HP OpenView Connect-It

Software version: 3.70

User's Guide

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

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.

Table of Contents

List of Figures

List of Tables

Introduction

Connect-It is an Hewlett-Packard application, the components of which are detailed in the following diagram.

Figure 1. Connect-It - Application components

What does Connect-It do?

Connect-It is an EAI (Enterprise Application Integration) type integration platform. An EAI solution enables a company to integrate the different

applications from which it can obtain or to which it can provide internal data (Internal support, equipment management software, etc.) or external data (ERP, B2B, B2C).

Connect-It integrates not only data, but also a company's application processes.

You can use Connect-It to:

- Transfer information from one database to another.
- Duplicate the information from one database to another in real-time.
- Import information from e-mails, delimited text files, XML files or other formats into a database.
- Export information from a database to e-mails, delimited text files, XML files or other formats.
- Import NT Security-based information into a database.
- Etc.

Who is Connect-It intended for?

Connect-It is intended for IT technicians in charge of integrating different applications used by their company.

Using Connect-It requires the following expertise:

- Advanced knowledge of the applications being integrated.
- Understanding of Basic language used in the mapping scripts.

How to use this guide

Installation

This chapter explains the Connect-It installation procedure.

General concepts

This chapter presents an overview of how Connect-It works.

Ergonomics

This chapter presents the layout and menus of the Scenario builder, a tool that helps you create integration scenarios.

Implementing an integration scenario

This chapter explains how to set up an integration scenario. This implementation involves:

- Choosing the source and destination connectors.

  Connectors communicate with external applications that Connect-It helps you integrate. These connectors include the Database connector, the E-mail connector and numerous other connectors.

- Creating the document types that are either produced or consumed by the connectors.

  A document type is the equivalent - in an XML format - of the data sets managed by the external applications: database tables, e-mails, text files, property formats, etc.

- define mappings and mapping scripts

- set up the monitors to use

- automate the operation of a scenario created by the declaration of services and the implementation of scheduling

Managing an integration scenario

This chapter explains how to run an integration scenario:

- deploy a scenario under Unix
- use the application in command line format
- test and debugging
- performance optimization

How to signal a problem appendix

This appendix explains how to report an incident to the Hewlett-Packard support team. The If you signal a problem about Connect-It [page 269] section explains how to report incidents for Connect-It.

Glossary

This glossary regroups a non-exhaustive list of terms specific to Connect-It.

1 Installation

This chapter explains how to install the Connect-It integration platform.

Supported environments

Connect-It is Windows compatible.

▶ Hewlett-Packard support compatibility matrix.

Connect-It can also be installed in Unix (Linux, Solaris, AIX), but it does not function graphically.

Connect-It must be installed on a computer having access to the applications that you want to use with the scenarios. This computer does not require a full client-installation of the applications used in the scenarios. On the other hand, a full installation can be quite useful, especially in test phase, to verify that the data has been successfully written to the external applications.

You can set the amount of memory allocated to the application log by using the **Options** command in the **Edit** menu of the Scenario builder. By default, the amount of memory allocated to document tracking is set to 5 MB.

For a complete presentation of the Document log and document-tracking management, refer to the chapter Ergonomics [page 39], section Document log [page 214], and additionally the chapter Define the supervision monitors [page 151].

Installing Connect-It

To install Connect-It in Windows

1 Make sure you have administration privileges

The installation program runs automatically upon inserting the installation CD-ROM.

If this does not happen, explore the files on your cd-rom drive, launch the `autorun.exe` program and select the appropriate installation option.

The installation program lets you choose from several installation types:

- Standard installation
- Custom installation

Integration Pack

The installation of Connect-It includes the installation of an Integration Pack.

An integration pack is made up of:

- Connectors
- Associated scenarios
- Favorites
- Datakit
- Documentation

> **Warning:**
> DO NOT interrupt the installation of the Integration Pack.

Standard installation

The standard installation installs:

- All Connect-It applications: Connect-It, Connect-It Web and Connect-It Services Console
- The documentation in PDF and CHM formats

The following is not installed:

- the MSDE database for the monitors

Custom installation

The custom installation, by default, installs:

- Connect-It and the full set of connectors

You can choose to install the following components during the custom installation:

- the MSDE database for the monitors
- the documentation in PDF format
- Connect-It Web

This installation lets you add or remove connectors.

Installing / uninstalling from the command line

⚠️ Warning:

Installing or uninstalling from the command line is not supported by Hewlett-Packard.

Modifying the installation

If Connect-It is already installed, components can be added or removed via the Add or Remove Programs utility in the Control Panel (select the Connect-It line and then click **Change**).

Integration Pack

Connectors can be added or removed using the Add or Remove Programs utility in the Control Panel (select the Connect-It Integration Pack line and then click **Change**).

An Integration Pack can be modified independently from Connect-It and updated separately.

New connectors, new scenarios as well as connector and scenario updates can be distributed via Integration Packs. Installing an Integration Pack does not modify the Connect-It binary.

By default, when updating an Integration Pack, the connectors selected during the previous installation are maintained. Previously installed connectors are automatically selected. If the update is done in custom mode, new connectors can be selected and added.

A new connector can be added by selecting it from the list of additional connectors.

⚠️ Warning:

If a previously installed connector is no longer selected or if the new Integration Pack does not reference it in the new installation, the connector is removed and will no longer be available in Connect-It. All of that connector's dependencies (scenarios, maptables, etc.) and all user changes made to the scenarios or other files will be lost.

> ⚠️ Warning:
>
> Performing an update using an Integration Pack requires that all elements contained in the Integration Pack be updated. No partial updates are possible.

Update

All files from the previous version (scenarios, maptables, monitors, etc.) are deleted when updating Connect-It. Also, all user changes made to the scenarios or other files will be lost.

Removing the program

In Windows, you must use the Add or Remove Programs utility in the Control Panel to correctly uninstall Connect-It (select the Connect-It Integration Pack line and then click **Change**). Don't forget to make backup copies of your scenarios and customizations.

Integration Pack

An Integration Pack cannot be uninstalled by itself. Uninstalling Connect-It will uninstall the Integration Pack.

MSDE monitors database

To use the Connect-It database that is required by the monitors, you must install the MSDE database components.

When installing the monitors database, a database server configuration window will appear. Specify the following in this window:

- The database server
- The connection ID and password

If AssetCenter has already been installed, you can use the installed instance of MSDE:

Parameter	Value
Instance name	ASSETCENTER
User with administration privileges	sa
Password associated with the **sa** user account	saacpassword
Security system	SQL
Network protocol	Enabled

▶ Monitors database administration [page 261].

To install Connect-It in UNIX

To install Connect-It in UNIX, you must decompress the `.tgz` file provided by executing the following command:

```
tar -xfz cnit-[operating system]_[version].[build]_[language prefix].tgz
```

✎ Note:

You can add the command line option **v** to the **tar** (tar -xvzf) command line.
This option lets you enable the verbose mode.

The following prefixes are available: *en* for the English language version, *fr* for the French language version.

The installation of Connect-It in UNIX is always a full installation. This installation contains the following packages:

- Base connectors [page 31]
- Optional connectors [page 32]
- Datakit [page 32]
- Documentation
- Example scenarios

✎ Note:

Some connectors only function in a Windows environment. For more information about the compatibility of these connectors, consult the compatibility matrix on the Hewlett-Packard support site at:
www.hp.com/managementsoftware/peregrine_support\

Dynamic libraries (.so) in SOLARIS and Linux

In order to use Connect-It, you need to use the dynamic libraries (.so) located in the `bin` folder of the Connect-It installation folder. The `[Connect-It installation folder]/bin` path must be added to the *LD_LIBRARY_PATH* environment variable.

For a csh (csh, tcsh) compatible interpreter, you must execute the following command line:

```
setenv LD_LIBRARY_PATH=[Connect-It installation folder]/bin
```

For sh (ash, ksh, bash, etc.) compatible interpreters, you must execute the following command lines in the `.profile` file:

```
LD_LIBRARY_PATH=[Connect-It installation folder]/bin
export LD_LIBRARY_PATH
```

Dynamic libraries (.so) in AIX

In order to use Connect-It, you need to use the dynamic libraries (.so) located in the `bin` folder of the Connect-It installation folder. The `[Connect-It installation folder]/bin` path must be added to the *LIBPATH* environment variable.

For an interpreter like the C (csh) command, you must execute the following command line:

```
setenv LIBPATH=[Connect-It installation folder]/bin
```

For interpreters like the K (ksh) or Bourne (sh) commands, you must execute the following command line in the `.profile` file:

```
LIBPATH=[Connect-It installation folder]/bin export LIBPATH
```

Install Connect-It Web

> ⚔ Important:
>
> Installing Connect-It Web requires knowledge on how to configure web servers.

Prerequisites

Before installing Connect-It Web, you must install, configure and start the components listed below and do so following the recommendations provided by each of the components' editors.

- Tomcat

  Consult the compatibility matrix to find out which Tomcat versions are supported.

  Tomcat must be protected by your network infrastructure (firewall, proxy, etc.) but allow connections from an Internet browser.

- The following components are provided with the installation:
  - **ddl** folder: `.ddl` files used to create the Connect-It Web database for the different database engines that are supported.
  - webapp folder:
    - `ovcit.war` file, web application file
    - `ovcit.xml` file, sample Tomcat context file for the Connect-It web application
  - **licenses** folder: Licenses for all third-party components and source code of third-party libraries that are used

- **install** folder:
  - `install.bat`, Windows installation script
  - `install`, Linux and Unix installation script
  - `crypt.bat`, Windows script to encrypt passwords for the Connect-It Web authentication module
  - `crypt`, Linux and Unix script to encrypt passwords for the Connect-It Web authentication module

Important:

Consult the compatibility matrix at
www.hp.com/managementsoftware/peregrine_support to find out which versions of the following components are supported:
- Tomcat server
- Web servers

To install Connect-It Web automatically

1 Use the installation script provided by the Connect-It installation depending on the target environment (Windows, Linux or Unix): `install.bat` or `install`
2 Select the installation type: Quick installation with test database, quick installation or custom installation

Note:

All default values provided by the installation program are placed between parenthesis on the command line that is displayed.
Press Enter to validate the default value.

Quick installation with database

The supplied test database does not require a server and is needed during production.
This database is different from the one that is used for the monitors.
1 Start the `install.bat` installation program.
2 Specify the Connect-It installation folder
3 Specify Tomcat's root folder.

Note:

Consult the compatibility matrix to find out which Tomcat versions are supported.

4 Specify the Connect-It Web installation folder.

This folder will contain the files required by the Web application.

If the specified folder is not empty, an error message will be displayed.

> Important:
>
> Due to a known issue in Sun's JRE 5.5, the path to the Connect-It Web folder must NOT contain any spaces.

5 Specify a user name and a password to access Connect-It Web. This data will be saved in the associated authentication file.

6 A summary of the operations that were performed.

The following elements were installed:

- A context file, ovcit.xml, was created and saved under [Tomcat root directory]/conf/Catalina/localhost/
- A Connect-It Web folder was created and contains data required by the application (*conf* folder):
    - `log4j.properties`, *log4j* configuration file, standard work environment Java applications. In this file, the `log4j.appender.file.file` section defines where the log file will be created.
    - `log4j_jmx.properties` , configuration file for *log4j*, used by the service's sub-processes in order to create a different log file for each service. The name of the log file for each service uses this format `ovcit_<service_name>.log`, assuming the main log file is named `ovcit.log`. The log file associated with each service is located next to the main log file.
    - `jaas.config`, JAAS configuration file (http://java.sun.com/products/jaas/).

