.1. Query editor - composition modes



Using the query editor, a power user or an administrator can create, modify or delete AQL queries. These queries can be used in the appropriate context by their author or other users.

Accessing the query editor

You can access the query editor in different ways:

- Via the **Tools/ Queries** menu item. Using this menu, you can create queries for your own use, which can also be used freely by other users. The queries can be executed:
 - Either directly via the window displayed by the **Tools/ Queries** menu item.
 - Or by when using a "query filter" when displaying the main table of the query.
- From numerous AssetCenter functions that call on queries: access restrictions, query filters, list configurations, purchase-request validations, tax formulas, etc.
- From external programs: Export component of the AssetCenter application, etc.

The version of the query editor that is shown is simplified to a certain degree according to the context.

Example: Let's suppose that you have a query such as the following:

```
SELECT [FIRST_ROWS] <field>[, <field>...] FROM <table> [WHERE <clause>] [ORDER BY <clause>]
```

In the simplified versions of the query editor (simple filters, query filters, etc.), you only have to define the WHERE clause of the query. The other components of the query (starting table, fields, etc.) are implicit. For example, in the case of a query filter, the table is that on which the filter is applied, the fields and the sort conditions are the columns and the sort conditions which are defined via the **Configure list** shortcut menu item. The same is true for the query editor accessed via the **Tools/ Queries** menu item.

Thus, the following query given in full:

```
SELECT self FROM amModel WHERE Brand.Name='Compaq'
```

is written as follows when used in a query filter (only the WHERE clause is explicitly given) used on the table of models:

```
Brand.Name='Compaq'
```

On the other hand, the **Configure list** command enables you to access a more comprehensive version of the query editor:

- The **Columns and sort** tab defines the fields to be displayed in columns and the sort conditions (these sort conditions correspond to the ORDER BY clause).
- The **Force indexes** box replaces the FIRST_ROWS clause in the SQL code.
- The **Filter (WHERE clause)** tab defines the "WHERE" clause.
- The table is implicit.

Creating a query using the query editor

To create a query using the query editor, select the **Tools/ Queries** menu item. The window that is displayed has two tabs, **Filter (WHERE clause)** and **Preview**:

- The **Filter (WHERE clause)** tab is a graphical interface that determines the conditions of your query. It defines the elements of the WHERE clause.
- The **Preview** tab displays the transcription of your query in SQL code and enables you to test it.

Step 1: Populate the fields at the top of the query detail.

You must specify the starting table of your query.

If you want the query you are creating to be able to be accessible to other users, uncheck the **Not shared** option (SQL name: bPrivate).

 **Note:**

The administrator has access to all queries stored in the database, even those queries that are **Not shared**.


Once you have filled in the basic information on the query, click **Create** to be able to access the detail tabs of the query.

Step 2: Define the filter conditions in the Filter (WHERE clause) tab.

The AssetCenter query editor enables you to use criteria that involves fields, uses calculation expressions and integrates constants and operators.

You can define one or more filter conditions.


To define a filter condition:

- 1 From a start table, select a field, constant or expression (**Field 1**), that you will compare with a field, constant or expression (**Field 2**).
- 2 Confirm the filter condition by transferring it to the lower part of the screen using the  button.
- 3 Confirm the query by clicking **Modify** in the query detail.

To define several filter conditions linked by the AND and OR logical operators:

- 1 Create a first filter condition as indicated above.
- 2 Define the other filter conditions and confirm them using the **AND** or **OR** buttons.
- 3 Confirm the query by clicking **Modify** in the query detail.

 Note:


In order to make a modification to the selected conditions, click the  button to delete the contents of the window, or else modify the AQL code directly.

 Note:

In place of the graphic tool, you can enter your query directly in AQL in the zone at the bottom of the **Filter (WHERE clause)** tab.

Step 3: Preview the execution of the query

To test the query and see its transcription in SQL:

- 1 Go to the **Preview** tab in the query detail.
 - 2 Click  : AssetCenter displays a view of the query result in the form of a list of records. The number of records that respond to the query is listed in the lower right-hand corner of the window.
-

 Note:


The SQL code contained in the **Preview** tab cannot be modified directly.

Fields used in a query

When defining filter conditions in queries, you can call on:


- A field in the table concerned by the query.
- A linked field.
- Features associated with the table.