        JAAS is an authentication standard used in Java applications. Connect-It Web provides a default module which uses a password file. If you would like to change the password management policy, please see the JAAS policy section.
    - `users`, file in which the user passwords for the Connect-It JAAS module are stored.
- `ovcit.policy`, sample file to facilitate the deployment of Connect-It Web in a secure Java environment.

To install Connect-It Web manually

Connect-It Web only operates with a Tomcat server.

Use a custom database for Connect-It Web

You can use a database other than the one provided by default with Connect-It.

1 Create a database using the DDL creation script that corresponds to your database engine which is located in the `<Connect-It installation folder>/web/ddl` folder.

2 Select a JDBC driver for the database that you are going to use. This driver is generally a Java library (jar) defined with the application's environment variables.

3 Define the JDBC driver as a classpath for Connect-It Web.

This is done via either:

- Tomcat's shared environment variables
- Connect-It Web application folder (<Connect-It Web installation folder> /webapp/WEB-INF/lib)

4 Retrieve the URL used to access the database. The URL depends on the JDBC driver that is used.

5 Install Connect-It Web

6 The installation program will request the name of the JDBC driver, the URL to access the database and the user names and passwords to access the database for Connect-It Web.

JDBC Configuration types

The following configuration changes must be made depending on your database engine. The following is a sample configuration.

Oracle

A JDBC driver is provided with your Oracle client installation. This driver is located in the <Oracle installation folder>/client/jdbc folder.

- For Oracle 9 or later, use the `ojdbc14.jar` library.
- The JDBC driver is: `oracle.jdbc.driver.OracleDriver`
- The URL to access the database uses this format:

```
jdbc:oracle:thin:@<server address>:<port>:<service name>
```

The Connect-It configuration file uses this format:

```
<Environment name="com.hp.ov.cit.console.db.driver" override="false"
type="java.lang.String" value="oracle.jdbc.driver.OracleDriver"/>
<Environment name="com.hp.ov.cit.console.db.url" override="false"
type="java.lang.String" value="jdbc:oracle:thin:@oraclesrv.hp.com:1521:orc
l"/>
<Environment name="com.hp.ov.cit.console.db.username"
override="false" type="java.lang.String" value="CIT37"/>
<Environment name="com.hp.ov.cit.console.db.password"
override="false" type="java.lang.String" value="password"/>
```

MS SQL

The MS SQL driver is available on Microsoft's web site.

- Use the `sqljdbc.jar` library.
- The JDBC driver is: `com.microsoft.sqlserver.jdbc.SQLServerDriver`
- The URL to access the database uses this format:

```
jdbc:sqlserver://serverName;instanceName:portNumber;property=value[;pro
perty=value]
```

The Connect-It configuration file uses this format:

```
<Environment name="com.hp.ov.cit.console.db.driver" override="false"
type="java.lang.String" value="com.microsoft.sqlserver.jdbc.SQLServerDrive
r"/>
<Environment name="com.hp.ov.cit.console.db.url" override="false"
type="java.lang.String" value="jdbc:sqlserver://localhost;databaseName=CIT
Demo37;instanceName=MSDE"/>
<Environment name="com.hp.ov.cit.console.db.username"
override="false" type="java.lang.String" value="CIT37"/>
<Environment name="com.hp.ov.cit.console.db.password"
override="false" type="java.lang.String" value="password"/>
```

MS SQL 2K

The MS SQL driver is available on Microsoft's web site.

- Use the `sqljdbc.jar` library.
- The JDBC driver is: `com.microsoft.sqlserver.jdbc.SQLServerDriver`
- The URL to access the database uses this format:

```
jdbc:microsoft:sqlserver://server1\\instance1;User=test;Password=secret
```

MySQL

The driver is available on www.mysql.com.

- Use the `mysql-connector-java-5.0.4-bin.jar` library.
- The JDBC driver is: `com.mysql.jdbc.Driver`
- The URL to access the database uses this format:

```
jdbc:mysql://[host][,failoverhost...][:port]/[database]
[?propertyName1][=propertyValue1][&propertyName2][=propertyValue2]...
```

The Connect-It configuration file uses this format:

```
<Environment name="com.hp.ov.cit.console.db.driver" override="false"
type="java.lang.String" value="com.mysql.jdbc.Driver"/>
<Environment name="log.properties" override="false"
type="java.lang.String" value="E:/CITWEB/conf/log4j.properties"/>
<Environment name="com.hp.ov.cit.console.jaas.loginConfig"
override="false" type="java.lang.String" value="file:E:/CITWEB/conf/jaas.c
onfig"/>
<Environment name="com.hp.ov.cit.api.library.path" override="false"
type="java.lang.String" value="E:/Program Files/HP OpenView/Connect-It 3.7
```

```
0 en/bin"/>
<Environment name="com.hp.ov.cit.console.db.username"
override="false" type="java.lang.String" value="admin"/>
<Environment name="com.hp.ov.cit.console.db.url" override="false"
type="java.lang.String" value="jdbc:mysql://16.16.63.44/test"/>
<Environment name="com.hp.ov.cit.jmx.scenario.jar.location"
override="false" type="java.lang.String" value="E:/Program Files/HP OpenVi
ew/Connect-It 3.70 en/lib"/>
<Environment name="com.hp.ov.cit.console.db.connectionProperties"
override="false" type="java.lang.String" value=""/>
<Environment name="com.hp.ov.cit.console.db.password"
override="false" type="java.lang.String" value=""/>
```

DB2

Use version 8 of the DB2 tools.

- Use the db2jcc.jar library.

  A b2jcc_license_*.jar license file is required.
- The JDBC driver is: com.ibm.db2.jcc.DB2Driver
- The URL to access the database uses this format:

```
jdbc:db2://<server>[:<port>][/<database>][;<property>=<value>[;...]]
```
The default port is 5000.

The Connect-It configuration file uses this format:

```
<Environment name="com.hp.ov.cit.console.db.driver" override="false"
type="java.lang.String" value="com.ibm.db2.jcc.DB2Driver"/>
<Environment name="com.hp.ov.cit.console.db.username"
override="false" type="java.lang.String" value="db2admin"/>
<Environment name="com.hp.ov.cit.console.db.url" override="false"
type="java.lang.String" value="jdbc:db2://localhost:5021/CITWEB"/>
<Environment name="com.hp.ov.cit.console.db.connectionProperties"
override="false" type="java.lang.String" value=""/>
<Environment name="com.hp.ov.cit.console.db.password"
override="false" type="java.lang.String" value="password"/>
```

JAAS module

JAAS: An introduction

Authentication in Connect-It Web is based on JAAS (Java Authentication and Authorization Specification).

JAAS is an authentication standard used in Java applications. JAAS is based on contexts that are defined in a configuration file with which one or more authentication modules can be associated. For more information , please visit the JAAS web site at http://java.sun.com/products/jaas/.

JAAS module provided

Connect-It Web provides a JAAS authentication module.

This module is configured by default during the quick installation of Connect-It Web.

The JAAS module that is provided uses a password file. This password file contains user name/password combinations.

Passwords in the password file are encrypted using the standard MD5 algorithm.

During a default installation, a users file containing passwords is created in the <Connect-It Web installation folder>/conf folder. This file contains the user name/password combination that is requested during installation.

You can add as many users as you like in this file.

The syntax is as follows:

```
user1:encrypted_password1
user2:encrypted_password2
(...)
```

For example, for a user named 'admin' and a user named 'st':

```
admin:5f4dcc3b5aa765d61d8327deb882cf99
st:627fcdb6cc9a5e16d657ca6cdef0a6bb
```

Use the provided crypt script to encrypt your passwords. The script is located in the Connect-It installation folder under: <Connect-It installation folder>\web\install.

For example:

```
crypt.bat password
>5F4DCC3B5AA765D61D8327DEB882CF99
```

Using other JAAS modules

Other JAAS authentication modules can be used with Connect-It Web.

 Important:

Defining this authentication requires knowledge on how to configure JAAS.

In Connect-It Web, the JAAS configuration is defined using the following two files:

- Tomcat context file
- JAAS configuration file

The first step is to create the JAAS configuration file to be used.

A JAAS file has one or more sections. Each section defines an authentication context. Each authentication context has an identifier.

```
default {
com.hp.ov.cit.jaas.FileLoginModule required file="E:/web/conf/users";
};
```

The first step is to specify where Connect-It should look to locate the JAAS configuration. This step is accomplished via the 'com.hp.ov.cit.console.jaas.loginConfig' environment property in the `context.xml` file.

```
<Environment name="com.hp.ov.cit.console.jaas.loginConfig"
value="/path/to/jaas.config" type="java.lang.String" override="false" />
```

Connect-It must know which authentication context is to be used. This step is accomplished via the 'com.hp.ov.cit.console.jaas.loginContextName' environment property in the `context.xml` file.

```
<Environment name="com.hp.ov.cit.console.jaas.loginContextName" value="con
figurationName"
type="java.lang.String" override="false" />
```

In the last step, you must specify the JAAS profile that corresponds to a Connect-It Web administrator.

The default value, which corresponds to the profile generated by Connect-It Web's JAAS authentication module, is CIT_ADMIN. If you choose to use your own JAAS authentication module, you must specify the profile name corresponding to a Connect-It Web administrator that is generated by your authentication module.

This step is accomplished via the 'com.hp.ov.cit.console.admin.role' environment property in the `context.xml` file.

```
<Environment name="com.hp.ov.cit.console.admin.role" value="administrators
"
type="java.lang.String" override="false" />
```

Context file

In Tomcat, a Web application is configured using a context file.

This file is normally located in Tomcat's installation folder (<Tomcat installation folder>/config/Catalina/localhost).

A sample context file is provided with Connect-It Web (<Connect-It>/folder/web/webapp/ovcit.xml). A comment is given for each configuration parameter.

- com.hp.ov.cit.api.library.path

  Path to the Connect-It dynamic library (<Connect-It installation folder>/bin).

- com.hp.ov.cit.jmx.scenario.jar.location

  Path to the folder containing the Connect-It Java libraries (<Connect-It installation folder>/lib).

- com.hp.ov.cit.jmx.scenario.java.path

  Command used to run Java when launching a scenario via Connect-It Web.

  The command must reference the JRE 5 binary.

Supposing that Java is included in the path, the default value is 'java'. If Java is not included in the path or if the default JRE is not JRE 5, the value must be updated.

- com.hp.ov.cit.jmx.scenario.jvm.options

  Additional JVM options to be used when a scenario is launched via Connect-It Web.

- com.hp.ov.cit.jmx.scenario.management.port

  The scenarios are launched using the JMX protocol (client/server protocol).

  The variable defines the communications port that is used by the JMX server.

- com.hp.ov.cit.console.jaas.loginConfig

  URL used to access the JAAS configuration file.

  The quick installation creates a configuration file in the following folder: <Connect-It Web folder>/conf.

- Name of the JAAS context to use. This context must be defined in the JAAS configuration file.

  Property used to define the JAAS context name for Connect-It Web.

  The default configuration file which is generated during the quick installation uses 'Default' as the context name.