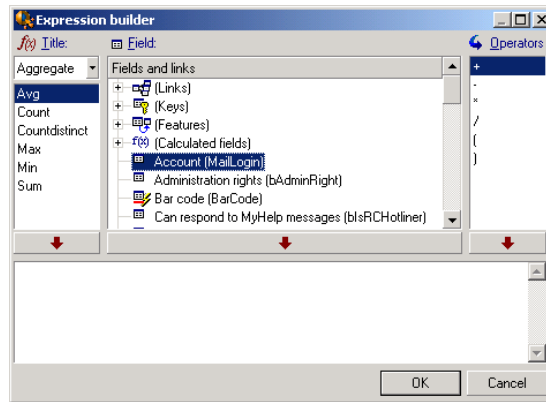
Writing an expression

Expressions  enable you to perform calculation operations in your query. For example, you can use the "Count" function to count the number of resultant records of a query.


To write an expression, you can either:

- Enter it directly in the corresponding field.
- Or use the constructor of the AssetCenter expression.


To use the expression builder, click the  button adjacent to the edit zone of the **Filter (WHERE clause)** tab of the query detail.



The expression builder comprises three columns:

- The "Function" column lists existing AQL functions. Clicking  applies a filter on the list of AQL functions according to their type: "Aggregate", "String", "Date", "Numeric", "Text".
- The "Field" column lists the different fields that can be used in a query.
- The "Operators" column lists the operators that can be used in the expression.

To insert a "Function", "Field" or "Operator" in the expression:


- 1 Select the function, field or operator.
- 2 Click .

Once the expression is defined, click **OK** to transfer it over to the **Filter (WHERE clause)** tab in the query detail.


Constants

Constants x are fixed values that you assign to selection criteria. For example, if you are searching all models with brand **3Com**, you assign the constant value **3Com** to the **Brand.Name** linked field in the table of models.

To select constant:

- 1 Click the  icon.
- 2 A selection window is displayed that shows the values present in the database for the field specified as search condition.

**Note:**

Even in the case of "Itemized list" type fields, the window displayed by clicking the  icon only shows those values used in the database.

AQL syntax

Using AQL requires familiarity with SQL. However, a detailed description of the syntax of SQL is beyond the scope of this manual. For further information, please consult the appropriate reference documentation.

Conventions

Here are the conventions used to describe the syntax of AQL:

Table A.1. AQL - syntax conventions

[]	These brackets surround optional items. Do not type in these brackets.
< >	These brackets surround logical items. Do not type in these brackets.
	The vertical bar indicates that choices are mutually exclusive.
...	The ellipsis indicates that the preceding text may be repeated once or several times.
FROM	Terms in uppercase letters indicate literal expressions.

Syntax of queries

Simple queries

SELECT [DISTINCT] [FIRST_ROWS] <selection list>
 [FROM clause [page 454]]

[WHERE clause [page 455]]

[GROUP BY clause [page 455]]

[HAVING clause [page 456]]

[ORDER BY clause [page 457]]

Sub queries

AQL supports the use of sub-queries in the place of fields.



Note:

In sub-queries, the SELECT statement only authorizes a single expression.

(SELECT [DISTINCT] <expression>

[FROM clause [page 454]]

[WHERE clause [page 455]]

[GROUP BY clause [page 455]]

[HAVING clause [page 456]]

)



Warning:

Sub-queries must be contained with parentheses.

Example of utilization:

```
SELECT Self FROM amAsset WHERE mPrice >= (SELECT Max(mPrice)/2 FROM amAsset)
```

UNION type queries

UNION enables you to group together the results of several queries:

SELECT <selection list>

[FROM clause [page 454]]

[WHERE clause [page 455]]

[GROUP BY clause [page 455]]

[HAVING clause [page 456]]

[**UNION** | **UNION ALL** | **INTERSECTS** | **MINUS**]

SELECT <selection list>

[FROM clause [page 454]]

[WHERE clause [page 455]]

[WHERE clause [page 455]]

[GROUP BY clause [page 455]]

[HAVING clause [page 456]]...

[ORDER BY clause [page 457]]

Elements of a query

Fields and links

The queries call into play fields and links of the AssetCenter database.

You can indicate the name of a field:

- ◆ In reference to the starting table of the query. In this case, it is not necessary to specify the name of this table:

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

Examples from the **Portfolio items** table (**AmPortfolio**):

```
Model
User.Name
User.Location.Name
```

- ◆ As an absolute reference. In this case, you need to indicate the name of the table to which the field belongs:
 - ◆ Either by declaring the table in the **FROM** clause and using its name (or a possible alias):
 - <table.[link...]field>
 - <alias.[link...]field>
 - ◆ Or by not declaring the table in the **FROM** clause and instead using ":":
 - <table.[link...]field>
 - <table[_alias]:[link[_alias]...]field>

These last two notations are particularly useful if you cannot use the **FROM** clause.

For example, when you write a query in AssetCenter, you only have access to the **WHERE** clause. The query's starting table is implicit (table to which the filter applies, **Table field (TableName)** of the query's detail, etc.). However, you might need to use other tables in the query. In this case, the ":" character enables you to do this.

Constants

The following syntax is valid for the constants that may be involved in queries.

Numeric constants

The period is used as decimal separator.

Examples:

12

52.23

Text type constants

They are contained within single quotes.

Examples:

'Computer'

'Monitor'

Date or time type constants

Date or time type constants are contained between # characters. Their format must respect the following rules:

- Years are expressed as 4 figures.
- Dates are expressed as Year-Month-Day.
- Time is expressed as Hours-Minutes-Seconds.
- The 24-hour clock is used (and not the 12-hour clock with A.M. or P.M.).
- The separator used in dates is the "/" or "-" character.
- The separator used for time is the ":" character.
- Months, days, hours, minutes, and seconds are expressed as 2 figures.
- When date and time are expressed together, the data always precedes the time, and they are separated by a space.

Examples:

#yyyy-mm-dd hh:mm:ss#

#yyyy-mm-dd#

#hh:mm:ss#

#2004-01-01 01:00:03#

Expressions

Expressions are formed using:

- Constants
- Fields
- Functions
- Sub-queries

You can combine these elements with operators and parentheses in order to build complex expressions.

Comparison expressions take the form:

<expression> <comparison operator> <expression>

Logical expressions take the form:

<comparison operator> <AND | OR> <comparison operator>

You can use parentheses to combine several logical expressions.

Operators

Logical operators

Logical operators are applied to link two expressions:

Table A.2. AQL - logical operators

Operator	Meaning
AND	Logical "AND"
OR	Logical "OR"

In order to optimize a query, it is sometime wise to avoid logical operators if a comparison operator can be used instead. The following example illustrates how to optimize a query filter used to select portfolio items whose **Assignment** field (SQL name: seAssignment) is set to **Awaiting delivery** or **Return for maintenance**. The values of these two elements of a system itemized list are "3" and "4" respectively. It is therefore possible to write:

```
(seAssignment=3) OR (seAssignment =4)
```

The last value of the system itemized list being "4", it is preferable to write the query as follows:

```
seAssignment >=3
```

Comparison operators

Comparison operators are used to compare two expressions.

Table A.3. AQL - comparison operators

Operator	Meaning
=	Equals
<>	Different than
=!	
>	Greater than
<	Less than
>=	Greater or equal to
=<	Less than or equal to
=*	Right outer join. Because of the way in which AQL handles links, the use of this operator is limited.
*=	Left outer join. Because of the way in which AQL handles links, the use of this operator is limited.
LIKE	Works like the = operator and allows you also to use wildcard characters.
NOT LIKE	<p>The following wildcard characters are available:</p> <p>"%" replaces any character string.</p> <p>"_" replaces any single character.</p> <p>Depending on the database engine used (SQL Anywhere, SQL Server and Sybase support it, Oracle for WorkGroups doesn't):</p> <p>[abc?] lets you define a list of possible characters (no space between the possible values.)</p> <p>[a-c] lets you define a range of possible values.</p> <p>DB2 does not support use of the LIKE X operator, if X includes a SQL column name. Only constants are supported for this operator. For example, the following query is not correct for DB2:</p> <p>SELECT COL1, COL2 FROM TABLE1 WHERE COL1 LIKE COL2</p>
IS NULL	Tests whether the value of a field is "NULL" or not.
IS NOT NULL	AssetCenter only authorizes the "NULL" value for empty text fields and for Date or Date+Time fields that are not populated.

 Note:

SQL Anywhere is not able to process "LIKE X" clauses when X contains more than 128 characters. If X is larger than 128 characters, applying the query provokes an ODBC error message. This problem can, for example, occur when displaying lists in tree view since this operation uses a "LIKE" clause on a "FullName" field.

Operators specific to sub-queries

You can compare a value to the results of a sub-query using the following operators:

- = **ANY (sub-query)**
- = **ALL (sub-query)**
- = **SOME (sub-query)**

Example:

- ◆ The following query provides the list of portfolio items whose brand is used at the Milwaukee site:

```
SELECT lModelId, Model.Brand FROM amPortfolio WHERE Model.Brand = ANY (SELECT Model.Brand FROM amPortfolio WHERE Location.FullName = '/Site Burbank')
```

Selection list

Selection lists define the items to be extracted or displayed. They specify the SELECT statements in queries.