- com.hp.ov.cit.console.admin.role

  JAAS profile name corresponding to a Connect-It Web administrator.

  The default value corresponding to the profile that is generated by the Connect-It Web JAAS authentication module is CIT_ADMIN.

  If you choose to use your own JAAS authentication module, you must specify the profile name corresponding to a Connect-It Web administrator that is generated by your authentication module.

- com.hp.ov.cit.console.db.driver

  Class name of the JDBC driver used to access the Connect-It Web database.

- com.hp.ov.cit.console.db.url

  URL of the Connect-It Web database. The format of the URL depends on the JDBC driver that is used.

- com.hp.ov.cit.console.db.username

  User name to connect to the database.

- com.hp.ov.cit.console.db.password

  Password used to connect to the database.

- com.hp.ov.cit.console.db.connectionProperties

  Additional JDBC driver configuration properties.

  The properties are defined in a list of name=value pairs delimited by one space.

- log.properties

  Path of the log4j configuration file.

  The file that was generated by the quick installation is located in <Connect-It installation folder>/conf.
- com.hp.ov.cit.jmx.log.properties

  Path of the log4j configuration file for the services.

  A default configuration file for the service is generated by the quick installation and is located in <Connect-It installation folder>/conf.

Install a maintenance version

If you want to install a maintenance version of Connect-It you must:

1   make a backup copy of all the customized elements (scenarios, .ini files).

> 🖌 Important:
>
> It is highly recommended that you make a backup copy of all customized elements.

2   Install the new version to update the previous version.

> 🖋 Note:
>
> The maintenance version erases all the files of previous versions present on the workstation.

Entering the authorization certificate

When you finish installing Connect-It, you need to enter the authorization certificate provided by Hewlett-Packard into the scenario builder.

The authorization certificate is a text file that contains:

- The list of optional connectors that you have the right to use.
- The date until the optional connectors are valid.
- An encrypted authorization key.

To enter the authorization certificate

1   Launch the Connect-It Scenario builder.
2   Select **Edit authorization certificate** from the **Administration** menu.

3   Enter the authorization certificate in the dialog box that appears.
4   Click **OK**.

This action creates a license.txt file in the Connect-It installation folder.

📝 Note:

In the Unix version, Connect-It does not have a graphical interface. You thus need to enter the authorization certificate using the following procedure:
1   Copy the contents of the license file into a new text file. Do not modify the contents of this file.
2   Save this file as license.txt.
3   Copy the license.txt file into the Connect-It installation folder.

Contents of the installation

This section details the tools and the installed files.

The presence of these files on your computer depends on the type of installation you selected: minimum installation, full installation, customized installation.

Files in the installation folder

The following table presents the file organization of the Connect-It installation folder.

📝 Note:

Certain folders only exists in given operating systems.

Name of folders	Main contents
bin	■ Executables (conitgui.exe, conitsvc.exe, console.exe, logview.exe)
	■ Dynamic libraries (.dll or .so files)
	■ Connect-It text editor configuration file (CodeEdit.cfg)
	■ Various files necessary in order for the provided scenarios and Connect-It to work.
config	Files associated with connectors in all scenarios:
	■ .mpt (maptables)
	■ .str (string files)
	■ .bas (Basic functions and global variables)
datakit	Datakit used by the Connect-It tools.

Name of folders	Main contents
dbb	■ cit33.db: Demonstration database
	■ gbbase.xml: Database description
doc	Connect-It documentation in the following formats:
	■ .pdf
	■ .chm
	■ .html (specific to the *javadoc* documentation).
lib	JAR files containing:
	■ Java classes developed by Hewlett-Packard
	■ Third-party Java classes.
scenario	■ .scn files containing the Connect-It sample scenarios.
	■ Files associated with connectors for the sample scenarios.
web	■ .dll file required to create the database used by Connect-It Web.
	■ ovcit.war file
	■ ovcit.xml file
wizards	Files necessary in order for Connect-It to work.

Base connectors

The base connectors are those installed with any Connect-It installation mode, and as such, they can be used in any mode.

The base connectors are:

- Asset Management connector
- BizDoc connector
- Command-line connector
- Desktop Administration connector
- Enterprise Discovery Connector
- Hewlett-Packard OpenView Service Desk Connectors
- Hewlett-Packard OpenView Configuration Management connectors
- NT Security connector
- HP OpenView Network Inventory connector
- HP OpenView Desktop Inventory connector
- RDS connector
- SCAuto listening connector
- ServiceCenter connector
- ServiceCenter Web Service connector
- Delimited-text connector
- XML connector

Optional connectors

Optional connectors are installed if you select the full installation of Connect-It or a custom installation where you select the optional connectors of your choice.

The optional connectors are:

- Action Request System connector
- Asset Insight connectors
- Database connector
- E-mail connectors
- XML listening connector
- iInventory connector (Lan Auditor)
- LANDesk connectors
- LDAP connector
- Lotus Notes Connector
- Mercury Connectors
- MQSeries connector
- SAP ALE connector
- SAP BAPI connector
- SAP IDoc connector
- System Management Service 2.x and 2003 connectors
- Tivoli Enterprise Console connectors
- Tivoli Inventory (version 4.0) connectors
- Tivoli Configuration Manager for Inventory connector
- Tivoli Configuration Manager for Distribution connector
- TS.Census 2.x and 3.0 connectors
- Unicenter AMO connector
- Web Services connector
- Winpark Actima connector

Datakit

The following table shows the contents of the installed datakit.

Folder	Description
ac/ac4	Contains import scripts that enable you to create categories and features in an Asset Management application.
ac/stdfeat	When you import the script file (sdtfeat.scr), you create records in the Features table of the Asset Management application. These features appear in the document types published by the Asset Management connector.

Folder	Description
doctrans	This folder contains files required for the automatic generation and documentation of the scenarios.
hpovsd	This folder contains the files required by the AssetCenter to Service Desk and Service Desk to AssetCenter scenarios.
javasamples	This folder contains: ■ The sample servlet used by the XML listening connector ■ Examples showing how to implement the SDK
mail	This folder contains the .xml file required for the AssetCenter scenarios (**scenario/mail** folder).
mqseries/acimport	This folder contains the files required for the MQSeries scenario for AssetCenter.
merq	This folder contains the .unl files required for the Mercury to ServiceCenter scenarios.
mqseries/scunload	This folder contains the .unl file required for the MQSeries scenario for ServiceCenter.
pdi/pdi7	This folder contains the files required by the Desktop Inventory 7 scenario for AssetCenter or ServiceCenter.
pdi/pdi8	This folder contains the files required by the Desktop Inventory 8 scenario for AssetCenter or ServiceCenter.
sap/common/IDOC sap/feat	This folder contains files required by the SAP IDOC connector and features required by AssetCenter.
sc/sc61 sc/sc62	This folder contains the .unl files required for the AssetCenter-ServiceCenter replication scenarios.
tecevent	This folder contains a self-extracting file required to use the connector.
tim	This folder contains the parameters for the Tivoli Inventory Management connector configuration wizard.

Associated files

Certain files (.mpt, .str, .bas, .scp) are associated to a scenario. If the scenario needs to be moved to another folder, these files must be moved at the same time and in the same position. For example: The files common to several example scenarios are located in the shared folder. This folder must be created in the same place as where you want to move your scenario.

📝 Note:

Rather than move a scenario using a file manager, we always recommend that you save a scenario from the Scenario builder to the desired destination, whenever this is possible. This guarantees that all the files previously associated to the scenario will continue to be correctly referenced.

Information about Connect-It

Connect-It enables you to access all the information relating to the software and its work environment.

To access this information:

- Select **Help/ About**.
- Click **More**.

    A window will open displaying information relating to the:
    - Software
    - System
    - Database engine

To access information related to the installed Integration pack (Windows installation):

- Select the *Add or Remove Programs* tool.
- Select the *HP OpenView Connect-It Integration Pack* program
- Select *Click here for support information*

    A window will open and display information about the:
    - Vendor
    - Support
    - Version of the installed Integration Pack

Hardware configuration

The different configurations mentioned below are given for informational purposes only. For compatibility-related information, please consult the compatibility matrix on the HP OpenView web site or Hewlett-Packard support site.

The performance levels of Connect-It depend mainly on the CPU speed rather than the read/ write (I/O) capabilities of the computer used.

The recommended hardware requirements for Connect-It are the following:

- Windows configuration
    - CPU: Pentium 4, 2 Ghz
    - RAM: 512 MB
- Unix configuration
    - CPU: SunFire V240, 1GHz Ultra SPARC IIIi
    - RAM: 1 GB

A scenario can be executed with a less powerful hardware setup; Only peformance levels will be affected.

The amount of memory used depends on the number of connectors and mapping boxes used in the scenario. If multiple scenarios are running concurrently, or a scenario contains more connectors, the memory size must be increased.

Table 1.1. Minimum recommended memory size according to operating system

Operating system	Memory size
Solaris 9	RAM (recommended):
▪ conitsvc	▪ 7 MB
▪ cntrsrv *	▪ 8 MB
AIX 5.1	RAM (recommended):
▪ conitsvc	▪ 7 MB
▪ cntrsrv *	▪ 6 MB
Linux	RAM (recommended):
▪ conitsvc	▪ 4 MB
▪ cntrsrv *	▪ 5 MB
Windows	RAM (recommended):
▪ conitsvc	▪ 6 MB
▪ cntrsrv *	▪ 7 MB

Table 1.2. Minimum memory amount for Connect-It Web

Software	Memory size
Tomcat	256 MB

Memory for conitsvc depends on the size of the documents processed.

*: The memory used by a Java connector may be greater than indicated because of the Java Virtual Machine.

Memory for DLL files required by the connector must be added to the given memory size.

Multiple-processors

Connect-It enables target connectors to parallelize the execution of multiple processes. Performance is improved on both single and multiple-CPU systems. Using a multiple-CPU computer improves document processing time compared to a single-CPU computer.

> **Note:**
>
> Performance improvements linked to parallelization depend on the difference between the speed of the production and the speed of the consumption of documents. For example, if a scenario produces documents more slowly than it consumes them, parallelizing consumption will be of no benefit. Other factors can influence performance, such as database architecture, network issues, etc.

Installed programs

The following programs are installed besides the Connect-It application:

◆ Service Console
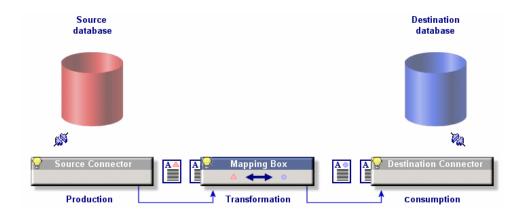
Service Console

This element of the Connect-It software suite is described in chapter Define a Connect-It service (Windows environment) [page 167], section Managing an integration scenario [page 191].

2 General concepts

Processing data

Figure 2.1. Transferring data from one database to another in Connect-It

Connect-It interacts with external applications through the use of connectors (also called components). During a data transfer from one database to another:

■ A source connector produces XML-type documents.

Each document corresponds to a data container that is found in the source application.

- A mapping box reorganizes the structure of the documents produced by the source connector so that they can be consumed by the destination connector.
- A destination connector consumes XML-type documents.

  Each document corresponds to a data container that is found in the destination application.