A selection list is made up of one or more expressions separated by commas:

<expression> [,<expression>...]

Each expression can be linked to an alias. For example:

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

This is particularly useful at the level of export queries to attribute a name to the exported columns.

 Note:

Certain DBMSs limit the number of expressions in a given SELECT statement.

FROM clause

The **FROM** clause indicates the table or tables concerned by a **SELECT** statement. AQL authorizes the utilization of aliases for table names.

Syntax

FROM <name of the table> [alias of the table] [, <name of the table> [alias of the table>] ...]

Starting table of a query

The first table indicated in the **FROM** clause of a query is the starting table of the query.

If a query uses a field whose table is not specified, AQL considers that it belongs to the starting table of the query. The AQL **FROM** clause differs from the clause with the same name in SQL.

For example, in the following sub-query, AQL searches the **AssetTag** field in the **amAsset** table:

```
SELECT AssetTag FROM amAsset
```

Number of tables in a query

The number of tables authorized in a query depends on the DBMS used.

Example:

- Oracle: You can use as many tables as you like.
- Microsoft SQL Server or Sybase SQL Server: You are limited to 16 tables in a query.

Warning:

When counting the number of tables in a query, do not forget to take into account those tables that are not explicitly mentioned, in particular if the query uses links. Also look out for the "fv_" notation (search of feature values) which generates an additional join at DBMS level. Similarly, the "cf_" notion (calculated fields) can generate additional joins.

Examples

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

The following queries are equivalent:

```
SELECT AssetTag FROM amAsset
SELECT a.AssetTag FROM amAsset a
SELECT amAsset.AssetTag FROM AmAsset
```

WHERE clause

The AQL **WHERE** clause is equivalent to the clause in SQL of the same name. It specifies the search conditions, specifying the elements to extract from the database. These conditions can also express themselves in **WHERE** or **HAVING** clauses.

Syntax

WHERE <Search conditions>

Writing search conditions

In the majority of cases, you will need to write the conditions using the following form:

```
<WHERE | HAVING> [NOT] <expression> <comparison operator> <expression>
<WHERE | HAVING> [NOT] <logical expression>
<WHERE | HAVING> [NOT] <field> [NOT] LIKE 'xxxxx'
<WHERE | HAVING> [NOT] <logical expression> <AND | OR> <logical expression>
>
<WHERE | HAVING> [NOT] <field> IS [NOT] NULL
```

In some other case, you may need to write more complex queries, such as:

```
<WHERE | HAVING> [NOT] EXISTS (<sub-query>)
<WHERE | HAVING> [NOT] <expression> [NOT] IN (<list of values> | <sub-query>)
<WHERE | HAVING> [NOT] <expression> <comparison operator> <ANY | ALL> (<sub-query>)
```

GROUP BY clause

The AQL **GROUP BY** clause is equivalent to the clause in SQL of the same name.

Syntax

GROUP BY <expression without aggregate> [, <expression without aggregate>]...

Writing tips

GROUP BY specifies subsets of the table. The subsets are defined in the **GROUP BY** clause by an expression, which can be the name of a field, for example.

If aggregate functions are included in the selection list of the **SELECT** statement, **GROUP BY** searches the resulting value for each subset. These resultant values can be used in a **HAVING** clause.

When a query makes use of the **GROUP BY** clause, each expression of the selection list must provide a single value for each subset.

GROUP BY - Examples

The following query gives the total number of brands found in the database. For each asset with an associated brand, AssetCenter returns an occurrence of the brand.

```
SELECT Count (Model.Brand.Name) FROM amAsset
```

By using the **GROUP BY** clause, we obtain a list of brands and the number of assets of each brand:

```
SELECT Model.Brand.Name, count (lAstId) FROM amAsset GROUP BY Model.Brand
```

HAVING clause

The AQL **HAVING** clause is equivalent to the SQL clause of the same name.

Syntax

HAVING <Search conditions>

Differences with the WHERE clause

It specifies the search conditions like the **WHERE** clause. However, these two clauses differ as follows:

- The **HAVING** clause specifies the restrictions to be applied to aggregate functions in the selection list. The restrictions affect the number of resultant lines but do not affect the calculations linked to aggregate functions.

- When the query uses an **WHERE** clause, the search conditions restrict the lines subject to aggregate calculation functions but do not affect the resultant lines.

Examples

Example of query where the **WHERE** clause is equivalent to the **HAVING** clause: The following query returns the list of brands whose name starts with a letter after the letter **B** and the number of asset of each of these brands:

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset GROUP BY Model.Brand.N
ame HAVING Model.Brand.Name > 'B'
```

It is also possible to express the same query using the **WHERE** clause:

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset WHERE Model.Brand.Name
> 'B' GROUP BY Model.Brand.Name
```

Example of query using the **HAVING** clause:

The **HAVING** clause enables you to use aggregate functions (such as **Count**); This is not the case with the **WHERE** clause. Thus, the following query searches all brands represented by more than one asset:

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset GROUP BY Model.Brand.N
ame HAVING count(Model.Brand) > 1
```

ORDER BY clause

The AQL **ORDER BY** clause is equivalent to the SQL clause of the same name. Items can be sorted:

- In ascending order: **ASC**. This is the default sort order.
- In descending order: **DESC**.

Syntax

```
ORDER BY <expression> [ASC | DESC] [, <expression> [ASC | DESC]...]
```

INSERT clause

This clause enables you to insert one or more records into a database table.

Syntax

INSERT INTO <Name of the table> [alias of the table] (<Name of a field> [, <Name of a field>...] **VALUES** (<expression> [, expression]...) | **AQL sub-query**)

This clause is included in the AssetCenter API **AmDbExecAql**.

For more information about AssetCenter APIs, refer to the Programmer's Reference guide, chapter **Alphabetic reference**.

Example

The **INSERT** clause can simplify the code of a **Supplemental delivery information wizard**:

Wizard code not using the INSERT clause

```
hrAlarm = AmCreateRecord("amDateAlarm")
lErr = AmSetFieldLongValue(hrAlarm, "bSecondLevel", 0)
lErr = AmSetFieldLongValue(hrAlarm, "dtTrig1", AmGetFieldLongValue(hrAsset, 2) - lDaysBefore * 86400)
lErr = AmSetFieldLongValue(hrAlarm, "lAction1Id", lActionId)
lErr = AmSetFieldLongValue(hrAlarm, "lMonitObjId", lAstId)
lErr = AmSetFieldStrValue(hrAlarm, "MonitoredField", "dWarrEnd")
lErr = AmSetFieldStrValue(hrAlarm, "MonitoredTable", "amAsset")
lErr = AmSetFieldLongValue(hrAlarm, "sDaysBefore1", lDaysBefore)
lErr = AmInsertRecord(hrAlarm)
```

Wizard code using the INSERT clause

```
lErr = AmDbExecAql("insert into amDateAlarm (bSecondLevel, dtTrig1, lActionId, lMonitObjId, MonitoredField, MonitoredTable, sDaysBefore1) values ( 0, " & AmGetFieldLongValue(AmGetFieldLongValue(hrAsset, 2) - lDaysBefore * 86400 & ", " & lAstId & ", 'dWarrEnd', 'amAsset', " & lDaysBefore & ")")
```

UPDATE clause

This clause enables you to update one or more fields of a record in a database table.

Syntax

UPDATE<Name of the table> [alias of the table] **SET** (<name of a field> [, <name of a field>...] [**FROM clause** [page 454]] [**WHERE clause** [page 455]])

Example

The **UPDATE** clause can help simplify the code of an action that triggers an order action.

Action code not using the **UPDATE** clause

```
hr = AmGetRecordFromMainId("amPOrder", [lPordId])
lErr = AmSetFieldLongValue(hr, "seStatus", "$ (IDS_POSTSTATUS_ORDERED) ")
lErr = AmUpdateRecord(hr)
```

Action code using the **UPDATE** clause