When using database-type connectors, each document corresponds to a record in a table (and possibly to a link to other tables). For example, a document from an AssetCenter database connector might contain information from the Assets table.

For the other connectors, the data containers correspond to e-mail messages, delimited text files, security information, etc.

The connectors you choose, their relationship to each other, and the definition of mappings between source and destination type documents all create an integration scenario. After the test and debugging phase, a scenario is associated to a schedule and then put into production phase.

As with all other integration processes, these documents can either be partially or totally rejected during the operation. By using processing reports and document logs, you can reprocess these rejected documents without having to redefine your entire integration scenario.
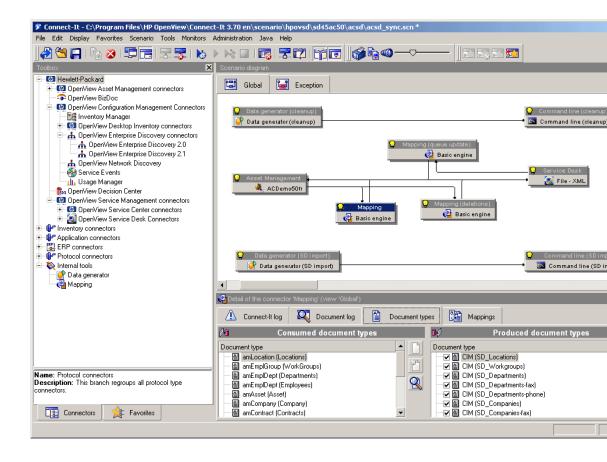
3   Ergonomics

The Scenario builder is the graphical interface that enables you to create integration scenarios. This chapter presents the ergonomics and principle elements of this builder, as well as the Connect-It Options window.

The Scenario builder

The main window of the Scenario builder has:

- A menu bar

  To learn about these menus, refer to this chapter's section Menus [page 46].
- Toolbars
- Panes

Figure 3.1. Scenario builder - Main window

Toolbar

The Scenario builder toolbars allows you to execute the most common operations without having to use the menus. These are the available toolbars:

- Toolbar (generic)
- Diagram bar (of the scenario)
- Views toolbar
- Favorites sidebar

Table 3.1. Scenario builder - Generic toolbar icons

Icons	Functions
	Launches the wizard that helps you create a minimal scenario with a source connector and a destination connector.
	Opens an existing scenario.
	Saves the current scenario.
	Enables you to copy the textual elements of Connect-It: ■ The connector names in the Toolbox. ■ A message line in the logs. ■ Etc.
	Deletes the selected tool of the scenario.
	Launches the configuration wizard of the selected tool.
	Enables the configuration wizards to display advanced configuration pages. Example: the *Parallelize consumption* page.
	Opens all the connectors in a scenario.
	Closes all the connectors in a scenario.
	Launches the scenario in scheduled mode.
	Tests the scenario (non-scheduled mode).
	Launches the scenario in debug mode.
	Stops the current scenario.
	Purges the lines of the **Connect-It log** and the **Document log**.
	Goes into offline/with connection mode.
	Enables you to test the scenario without having the destination connectors interact with the external application. For Database type connectors supporting these transactions, error messages are obtained corresponding to those messages that would normally be obtained if this option had not been selected.
	Displays or hides the **Toolbox** pane.
	Displays or hides the **Scenario diagram** pane.

Table 3.2. Scenario builder - Favorites sidebar icons

Icons	Functions
	Enables you to reconfigure a favorite selected in the **Favorites** tab of the Toolbox.
	Enables you to delete a favorite selected in the **Favorites** tab of the Toolbox.
	Enables you to import the Favorite.fav into the folder of your choice.
	Enables you to export the Favorite.fav file to the folder of your choice. To use the favorites, other users can import this file in the bin folder of their Connect-It installation folder.

Table 3.3. Scenario builder - Scenario builder's toolbar icons

Icons	Functions
🗄	Prints the scenario diagram.
	(A print preview window enables you to configure the parameters of the print job.)
🗄	Copies the scenario diagram to the Clipboard.
🔵	The slider next to this icon enables you to zoom in or out of the scenario diagram.

Table 3.4. Scenario builder - Views toolbar icons

Icon	Function
🔲 Global ▾	Before creating a mapping, this enables you to select the view in which the source and destination connectors of the mapping appear once the mapping is created.

To display or hide a toolbar

1 Select the **Display** menu.
2 Select or unselect one of the following toolbars:
    - Toolbar (generic)
    - Diagram bar (of the scenario)
    - Views toolbar

To move a toolbar

1 Click the toolbar's handle (‖).
2 Move the toolbar to the desired location.

You can move a toolbar in order to:
- Put it in one of the four corners of the Scenario builder window.
- Make it a floating toolbar, able to rest anywhere on your screen.

To dock a floating toolbar

1 Place the pointer on one of the borders of the floating toolbar.
2 Wait for the pointer to change (↘).
3 Double-click.

    The toolbar will automatically dock itself in its original location.

Panes

The main window of the Scenario builder is comprised of three resizable panes:

- A Toolbox that enables you to choose the tools or the favorites for your scenario.
- A pane, called the Scenario diagram, in which the tools and links of the current scenario are represented graphically.
- A Detail pane showing several tabs:
  - The Connect-It log tab
  - The Document log tab
  - The tabs relating to the tool selected in the scenario: produced or consumed document-types, mappings, etc.

To display or hide a pane

You can display or mask the following panes:
- The Toolbox pane

  To display or hide this pane, do one of the following:
  - Select or unselect the menu **Display/Toolbox**.
  - Click ▥.
- The Scenario diagram pane

  To display or hide this pane, do one of the following:
  - Select or unselect the menu **Display/Scenario diagram**.
  - Click ▥.

To move a pane

1 Click the upper border of the pane.
2 Move the pane to the desired location.

Scenario-diagram views

The **Scenario diagram** pane contains two tabs: **Global** and **Exception**. The first tab displays all components of the current scenario: source and destination connectors, links and mapping boxes. The second tab by default does not display any information but it is intended for scenario error handling.

You can create different views in the **Scenario diagram** pane. These views let you display only those tools used by one or more given mappings.

Example: In a scenario using several source connectors, you select the mappings and one, single source connector. This way you create a view that just displays that one source connector.

Using the scenario-diagram views

The views created for a scenario appear each time that you open this scenario in the Scenario builder.

To create a scenario-diagram view

1 Open your scenario.
2 Select **Scenario/ View manager**.
3 Click ▯ in the window that appears.
4 Name your view. This name must be unique for a scenario.
5 Select the mappings for which you want to have a scenario-diagram view.
6 Click **Create**.

A tab named after your view appears in the **Scenario diagram** pane after closing the view editor.

To modify a view of a scenario diagram

1 Open your scenario.
2 Select **Scenario/ View manager**.
3 Select your view.
4 Rename your view or select/unselect the mappings of your choice.
5 Click **Modify**.
6 Click **Close**.

To delete the scenario-diagram view

1 Open your scenario.
2 Select **Scenario/ View manager**.
3 Select your view.
4 Click 🗎.

The tab corresponding to this view disappears from the **Scenario diagram** pane.

Printing the views of a scenario diagram

You can print each view of a scenario diagram.

The printout includes the parameters defined for the schedulers.

To print a view of a scenario diagram

1 Select a view.

Example: **Global**.
2 Perform one of the following actions:
   - Click 🗊.
   - Select the **File/Print scenario diagram** menu.
3 Wait for the print preview window to appear.
4 Click **Print**.

To configure how a view is printed

1 Launch the print of a view.
2 Click **Configure**.
3 Wait for the Window's Configuration panel appears.
4 Change the configuration parameters.

To change the page setup

1 Launch the print of a view.
2 Click **Page setup**.
3 Wait for the **Page setup** dialog box to appear.
4 Modify the page setup parameters:
   - Left margin
   - Top margin
   - Right margin
   - Bottom margin
   - Header margin
   - Footer margin
   - Zoom

   The centimeter is the unit used.
- Page number (**#**)
- Current date (**date**)
- Current time (**H**)
- Full path to the `.scn` file

The **Header margin** and **Footer margin** buttons display new dialog boxes that let you define the information printed in the header and footer of your page.

The **A** button enables you to select a character font for the printed information.

Scenario diagram views - How they work

Scenario-diagram views obey the following rules:
- It is not possible to drag a component inside a view other than **Global**.
- Deleting a mapping in a view does not delete the <LAYER> element in the corresponding SCN file.
- You can only edit mappings, connectors and links in the **Global** view.
- Clicking a tool in a view filters the its associated tracking lines in the Document log.
- Views do not have an impact on the other Connect-It functions.

Menus

This section describes the:

- Menu toolbar in the Scenario builder.
- Shortcut menus in the Scenario diagram.

Menu toolbar in the Scenario builder

File menu

The following table details the commands available in the **File** menu.

Table 3.5. Description of the File menu commands

Command	Function
New	Launches the 'Configure the scenario' wizard. This wizard enables you to choose and configure a source and a destination connector. It links them and places a mapping box between them thus creating a new scenario.
Open	Opens a scenario file (.scn files) that has previously been created.
Save	Saves the current scenario.
Save as	Saves the current scenario under a different name than the one with which it was opened.
Scenario history	Displays a window in which you can enter information about the current scenario.
Print scenario diagram	Displays a window in which you can enter information about how to print your scenario diagram.
Copy scenario diagram	Copies the image of your scenario diagram into the Clipboard. You can then paste this image in a graphical application like Jasc Paint Shop Pro, Adobe Photoshop, etc.
Create scenario documentation	Enables you to create a documentation of the current scenario in a XML, DBK or HTML format.
	For more information about this function, refer to the chapter Scenario documentation [page 185], section Viewing a scenario documentation in HTML format [page 187].
View the HTML scenario documentation	Enables you to display a documentation of the current scenario in an Internet browser.
	For more information about this function, refer to the chapter Scenario documentation [page 185], section Viewing a scenario documentation in HTML format [page 187].

Command	Function
Properties of the scenario documentations	For more information about this function, refer to the chapter Scenario documentation [page 185], section Editing the scenario documentation properties [page 188]. Launch the 'Properties of the scenario documentations' window.
Exit	Enables you to leave the Scenario builder.
Enumerated list	Presents the most recently built scenarios. By clicking one of the elements in the list, you can directly open a scenario.

Edit menu

The following table details the commands available in the **Edit** menu.

Table 3.6. Description of the Edit menu commands

Command	Function
Cut	None
Copy	Copies lines from the Connect-It log or the Document log to the Clipboard.
Paste	None
Delete	Clears a previously selected tool from the current scenario.
Options	Displays the Options window in the Scenario builder.

Display menu

The following table details the commands available in the **Display** menu.

Table 3.7. Description of the Display menu commands

Command	Function
Toolbox	Displays or hides the pane containing the list of connectors.
Scenario diagram	Displays or hides the Scenario diagram pane.
Toolbar	Displays or hides the general toolbar.
Diagram bar	Display or hides the diagram's toolbar
Favorites sidebar	Displays or hides the favorites sidebar.

Favorites menu

The following table details the commands available in the **Favorites** menu.

Table 3.8. Functions of the commands on the Favorites menu

Command	Function
Configure	Configures the favorite associated with the connector once this connector is defined
Delete	Deletes the favorite associated with the connector

Command	Function
Export	Exports the favorites associated with the scenario
Import	Imports the favorites from the current folder

For further information on using folders, refer to chapter Favorites [page 207].

Scenario menu

The following table details the commands available in the **Scenario** menu.

Table 3.9. Description of the Scenario menu commands

Command	Function
Open all connectors	Opens all the connectors and mapping boxes in the current scenario.
Close all connectors	Closes all the connectors and mapping boxes in the current scenario.
Synchronize the caches	Synchronizes the cache files of all open connectors in the current scenario.
Delete the caches	Deletes the cache files of all open connectors in the current scenario.
Manage views	Displays the view manager.
	To learn more about managing views, refer to this chapter's section The Scenario builder [page 39]/ Scenario-diagram views [page 43].
Schedulers	Displays the schedulers edit window.
Scheduling	Starts the associated scheduling.
String tables	Displays the string-table edit window.
Maptables	Displays the maptables edit window.
Global functions	Displays the global-functions edit window.
Script constants	Displays the script constants editor
User formats	Enables you to create date-type and numeric-type formats that you can use in mapping scripts.
Work offline	Enables you to work offline.
	This option can be activated by clicking in the toolbar.
	For more information about working offline, refer to the chapter Testing and debugging an integration scenario [page 212] , section Working offline [page 243].
Test mode	Enables you to test the current scenario without needing the destination connectors to transmit to the external applications.
	This option can be activated by clicking in the toolbar.
Start all schedulers	Starts all schedulers associated to the document types produced by the current scenario's source connectors.
Stop	Stops the scenario.

Tools menu

The following table details the commands available in the **Tools** menu.

Table 3.10. Description of the Tools menu commands

Command	Function
Configure	Launches the configuration wizard of the selected tool.
Advanced configuration	Enables you to display the advanced configuration pages in the configuration wizards (F2).
	This option can be activated by clicking 🔳 in the toolbar.
Modify the relational model	Enables you to define the structure of the document types produced by the connector.
	For further information, refer to chapter Edit the configuration file [page 67]
Open	Opens the selected connector.
Close	Closes the selected connector.
Cache/Synchronize the cache	Synchronizes the selected connector's cache file.
Cache/Delete the cache	Deletes the selected connector's cache file.
Wizard/Manage processing report	Enables you to create a feedback loop for a given connector.
	For further information, refer to section Manage processing reports wizard [page 233].
Edit document type	Launches the **Select document type** wizard that enables you to choose a document type produced or consumed by the selected tool.
Edit mapping	Launches the **Select mapping** wizard that enables you to choose one of the mappings from the selected mapping box.
Produce now	Enables the selected tool to produce documents for which a document type has been created.

Monitors menu

The following table details the commands available in the **Monitors** menu.

Table 3.11. Description of the Monitors file commands

Command	Function
Configure monitors	Displays the window that enables you to configure the document log (Document log, Database, Files, E-mail)
Refresh tracking lines (F9)	Enables you to refresh the contents of the document log.
Clear displayed tracking lines	Clears the displayed tracking lines from the **Document log** tab.
Manage saved sessions	Displays the window showing all the saved sessions for the active scenario.
Import saved sessions	Imports saved sessions into the current scenario.
	For further information, refer to section Importing saved sessions [page 166].

Administration menu

The following table details the commands available in the **Administration** menu.

Table 3.12. Description of the Administration menu commands

Command	Function
Edit authorization certificate	Displays the authorization certificate's edit window.
Administer the monitors database	Displays the window enabling you to administer the database used by monitors.

Java menu

The following table details the commands available in the **Java** menu.

Table 3.13. Description of the Java menu commands

Command	Function
Configure JVM	Displays a window in which you can specify the path of your Java classes.

Help menu

The following table details the commands available in the **Help** menu.

Table 3.14. Description of the Help menu commands

Command	Function
Online help	Displays Connect-It's online help.
Tip of the day	Displays the Tip of the Day window shown at startup of Connect-It. The **Show at startup** option can be cleared.
About	Displays the **About** window that contains general information about your version of Connect-It.

Shortcut menus in the main window of the Scenario builder

In the main window of the Scenario builder, shortcut menus are available in the following situations:

- When you place your pointer in the pane of the Scenario diagram.
- When you place your pointer in the Connect-It log tab.
- When you place your pointer in the Document log tab.

When you place your pointer in the pane of the Scenario diagram

The following table details the list of commands in the shortcut menu that appears when you place your pointer in the Scenario diagram pane.

Command	Function
Configure connector	Launches the **Configure the connector** wizard that enables you to configure the selected tool.
Open connector	Opens the selected connector.
Close connector	Closes the selected connector.
Favorites/Add to the favorites without association	Adds the connector configuration to the favorites without associating it
Favorites/Add to the favorites with association	Adds the configuration of the connector to the favorites and associates it. Reconfiguring the favorite will reconfigure the associated connector in all scenarios.
Cache/ Synchronize the cache	Synchronizes the selected connector's cache file.
Cache/ Delete the cache	Deletes the selected connector's cache file.
Edit a document type.	Launches the **Select document type** wizard that enables you to choose a document type produced or consumed by the selected tool.
Edit a mapping...	Launches the **Select mapping** wizard that enables you to choose one of the mappings from the selected mapping box.
Produce now	Enables the selected tool to produce documents for which a document type has been created. This function is only available if at least one of the connectors is capable of producing data. If several connectors are capable of producing data, an error message is displayed.
Delete	Deletes the selected tool.
Show tracking lines	Displays the tracking lines relating to the selected tool in the Document log.
Show toolbox	Shows or hides the Toolbox.
Show scenario diagram	Shows or hides the Scenario diagram.

When you place your pointer in the Connect-It log tab.

Command	Function
Unfold one level	Unfolds the selected level of the tracking lines.
Unfold all levels	Unfolds all the levels of the tracking lines.
Collapse all levels	Refolds, or collapses, all levels of the tracking lines.
Show toolbox	Shows or hides the Toolbox.
Show scenario diagram	Shows or hides the Scenario diagram.

When you place your pointer in the Document log tab

Two shortcut menus are available in the Document log tab:

- When you place your pointer on a tracking line.

- When you place your pointer on an element in the detail of a source document.

When you place your pointer on a tracking line

Command	Function
Copy this XML document	Copies the XML document that corresponds to the tracking line on your computer's Clipboard.
Copy this XML DTD	Copies the XML DTD of the document corresponding to the tracking line on your computer's Clipboard.
Open this XML document	Opens the document with the application designated to open XML files, which is specified on your operating system (for example: Internet Explorer).
Show toolbox	Shows or hides the Toolbox.
Show scenario diagram	Shows or hides the Scenario diagram.

When you place your pointer on an element in the detail of a source document

Command	Function
Copy this XML document	Copies the XML document on your computer's clipboard.
Copy this XML DTD	Copies the document's XML DTD on your computer's clipboard.
Open this XML document	Opens the document with the application designated to open XML files, which is specified on your operating system (for example: Internet Explorer).
Copy path	Copies the path of the selected element to your Clipboard.
Find	Enables you to find elements in a document detail in the document log.
Find next	Finds the next element
Find prev.	Fins the previous element
Change the column sort order	Sorts by the ID, date, document type, connector or the contents of the document
Show toolbox	Shows or hides the Toolbox.
Show scenario diagram	Shows or hides the Scenario diagram.

Shortcut menus accessible from the mapping box

You can access the following menus from the mapping box:

- Script constants
- Global functions
- User formats
- Change history
- Options
- String tables
- Mapping tables

The following tabs are available, depending on the element chosen in the mapping box:

- Root document:
  - General tab
  - Mapping tab
  - Additional scripts tab
  - Reconciliation tab
- Complex element of type field:
  - Mapping tab
  - Reconciliation tab
  - Reconciliation scripts tab
- Complex element of type structure:
  - Mapping tab
  - Reconciliation tab
- Complex element of type collection:
  - Mapping tab
  - Reconciliation tab
  - Advanced reconciliation tab

4 Implementing an integration scenario

Implementing an integration scenario consists of:

- Choosing the tools.
- Linking the tools.
- Create a new configuration file if necessary
- Define the document types produced and consumed

Implementing the scenario with a wizard

The **Scenario implementation** wizard enables you to:

- Choose the source and destination connectors for your scenario.
- Perform the configuration of these connectors.

After the configuration of your destination connector is finished, the two connectors are linked to a mapping box. The connectors and the mapping box appear in the Scenario diagram pane of the Scenario builder.

To launch the Scenario implementation wizard

1 Perform the following actions:

- Click in the Scenario builder toolbar.
- Choose **New** from the **File** menu.

Manually implementing a scenario

To manually implement a scenario you must:

1 place your scenario's tools (connectors and mapping box) in the Scenario diagram

2 Link the tools.

Placing the tools in the Scenario diagram

Placing the tools in the Scenario diagram automatically launches the **Configure the connector** wizard.

To place a tool in the Scenario diagram

1 Drag it from the Toolbox into the Scenario diagram or selection it and double-click on it.

2 Configure it by entering values in the **Configure the connector** wizard that appears.

For information about configuring connectors, refer to the *Connectors* guide, chapter *Configuring connectors*.

Figure 4.1. Toolbox in the Scenario builder

To reconfigure a tool already in the scenario diagram

1 Select it in the Scenario diagram.

2 Perform one of the following actions:

- Select the **Tools/ Configure** menu.
- Right-click and choose **Configure connector** from the shortcut menu that appears.

Linking the tools

The way in which you link the tools enables you to define which one(s) will become source connector(s) and which one(s) will become destination connector(s).
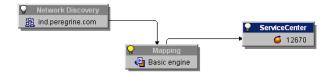
- A link leaving the connector indicates that it is a source connector.
- A link entering the connector indicates that it is a destination connector.

Note:

A source connector is almost never linked directly to a destination connector. It almost always goes through a mapping box, which, itself, is linked to the destination connector.

In the following example, the Network Discovery connector is the source connector, and the ServiceCenter connector is the destination connector. When the Network Discovery connector produces documents, these documents are consumed by the ServiceCenter only after having been transformed by the mapping box.

Figure 4.2. Linking tools in a scenario
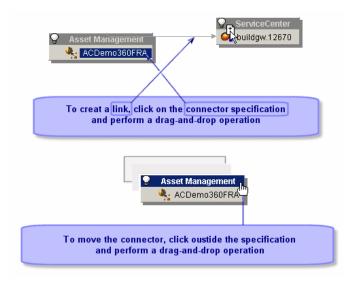
To link a tool to another tool

1 Click on the lower part of the tool.
2 Wait for the tool's specification to appear in white.
3 Keep the left mouse key held down.
4 Drag the link that appears as a line to the other tool.

Figure 4.3. Creating a link or moving a connector

To link two connectors without going through a mapping box

When you link the two connectors, Connect-It automatically creates a mapping box between these two connectors. If you don't want this mapping box to be created, link the two connectors while holding down the **Shift** key.

The direct link between two connectors applies to certain connectors (such as the Delimited-text connector and the XML connector). This link enables you to directly export data from the source connector using the structure of the document produced by the source connector.

Deleting a tool

To delete a connector, a link or a mapping box in the Scenario diagram:

1 Select the tool.
2 Perform one of the following actions:
    ▪ Select the **Edit/ Delete** menu.

- Press **Del** on the keyboard.
- Right-click and select **Delete** from the shortcut menu that appears.