```
lErr = AmDbExecAql("update amPOrder set seStatus = 21 where lPordId = " &
[lPordId])
```

DUPLICATE clause

This clause enables you to duplicate a record existing in a database table.

This function is specific to AssetCenter.

For more information, refer to the **User interface** guide, chapter **Operations on records**, section **Duplicating a record**.

Syntax

DUPLICATE <Name of the table> [alias of the table] **SET** (<name of a field> [, <name of a field>...]) [**FROM** clause [page 454]] [**WHERE** clause [page 455]]

DELETE clause

This clause enables you to delete the fields of a record in a database table.

Syntax

DELETE [**FROM** clause [page 454]] [**WHERE** clause [page 455]]

AQL function references

The following AQL functions can be used in queries and formulas:

- Aggregate type AQL functions
- String type AQL functions
- Date type AQL functions
- Numeric type AQL functions
- Test type AQL functions



Note:

You can also use native SQL functions of your DBMS. In this case, the resulting code is not portable.

Aggregate type AQL functions

Table A.4. AQL - Aggregate-type functions

Function	Description
Avg(<column>)	Returns the average value of a "number" type column. Returns "0" if the column does not have any records.
Count(<column>)	Counts the non-null values in a column.
Countdistinct(<column>)	Counts the distinct non-null values in a column.
Max(<column>)	Returns the maximum value in a "number", "string" or "date" type column. If the column does not have any records, returns "0" ("number" type column), "empty string" ("string" type column), or "empty date" ("date" type column).
Min(<column>)	Returns the minimum value in a "number", "string" or "date" type column. If the column does not have any records, returns "0" ("number" type column), "empty string" ("string" type column), or "empty date" ("date" type column).
Sum(<column>)	Returns the sum of the values of a "number" type column. Returns "0" if the column does not have any records.

These functions jointly use the "GROUP BY" and "HAVING" clauses.

String type AQL functions

Table A.5. AQL - String-type functions

Function	Description
Ascii(<String>)	Returns the ASCII value of the first character of the <string>.
Char(<n>)	Returns the character with ASCII code "n".
Left(<String>, <n>)	Returns the "n" first characters of the <string>.
Lower(<String>)	Returns the <string> in lowercase.
Ltrim(<String>)	Removes the spaces at the left of the <string>.
Right(<String>, <n>)	Returns the "n" last characters of the <string>
Rtrim(<String>)	Removes the spaces at the right of the <string>
Substring(<String>, <n1>, <n2>)	Extracts the sub-string starting at character "n1" in the <string> and with length "n2" (the 1st character of the <string> is considered as character number 1).
Upper(<String>)	Returns the <string> in uppercase.

Date type AQL functions

Table A.6. AQL - Date-type functions

Function	Description
Year(<date>)	Returns the number representing the year for a "date" or "date and time type field" (e.g: 2000).
Month(<date>)	Returns the number of the month for a "date" or "date and time type field" (1=January, ?, 12=December).
Day(<date>)	Returns the number of the day in the month for a "date" or "date and time" type field (1-31).
DayOfYear(<date>)	Returns the number of the day in the year for a "date" or "date and time" type field (1-366).

Function	Description
WeekDay(<date>)	Returns the number of the day in the week for a "date" or "date and time" type field. This number depends on how the server is configured. For example, the default configuration under Sybase or Microsoft SQL Server is (1=Sunday, 2=Monday, ..., 7=Saturday). The default configuration under Oracle is (1=Monday, ..., 7=Sunday).
Hour(<hour>)	Returns the number of the hour in the day for a "time" or "date and time" type field (0-23).
Minute(<hour>)	Returns the number of minutes for a "time" or "date and time" type field (0-59).
second(<hour>)	Returns the number of seconds for a "time" or "date and time" type field (0-59).
Getdate()	Returns the server's current system date.
AddDays(<date>, <number>)	Adds a given number of days to a "date" or "date and time" type field.
AddHours(<date>, <number>)	Adds a given number of hours to a "date" or "date and time" type field.
AddMinutes(<date>, <number>)	Adds a given number of minutes to a "date" or "date and time" type field.
AddSeconds(<date>, <number>)	Adds a given number of seconds to a "date" or "date and time" type field.
DaysDiff(<date1>, <date2>)	Number of days between the dates date1 and date2 ("floating point" number with decimals)
HoursDiff(<date1>, <date2>)	Number of hours between the dates date1 and date2 ("floating point" number with decimals)
MinutesDiff(<date1>, <date2>)	Number of minutes between the dates date1 and date2 ("floating point" number with decimals)
SecondsDiff(<date1>, <date2>)	Number of seconds between the dates date1 and date2 ("floating point" number with decimals)
DbToLocalDate(<date>)	Converts a date expressed in the time zone of the database server to a date expressed in the time zone defined at client machine level.
LocalToDbDate(<date>)	Converts a date expressed in the time zone of the client machine to a date expressed in the time zone of the database server.

Table A.7. AQL - Examples of Date-type functions

Description	AssetCenter query language
All records modified during the last week	AddDays(dtLastModif,7)>=Getdate()

Description	AssetCenter query language
All work orders notified in the last hour	HoursDiff(Getdate(), dtNotif) <= 1 Or AddHours(dtNotif, 1) >= Getdate()
All work orders closed in the last half-hour	MinutesDiff(Getdate(), dtActualFixed) <= 30 Or AddMinutes(dtActualFixed, 30) >= Getdate()

The following query lists the work orders effectively carried out and resolved during the same day. The time zone of the client machine is taken into account:

```
SELECT Self FROM amWorkorder WHERE DayOfYear(DbToLocalDate(dtActualFixStar
t)) = DayOfYear(DbToLocalDate(dtActualFixed))
```

The following query lists all work orders that have effectively been started today:

```
SELECT Self FROM amWorkorder WHERE DayOfYear(DbToLocalDate(dtActualFixStar
t)) = DayOfYear(DbToLocalDate(GetDate()))
```

Numeric type AQL functions

Table A.8. AQL - Numeric-type functions

Function	Description
Abs(<Number>)	Returns the absolute value of a "number".
Ceil(<Number>)	Returns the smallest integer greater or equal to a "number".
Floor(<Number>)	Returns largest integer less than or equal to a "number".
Mod(<a>,)	Returns the remainder of the integer division from "a" by "b" ($a = qb + r$, with q integer and $0 \leq r < q$).
Round(<a>, <n>)	Rounds "a" to "n" decimal places.
Trunc(<a>, <n>)	Truncates "a" to "n" decimals.

Examples of application:

Abs (2.516) = 2.

Ceil (2.516) = 3.

Floor (2.516) = 2.

Mod (6,4) = 2.

Round (31.16, 1) = 31.20.

Round (31.16, 0) = 31.00.

Round (31.16, -1) = 30.00.

Trunc (31.16, 1) = 31.1.

Test type AQL functions

Table A.9. AQL - Test-type functions

Function	Description
IsNull(<a>,)	If "a" is "Null", replaces "a" by "b". The types of "a" and "b" must be compatible.



Index

A

- Action after processing, 93 , 88
- Advanced configuration, 29
- Advanced mode, 16
- Advanced options, 40
- Application connectors, 133
- AQL, 427
 - Conventions, 447
 - Force indexes, 440
 - Joins, 430
 - ORDER BY, 439
 - Queries, 447
 - Query editor, 442
 - Simplified notation, 438
 - Syntax, 447
- Archival, 29
- Array fetching, 50
- Attachments - Lotus Notes, 143

B

- BIOS, 245
- Blob
 - Production directives, 50
 - ServiceCenter connector, 121

C

- Cache
 - Configuration, 39
- Case, 56
- Commit, 38
- Compatibility
 - Command-line connector, 242
 - Database connector, 194
 - LDAP connector, 228
 - NT Security connector, 173
 - XML listening connector, 205
- Configuration wizard, 16
 - Defining a connection type, 18
 - Determine server delay, 33
 - Reconnection parameters, 31
- Connection, 18
 - Configuring, 19
 - Connection type, 18
 - DB2, 26
 - MySQL, 24
 - ODBC, 20
 - Oracle, 21
 - Sybase, 23
- Connection test, 32
- Connector

- Action Request System, 133
- Altiris, 301
- Asset Insight, 304
- Asset Management, 63
- BizDoc, 98
- CA Unicenter AMO 3, 341
- CA Unicenter AMO 4, 344
- Command-line, 242
- Configuration, 16, 15
- Consumption directives, 51
- Database, 193
- Defining a connection type, 18
- Delimited-text, 246
- Desktop Administration 6.x, 76
- Directives, 47
- E-mail, 212
- Enterprise Discovery, 82
- iInventory, 319
- Inventory, 41
- LANDesk for Inventory 8, 309
- LANDesk Software Distribution, 312
- LDAP, 228
- Lotus Notes, 140
- Mercury, 177
- MQSeries, 150
- Naming and describing a connector, 17
- NT Security, 173
- OpenView Inventory Manager, 371
- OpenView Service Desk (Inbound), 366
- OpenView Service Desk (Outbound), 363
- Peregrine Desktop Inventory 7.x, 85
- Peregrine Desktop Inventory 8.x, 90
- Peregrine Network Discovery, 95
- Peregrine Systems, 63
- Production directives, 48
- SAP, 351
- SAP ALE, 360
- SAP BAPI, 353
- SAP IDoc, 357
- SCAuto listening, 103
- ServiceCenter, 112
- ServiceCenter Web Service, 130
- SMS 2.x, 315
- SMS 2003, 315
- Tivoli CM for Inventory 4.2, 327

- Tivoli Enterprise Console, 182
- Tivoli Enterprise Console (sending), 177
- Tivoli Enterprise Data Warehouse, 424
- Tivoli Inventory (version 4.0), 323
- TS.Census 2, 333
- TS.Census 3, 337