Configure the Java virtual machine (JVM)

The JVM used by Connect-It can be configured directly in the Scenario builder.

The configuration defined in the JVM is valid for all scenarios, but does not override the special configuration of a given scenario's components.

To configure the JVM

1 Launch the Scenario builder.
2 Select the **Java/Configure JVM** menu.
3 Enter the path of your Java classes in the **Classpath** field (jar or zip files) or the folder containing your .class files.
   - To select a Java library (.jar), click the **File** icon
   - To select a class folder, click the **Folder** icon
   - To add a Classpath element, click the **Plus** button.

   > ✎ Note:
   >
   > The order is significant in the Classpath definition.

4 Enter your JVM options in the **Options** field.

   Example: *-DmyDefine=value*.

5 Activate or deactivate the **Debug mode**.

   This option enables you to display the debug messages (**PrgnLog.debug**) and the stack traces (*stacktrace*) of the exceptions.

6 Activate or deactivate the **Specify the JRE to use** option. This option modifies the conit.ini file and the values are taken into account the next time the virtual machine is started.

   By default, the JRE used by Connect-It is the default JRE specified in the Windows Registry and the **Specify the JRE to use** option is not selected.

   The following fields are activated by this option:
   - JRE saved in the Registry: select this field to validate the version number of the JRE defined in the local machine's Registry. IBM JREs are prefixed with the value (IBM).
   - Path of the JRE dynamic library: populate this field if you want to use a JRE that is not registered in the Registry. This library is usually called jvm.dll. In this case, you are advised to also specify the JRE root folder (JAVA_HOME).

In some cases, the JRE specification includes additional constraints:

- No JRE is recorded in the Registry. In this case, the user must specify the JRE to be used by indicating the path of the JNI dynamic library to be used. The **Specify the JRE to be used** option is then checked and is read-only, as is the **Path of the JNI dynamic library of the JRE** option.

- No default JRE is specified in the Registry: in this case, Connect-It cannot determine the default JRE to use. Therefore the user must specify it. The **Specify the JRE to use** option is then checked and is read-only.

Additional information

- The Sun JREs are recorded in the Registry with the following key: HKEY_LOCAL_MCHINE/SOFTWARE/JavaSoft/Java Runtime Environment.

- IBM JREs are stored in the Registry under the following key: HKEY_LOCAL_MACHINE/SOFTWARE/IBM/Java2 Runtime Environment.

Edit the scenario options

Options are available in the Scenario builder for the following categories:

- **Display**
- **Confirmations**
- **Connector**
- **Documents**
- **Logs**
- **Lists**

To access the options window in the Scenario builder, choose **Options** from the **Edit** menu.

In this window, you can modify the options by double-clicking on the cell corresponding to your choice in the **Value** column. Enter the desired value in the text zone that appears.

Note:

When you modify one of the default values, the title of this option will appear in red. This enables you to quickly locate the options for which you have changed the value.

Display

This category regroups the display options for the Scenario builder.

Display reconciliation properties by document type

This option enables you to display the reconciliation properties for a document type element. The view differs depending on the complex element displayed:
- For the root node of a document type, complex elements with reconciliation keys (field, collection or structure) are displayed
- For complex elements of a collection or a structure, all their elements with reconciliation keys are displayed.

Display licensed connectors only

This option displays the connectors that your authorization certificate grants you permission to use in the Toolbox of the Scenario builder.

Change the color of wizard descriptions

This option enables you to change the color of the wizard descriptions. Example: You can change the color of the text that appears in the connector configuration wizards.

Disable the 'Esc' in fields

In Windows, using the **Esc** key enables you to leave an edit window. Example: the document-type edit window.

For security reasons, the **Esc** key in the Scenario builder only applies to the editing of the current element. Example: the editing of a collection-to-collection mapping in the mapping edit window.

To return to the default behavior in Windows, select this option.

Windows graphics

This option enables you to choose from either a flat or standard presentation of the Scenario builder.

Number of documents recovered in a block during the preview

This option enables you to set the number of documents to display in the document preview window. By default, the number of documents is set to 20.

Maximum number of child elements of an element to be unfolded automatically

This option sets the number of child nodes that can appear under a parent node in the documents and document types. This option is useful when certain nodes in your document types contain too many child nodes.

Maximum number of child elements displayed at one time

This option enables you to set the number of nodes appearing under parent nodes in documents and document types. Example: to only show the first three collections of a document type in the root node.

Maximum number of lines to display in the Connect-It log

This option enables you to limit the number of tracking lines appearing the Connect-It log.

Maximum number of levels (recursive) to map automatically

This option enables you to define the number of node levels (structure or collection) that can automatically be mapped during a table-to-table or node-to-node- mapping.

Tab

You can change the appearance of tabs in Connect-It by modifying the following options:

- **Display image**

  This option enables you to choose whether or not to display the icon that accompanies a tab name. Example: the 🔍 icon for the **Document log** tab.

- **Display ToolTip**

  This option enables you to activate the ToolTips that are displayed when you place the mouse pointer over a tab.

- **Tab representation style**

  This option enables you to change the look of the tabs in the produced document-type window.

Display ToolTip

This option enables you to activate the ToolTips that are shown when you place the mouse pointer over an object in Connect-It.

Confirmations

This category represents the confirmations that are asked of the user after having performed certain operations. The confirmations function by using dialog boxes.

Display a warning whenever a collection-to-collection mapping is needed

This option displays a warning every time that a collection-to-collection mapping is necessary. Example: The fields of a source collection are mapped to the fields of a destination collection without the two collections being mapped to each other. In this case, a warning appears reminding the user that these collections must be mapped.

Display a message each time the JVM parameters are modified

This option enables you to have a message appear each time the JVM (Java Virtual Machine) parameters are modified.

Warn when the license is approaching expiration

When this option is selected, Connect-It will warn you when your license is approaching expiration. You can specify the delay before the warning is issued. If this delay is set to zero, a warning will be issued constantly.

Keep the backup copy of the scenario

When saving a scenario, a backup copy with the `.bak` extension is temporarily created in the scenario folder. This option enables you to keep a copy or to delete it if the current save operation is successful.

Disable the new scenario wizard

This option enables you to disable the wizard that is displayed automatically when creating a new scenario.

Do not authorize the validation of a script that includes compilation errors

This option prohibits you from validating a script that contains syntax errors.

Open all connectors of current scenario when starting a connector

This option opens all the connectors of the current scenario at connector startup. This option is especially useful because it enables you to quickly activate a scenario's mappings without having to open all the connectors one at a time.

Use a dialog box to select the document type to edit

This option displays a dialog box in which you select a document type or a mapping after having double-clicked on an open connector or mapping box.

Connectors

Options are available for the following connectors:
- Database connector
- LDAP connector
- ServiceCenter connector
- XML connector and Delimited-text connector

Other options are available for all connectors:
- **Show queries in tracking lines**

  This option displays your queries in the document log (messages pane). This is useful when testing/debugging your scenario.

- **Calculate number of documents to be processed**

  This option displays the number of documents processed in the status bar. Example: If four documents have been processed out of 200, *4/200 documents processed* is displayed in the status bar.

  > ⚠ Warning:
  >
  > This operation can compromise your computer's performances if there are too many documents to process.

- **Adjust dates in relation to the time difference with the server.**

  This option enables you to compensate for the time difference with a server for dates read and written.

  > 📝 Note:
  >
  > This adjustment was automatic through Connect-It 2.7.1.
  >
  > For more information about the server delay, refer to the *Connectors* guide, chapter *Configuring connectors*, section *Time difference with the server*.

  If this option is not activated, each time that you specify a server delay during connector configuration, the dates read or written are adjusted to take into account this delay.

- **Favorites file**

  This option enables you to define the favorites file used by default.

- **Number of documents to produce in a test production of documents**

  This option enables you to set the number of documents to produce when you test document production. (In this test mode, the scenario's source connectors produce documents, but the destination connectors do not interact with any external applications.) To limit this number, you must check the box in this option's row.

  This option is automatically taken into account when you test a produced document-type. On the other hand, when the scenario is executed as a Windows service, this option is automatically deactivated.

- **Maximum number of documents in the reconciliation cache**

  When the reconciliation cache is used in a mapping, this option lets you empty the cache when the maximum number of documents has been reached.

- **Validate by default the option to use a configuration wizard cache.**

  This option enables the default use of caches when a new connector is created. Enabling the cache saves the document types that were published or consumed by a connector and therefore avoids performing an autodescription each time you open the connector.

For more information about cache options, refer to the *Connectors* guide, chapter *Configuring connectors*, section *Configuring the cache*.

LDAP connector

The option **Read LDAP server schema from file** enables you to make a local copy of the remote server's description schema.

The **Display attributes defined at the DSE root** option enables you to determine if the LDAP server exposes an attribute which lets you recover the current server date.

The option **Save the LDAP server schema to file** enables you to save a local copy of the remote server's description schema.

The **Certificate** option enables you to define the access path of the certificate associated with the LDAP server.

The **Certificate folder** option enables you to specify the folder containing the certificates enabling secure access to the LDAP server.

ServiceCenter connector

The option **Display a warning for events used in asynchronous mode** enables you to warn the user on opening a scenario and for each document processed, that the scenario works in asynchronous mode.

The **Check data type in filter** option enables you to make sure that the value type of the ServiceCenter fields corresponds correctly to the type specified by the author of the request. For example: The sysmodtime > 3 filter in which a **date**-type field is associated with an integer is not valid. An error message will appear and the connector will not process the document.

The **Display progress information while connector publishes document types** option enables you to display the name of document types published by the connector that you opened in the Connect-It log.

The **ServiceCenter code pages** option enables you to indicate the code page number of your choice. By default, the value of this option is set to 0. This value means that the ServiceCenter code page is that of the computer on which Connect-It is installed.

 Note:

This option only applies to ServiceCenter versions 3.x, 4.x, 5.x or lower. In ServiceCenter 6 and higher, multilingual support with internal UTF-8 character coding is available. To handle a language other than English, French or Japanese (or Windows1252-compatible languages in general), you must use a Unicode English-language version of Connect-It. This version is available from customer support after validation of usage (language and connectors used).

Database

The **Display a message when the data type of a field is not supported** option enables you to display a message in the Document log when a field is not supported by a connector.

Delimited text and XML

The **Display the URL being processed in the Connect-It log** option enables you to display in the Connect-It log the URL being processed by the XML or Delimited text connector.

The option **Copy locally the files to read from FTP server** is usful if the connection cannot be maintained for the time required to process the files.

The option **Backup the existing file before starting in mode 'Append to the same file'** enables you to make a duplicate copy of the file between two Connect-It sessions.

Documents

Attention: The documents of this category are not the documents processed by the connectors but the SCN files that contain your scenarios.

This category regroups the options for loading your .scn files:

- **At startup, automatically load last document used**
- **Default path of scenarios**

  This option enables you to specify the default path of the scenario folder

- **Ask for confirmation before saving**

  This option enables you to display a dialog box before any save operation in Connect-It (saving a mapping, scenario, scheduler, etc.).

- **Maximum number of recently opened documents to store in File menu**

  This option concerns the list of SCN files appearing in the list of the **File** menu.

Logs

The options associated with logs are the following:

- Write a Java log file

  This option enables you to create a log file containing all the messages generated by the Java connectors. This log file has the same name as the application and carries the suffix _java.

- Log file

This option enables you to specify the name of the log file.

- Verbose mode

This option enables you to activate the verbose mode.

> **Note:**
>
> This option is not taken into account if the application is launched with the '-verbose' option

- **Log-file size (MB)**

This option enables you to set the size of your `.log` files. When this size is reached, new data overwrites the oldest data.

Lists

The options associated with logs are the following:

- Auto-close delay after selection (milliseconds)

This option enables you to set the delay for the automatic closing of drop-down lists. 0 disables automatic closing.

- Refresh delay after keyboard selection (milliseconds)

This option enables you to delay refreshing when selection is made with keyboard so as to not slow down navigation in lists.

- Arrow keys used to open and close nodes in tree structures

This option enables you to use the arrow keys to open and close nodes in tree structure. To navigate from one column to another using the keyboard, hold down the SHIFT key.

Edit the configuration file

Some connectors include an extension file for extending or modifying the documents displayed by the connector. In the case of a Database connector, the extension file allows for example to define the database's relational model and to define schedule pointers.

The connector configuration-file editor enables you to create or customize a relational database model.

The configuration-file editor applies to :

- the Database connector
- inventory connectors

You can access the configuration-file editor using the shortcut menu (right-click the connector then select **Modify the relational model**). This menu is available once the connector has been configured.

> ⚠ Warning:
>
> If you modify the relational model of a connector that is already used in one or more scenarios, the scenarios will be impacted and will need to be modified.

The file generated by the configuration-file editor has the file extension `.cfg`.

Limitations of the configuration-file editor
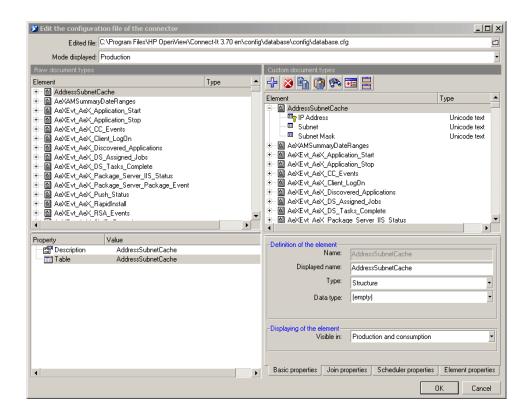
This editor does not enable you to edit the complex `amo.cfg`, `iInventory.cfg`, `ist.cfg` and `tscdb.cfg` files.

The editor window

There are two parts to the editor:

- Right part: The relational database model

■ Left part: Customized document types from the database, produced by the connector

The properties of the selected element are defined in the following tabs:

■ Basic properties

  This tab enables you to define the displayed name of the complex element, its type, its data type (binary, text, etc.) and the display conditions.

■ Join properties

  This tab enables you to define the join and filter the element.

■ Scheduler properties

  This tab enables you to define the scheduling conditions of the element.

■ Element properties

  This tab lists all the properties of the selected element.

Create a new document type

When opening the configuration-file editor, the source database structure is displayed in the left side of the window. This structure is automatically duplicated in the right side, which shows the document type that will be published by the connector.

The document type resulting from the source database is called a raw document. The document type modified and published by the connector is called a customized document.

To create a new customized document type, edit the elements shown in the right side:

- To delete an element in a customized document type, select it and then click ⊗ or press Del.
- To insert an element from a raw document type to the root position of a customized document type, drag the element while holding the CTRL key down.
- To create joins between items, drag an element from a raw document type to the desired element in the customized document type (from the left side to the right side of the window) or drag an element in the customized document to another element in the customized document type (action performed on the right side only).

Note:

All modifications can be reversed using the **Restore the element** command from the shortcut menu (right-click the root element in the customized document type).

All the modifications are validated when:

- a new name is given to the configuration file (**Edited file** field)
- the configuration-file editor is closed (**OK** button)

Customizing a document type

Cutomizing a document type consists of defining properties for each of the complex elements making it up.

Note:

Use the 譜 button to split the window in two and view the source and target elements at the same time.

A customized element is symbolized with the ☞ icon.

All modifications made to an element can be reversed using the **Restore the element** command from the shortcut menu (right-click the root element in the customized document type).

Basic properties of a customized document type

- Name: System name of the element (field, structure or collection). This field may not be edited.
- Displayed name: This field enables you to change the displayed name of the element (field, structure or collection) and to define its place in the list. The system name is unaffected.
- Type: This field enables you to define the type of a complex element (field, structure or collection). When a structure or a collection is defined, make sure you check the associated join.
- Data type: This field enable you to define the data type of the element. The following data types are available: Binary field, long text field (memo), Long Unicode text field, date, date and time, time, double-precision number, 8-bit integer, 16-bit integer, 32-bit integer, floating point number, text (size limited to 255 characters) and Unicode text.

  This field enables you to specify the data type if the database engine does not recognize it.
- Visible in: This field enables you to define the display conditions of the element, i.e. if it is visible in production, consumption, or consumption and production. The displaying of the element depends on the value defined in the configuration field (**Mode displayed** field visible below the **Edited file** field).

Join properties of a customized document type

The ▣ icon designates elements for which a join can be used. These elements are listed in the produced document-types edit window and the document-type customization window.

The join is the linking of two document-type elements with a hierarchy being created. In order for the join to be possible, a child element must exist in the two document types. For example, to recover a portfolio item user from AssetCenter, the *lUserID* field of the *amPortfolio* document type is common to the *amEmplDept* document type.

> ### ✎ Note:
>
> This join is given as an example only. You should not use the Database connector to connect to an AssetCenter database.

To perform a join:

1 drag a document type onto another

2   select the structure that is shown

3   change the structure to a collection, if required

4   in the **Join properties** tab, click **Suggest**

5   modify, if required, the join condition expressed (⊠ button)

6   validate the join condition (✓ button)

Join syntax

The join syntax is as follows:

```
<expression><operator><expression>
```

For example:

```
{..Device_ID}*={Object_ID}
```

This example shows the syntax used for a join between the Device table and the Enterprise Discovery NMID table.

An expression may be:

- a relative field ({<relative path of a field>})
- a constant

  Constants must be between single quotes (' '), be they strings, numbers, or dates. Only the keyword 'null' must not be between single quotes.

  ```
  {DeviceId}=null
  ```

The operator may be a comparison operator (=, *=, >, >=, <, <=)

💡 Tip:

To avoid making errors in writing the relative paths, you can drag the fields from the right in order to write the join condition.

✏ Note:

Certain *= queries may fail if the operator is not supported by the database engine, or simply reduce processing performance.

Sample queries

- This query returns all the unique IDs starting with 'demo'

  ```
  {DeviceId} like "demo%"
  ```

- This query returns all the unique IDs that do not start with 'demo'

  ```
  not({DeviceIdd} like "demo%")
  ```

Join sub-query

The **Perform a sub-query** option enables you to force a sub-query to be executed.

This option is particularly useful if the query becomes too large for the database engine because of too many 1-1 joins. In this case, you may force sub-queries in order to break down the main query. Select this option if the database engine does not support outer joins.

Sort expression

Use this control to specify the order in which the query will return the data.

Data can be sorted by:

- Ascending order (default behavior)
- Descending order (**DESC** instruction)

The syntax is as follows:

```
{<relative path1>} [desc][, {<relative path2>} [desc]...]
```

If no instructions are provided after the relative path, an ascending sort is performed. For a descending sort, use the **desc** instruction.

In the following example, a date type field named DateSold is used. The records are sorted by ascending order using the DateSold field (from the oldest to the most recent date).

```
{DateSold}
```

In the following example, a text type field named Name is used. The records are sorted by descending order using the Name field (from Z to A).

```
{Name} desc
```

Scheduler properties

A pointer is associated with a scheduler property. This property is visible when a date and time type field is defined as a scheduling pointer.

Schedule pointer

Defining a scheduling pointer means defining the reference element for schedulers in Connect-It.

To define a scheduling pointer:

1 select the field to be used as the scheduling pointer

  This field must be at the root of the document type and not included in one of its complex elements (structure or collection).

2 right-click and the select **Pointers/ Define** from the shortcut menu.

The ✐ icon is shown next to the field, symbolizing the scheduling pointer and the **Scheduling properties** tab is enabled.

3 in the **Pointer expression** frame, define the scheduling query (a WHERE clause) that will allow a filter to applied on the field.

The syntax used for the scheduling query is similar to the one used to define a join:

```
{CreationDate} >= "[SWLastScanDate]"
```

In this example, the value of the CreationDate field (defined between braces) is compared with the value (defined between quotation marks) of the schedule pointer (defined between brackets and containing the name of the document type followed by the name of the field).

You can use the **Suggest** button to view a sample of the syntax.

Define the produced or consumed document-types

A produced or consumed document-type is the template that a connector uses to produce or consume documents when a scenario is running. A produced or consumed document-type contains a selection of fields from a document type published by the connector.

Example :

The **amPortfolio** document-type consumed by the ed\ed2ac44\ediac.scn scenario's Asset Management connector only contains a small number of fields that are found in the **amPortfolio** document-type published by the same Asset Management connector.

Creating a document type produced or consumed by a connector consists of:
- Selecting a document type from among the document types published by the connector.
- Selecting fields among those of the chosen document type.

Example :

For example: You select **amProduct** from among the document types published by the Asset Management connector. Next, you select fields that interest you: the product barcode, its brand, price, and catalog reference, etc.

Description of a document type

XML and DTD files
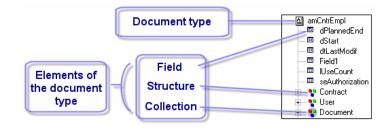
The documents used in Connect-It correspond to XML (eXtensive Markup Language) files, and the document types that these files use as templates correspond to DTDs (Document Type Definition).

The Connect-It Scenario builder enables you to transform all documents into XML files and all document types into DTDs.

Document type elements

Each document type is in the form of a tree structure composed of simple, self-contained elements (fields) and complex, unfolding elements like root nodes, structures and collections, which contain other simple and complex elements.

Figure 4.4. Document type elements

The simple elements are:

■ Fields (⊞)

Fields contain a particular type of data: numbers, long integers, fixed length text, dates, etc.

■ Blob-type fields (⊞)

Blob-type fields contain binary objects whose contents must be stored (binary large objects): images, sound files, video, etc.

■ Memo-type fields (⊟)

Memo-type fields contain variable-length text.

The complex elements are:

■ Structures composed of simple or complex elements (⁛).

Example 1: In a document corresponding to a table in a database, each structure within corresponds to another table. A structure indicates that the link between the two tables is 1:1. A link of this type means that a record in the first table can only be linked to one record in the other table.

Example 2: In a document type corresponding to a received e-mail message, a structure corresponds to the information about the author of the message. (A message can only have one author.)

- Collections composed of simple or complex elements (🌐).

Example 1: When a document type corresponding to a database table has a collection, this collection corresponds to another table in the database. A collection means that the link between the two tables is 1:N or N:N. This type of link means that a record in the first table can be linked to one or more tables in the second table, and visa versa.

Example 2: In a document type corresponding to a sent e-mail message, a collection corresponds to the message's recipients. (A message can have any number of recipients.)

🖊 Note:

Collections of attributes are collections with only one field. This field is the attribute. Each member of this collection is composed of one element repeated n times.