- Web Service, 269
- Winpark Actima, 347
- XML, 275
- XML listening, 204
- CurrentUser, 435

D

- DELETE, 459
- Directives, 47
 - Definition of consumption directives, 51
 - Definition of production directives, 48
 - Reconciliation, 52
 - WHERE and ORDERBY clauses, 49
- Document
 - Directives, 47
- Document type
 - Consumption directives, 51
- Document types
 - Definition, 42
- DSC, 262
- DSE, 231
- DTD
 - BizDoc connector, 100
 - E-mail connector, 218
 - MQSeries connector, 155
 - Peregrine Desktop Inventory connector, 94, 89
 - XML connector, 281
- DUPLICATE, 459

E

- ERP connectors
- External application, 15

F

- Foreign keys, 438
- FROM, 454
- FullName, 436

G

Gateway (connector), 315
 getdate(), 71
 GROUP BY, 455

H

HAVING, 456
 Hierarchic tables, 436

I

INSERT, 457
 Inventory connectors, 301-332

J

Java, 43
 Java virtual machine, 43
 JVM, 43

L

Learning mode

- MQSeries connector, 158
- XML connector, 284

 Links

- Following (Asset Management connector), 71

 Logical operators, 451

M

MAPI, 224
 Mapping

- Collection-to-collection mapping, 72
- Consumption directives, 51

N

NULL, 433

O

Optimized SQL, 428
 ORDER BY, 457, 49

- Description strings, 438

 Outer joins, 71
 Out-of-the-box scenarios, 375-425

P

Peregrine Systems connectors, 63-82
 Post-processing actions

- Delimited-text connector, 249
 - UrlFileInfo.Path, 280, 253
 - XML connector, 280

 Primary key, 432
 Protocol connectors, 193
 Proxy, 279
 Proxy server, 279

Q

Queries

- AssetCenter, 429

 Query

- Constants, 450
- Fields and links, 449
- Query editor, 442

 Query editor, 442

R

Reconciliation, 57, 52

- Behavior, 57
- Case, 56
- Collections, 58
- Key set, 56
- Reconciliation scripts, 60
- Reconciliation type, 52

 Reconciliation keys, 53

- Key set, 56

 Reconciliation scripts, 60
 Reconnection, 31

- Automatic, 31

 Replication

- Conflicts, 57

S

scdb.cfg, 118
 Scenarios

- Action Request System, 376
- Asset Insight, 378
- Asset Management, 379
- e-mail, 379
- Enterprise Discovery, 381

- LANDesk, 386
- LDAP, 389
- Mercury, 388
- MQSeries, 390
- NT Security, 392
- OpenView Inventory Manager, 385
- Peregrine Desktop Inventory, 396
- Peregrine Network Discovery, 394
- RDS, 399
- SAP
 - BAPI, 402
 - IDOC, 399
- ServiceCenter, 406
- ServiceCenter Web Service, 416
- SMS, 405
- Tivoli Inventory, 420
- TS Census, 418
- Unicenter AMO, 424
- Web Services, 425
- Winpark Actima, 425
- Schedule pointers, 36
 - Use, 36
- SELECT, 447
 - AssetManagement connector, 68
 - Description strings, 438
- Self, 434
- send2cit.sh, 184
- Server
 - Determine server delay, 33
- Session, 32
- sLvl, 437
- SQL92, 29
- Substring, 428
- Sybase
 - Index, 441
- sysdate, 71
- System itemized lists, 436

T

- TEDW, 424
- Tivoli CM for Software Distribution 4.2 , 326
- Tivoli CM for Software Distribution Status 4.2, 332
- Transaction, 38

U

- UNION, 448
- UNIX
 - Asset Management connector, 66
 - Command-line connector, 242
 - E-mail connector - Processing options, 215
 - ServiceCenter connector - using, 114
 - XML connector - Transformation rule for end-of-line markers, 299
- UPDATE, 458
- UrlFileInfo
 - Structure - BizDoc connector, 102
 - Structure - Delimited text connector, 268
 - Structure - XML connector, 299
- UrlFileInfo.Path
 - Delimited-text connector, 253
 - XML connector, 280

W

- WHERE, 455 , 49
 - Lotus Notes Connector, 145

X

- XSD
 - BizDoc connector, 100
 - MQSeries connector, 155
 - Peregrine Desktop Inventory connector, 94 , 89
 - XML connector, 281