Before processing, each collection in the document contains an undetermined number of members. It is impossible to know the number of members ahead of time since a collection is composed of n members.

Figure 4.5. Document type collection

Example :

For example: In the *Machine* document type published by the Desktop Discovery connector, which describes a company's computers, the software installed on the computers is represented by a collection. Since you don't know how many software products can be installed on all computers at all times, this data must be represented by a collection with n members. Each member of this collection describes the software: name, developer, version, etc.
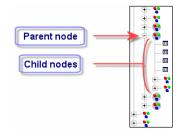
In a tree structure, each unfolding element has a child element. In the document-type description, we say that a parent node contains a child node.

Note:
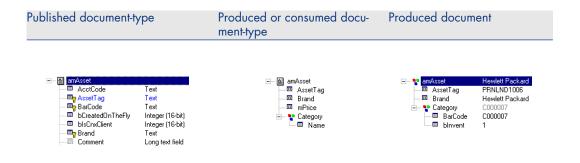
During document processing, the concept of parent and child nodes is important. For example, a child node that contains a rejected field can lead to the rejection of the parent node to which it belongs.

Figure 4.6. The tree structure of a document type

Processing a document type

The following table shows the relationship between the different types of documents processed in Connect-It.

Published document-type	Produced or consumed document-type	Produced document

Published document-type	Produced or consumed document-type	Produced document
Configuration	**Creating a document type**	**Running the scenario**
The connector publishes document types from which you will select one to use in the integration scenario.	Certain elements from a published document-type are selected to create a produced or consumed document-type.	The connector produces or consumes document instances.

Accessing the edit window of the produced or consumed document-types

The edit window of produced or consumed document-types is divided into two zones:
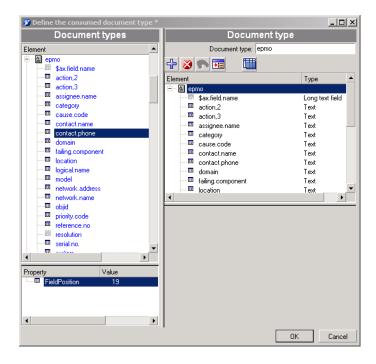
- A work area on the right in which you created the produced or consumed document-type.

  Under this area are tabs corresponding to directives. Some connectors do not have directives.

- A zone on the left showing the document types published by the connector. This zone is split into two panes:

  - A pane where the root nodes of published document-types appear (▣). In this pane you can unfold the nodes (root nodes, structures, collections) of your document types to see everything under them. Each node and each field (terminal fields) are described by their names and their types (text, integer, date, etc.). When you select a node in a document type, this node becomes the current selection.

  - A pane showing the properties and values of the current selection. These properties are bits of technical information that can help in understanding the current selection.

  📝 Note:
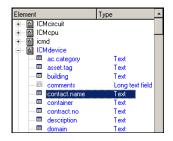
  For certain connectors, for example, the Asset Management or SAP BAPI connector, a description of the selected field is displayed (**Description** property).

Figure 4.7. Editing document types consumed by the ServiceCenter connector

For example: In the following screen, the **contact.name** field in the *ICMdevice* document type is a *text*-type field.

Figure 4.8. Detail of the zone showing the published document-types

1 Select the connector in the Scenario diagram.

2 Select the **Document types** tab or double-click the connector.

3 Select the produced document-types or consumed document-types pane.

4 Click ▯ to create a new document type or ⚲ to edit an existing document type.

Types of documents produced or consumed

There are several types of documents produced:

- Those specific to each connector
- The processing reports

  For more information on processing reports, refer to the chapter Use the processing reports [page 225].

- The feedback documents

Self-produced document

A self-produced document is a document produced by a connector on consuming another document.

This self-produced document is denoted by the icon ▧.

The following connectors use the self-produced document:

- Web Service
- SAP BAPI
- Command line
- MQSeries

If your mapping covers multiple document types, you can choose the document which when consumed will give rise to the self-produced document. Select it from the drop-down list in **Produce this document type on consuming**.

✐ Note:

A connector using a self-produced document type cannot be started directly because it can only be started as the result of a consumption.

Creating a produced or consumed document type

Creating a new produced or consumed document-type involves moving the root node or one or more elements (collection, structure, field) onto the workspace of the document-types' edit window.

To create a produced or consumed document-type

To create a produced or consumed document-type

1. Move the root node or one or more document-type elements onto the workspace by doing the following:

 - Drag and drop it onto the workspace.
 - Select it and click ✛.
 - Double-click this element or this root node.

📝 Note:

If you want to use a large number of fields relating to a node (root node, structure, collection), simply drag that particular node onto the work area. This moves all of the related fields onto the work area at once. From there, you can remove the unwanted fields.

When your selection is made, name your document type (by default, it inherits the name of the published document-type whose node you have dropped on the work area) and then click **OK**.

📝 Note:

When you perform a mapping in a mapping box, all the elements of the document published by the source connector, and used in the mapping, are automatically added to the produced document type. It is often useful to edit the document type in order to:

 - Delete the elements that are no longer used in the mapping.
 - Add elements that are not used in the mapping, but are indispensable anyway.

To modify a produced or consumed document-type

To modify a produced or consumed document-type:

1. Select it and then click 🔍, else double-click it

⚠ Warning:

Modifying an existing document type, and in particular modifying one of its complex elements (collection, structure, field), means modifying the mappings linked to this document type. As a consequence, you must adapt the different mappings.

Modifying a document type involves:

 - Deleting one of its complex elements

   Deleting an element of a produced or consumed document type is useful to improve performance in processing documents, especially when this element is not longer used in a mapping.

- Adding a complex element

  Adding a complex element to an element of a produced or consumed document type is useful when:
  - On scenario execution, these elements are usually those whose values must be recovered in the source to update the schedule pointer of the OpenView Network Discovery connector.

Deleting an element from the work area

1 Select it and click ⊗ or press **Delete**.
2 Or drag it back to the window of published document-types.

Copying the DTD of a produced or consumed document-type

Each document type in Connect-It corresponds to an XML document type. Connect-It enables you to copy the DTD linked to this document.

Copying the DTD of a produced or consumed document type

1 Create or edit a produced document-type.
2 Position your pointer on the work area.
3 Right-click.
4 Choose **Copy this XML DTD** from the shortcut menu that appears.

When you are in the edit window of a document type, you can display a shortcut menu by right-clicking, which lets you copy the DTD to your clipboard.

Using the data generator

The data generator allows to send statically-defined data to a connector.

This internal tool is especially useful for supplying starting parameters to the documents produced on consumption (Web Service connector, ServiceCenter Web Service connector, SAP BAPI connector, and the Command line connector). You can associate scheduling with this tool.

Specific features of the data generator

The data generator can be compared to a connector that only works in production mode, and that does not consume data, unlike this tool which does not expose any of its own document types.

To be able to expose a document type, the data generator must be linked to an open connector.

Behavior of a data generator with a connector

- When you create a link between the data generator and a connector, no mapping box is created.
- A data generator cannot be linked to a mapping box
- A data generator cannot create a produced document type unless it is connected to at least one destination connector
- When editing a document type produced by a data generator, the destination connector must be open

Document type produced by the data generator

When editing a document type produced by the data generator, the left-hand panel exposes the consumed document types of the destination connector. The directives panel exposes the fields specific to the data generator, as well as the consumption directives of the destination connector (for example, the **Reconciliation** tab).

The data generator exposes a production directive (**Value** column) for each element that accepts a value. The value of this directive is the value that will be assigned to the element when the document is produced by the data generator.

You can edit the value of the production directive **Value** via typed editing controls: a number editor for numeric types, a date/time editor for date/time types, a simple text editor for character strings, a multi-line editor for long text fields, and a file selector for blobs (the value of the produced blob will be the contents of the file).
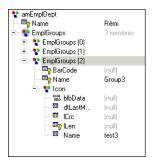
When you delete a document type consumed by a connector, and it is produced by a data generator, a dialog box indicates that the document produced by the data generator will also be deleted.

Duplicate collection

Collections are 1-N type links. When a collection is used in the data generator, it is possible to duplicate it as many times as required in order to populate the same element several times with different values. For example, it is possible to generate information related to the Departments and Employees table in order to test an AssetCenter database. This is done by changing the employee name each time so that each collection uses a different name.

Each duplicated collection is identified as a structure with the name of the collection followed by the duplication number (0, 1, 2, 3, etc.).

To duplicate a collection, right click the collection and select **Add collection element** from the shortcut menu. Each element contained in the collection is assigned a NULL value by default. Each collection or sub-element that is added can be deleted using the **Delete / Cancel an element** command from the shortcut menu.

When documents are produced, you can choose not to produce elements with null values via the **Do not produce null elements** command.

Example of a scenario using the data generator

We are going to create a scenario that queries a weather web service and then use the information supplied by this web service in AssetCenter.

The steps are as follows:

1 Configure the Web Service connector
2 Configure the data generator
3 Configure the AssetManagement connector
4 Define the data sent by the data generator to the web service
5 Define the mapping between the Web Service connector and the AssetManagement connector

Configure the Web Service connector:

1 Drag and drop the connector on the scenario builder.
2 Specify the WSDL address to use (or search for one using the UDDI browser and filter using %weather% as the keyword). The SOAP protocol is used by default.

   In this example we will use the WSDL service from capescience which is available at the following address:
   http://live.capescience.com/wsdl/AirportWeather.wsdl.

3 Validate your selection and click **OK** to complete configuration of this connector.

4   Open the correctly configured connector (right-click the connector and choose **Open connector** from the shortcut menu).

Configure the data generator:

📝 Note:

The data generator can only be connected if the connector that it is linked to is open.

1   Drag and drop the data generator (**Internal tools** toolbox) on the scenario builder.
2   Link the data generator to the Web Service connector.
3   Change the name if needed (right-click the connector and choose **Configure connector** from the shortcut menu).

Configure the AssetManagement connector:
1   Drag and drop the connector on the scenario builder.
2   Specify the database to use on the wizard's **Define the connection parameters** page.
3   Click Finish to validate your configuration.

Define the data sent by the data generator to the web service:
1   Select the data generator
2   Open the data generator
3   Create a new produced document type
4   Select the **getSummary** document and assign the value VHHH to *arg0* (value that corresponds to an airport's ICAO identifier).

    This value will be reused in the mapping for the Asset Management connector.
5   Click OK.

    The data generator is now set to send the weather report for Hong Kong International Airport. The GET instruction will remain the same. A scheduler can be used to execute the instruction on a regular basis in order to update the information sent to AssetCenter.

Define the mapping between the Web Service connector and the Asset Management connector:
1   Link the Web Service connector to the Asset Management connector
2   Create a new mapping:
    ■   Source: Web Service component, Document type: getSummary
    ■   Destination: AssetManagement component
3   For this mapping, the following elements are required and will be used in the mapping: bAllgroup, bChildGrps, dtEnd, dtStart, Msg, Name, Topic, Author.BarCode, Author.IDNo, Author.Name.

The following values are assigned to the following elements:

- bAllGroups: retval =1

  The message broadcast option for all employee groups is enabled.

- bChildGrps: retval =1

  The message broadcast option for all employee sub-groups is enabled.

- bBlink: retval =1

  The blink option for the message is enabled.

- dtEnd: DateAdd(Now(),1800)

  The validity end date of the message is defined based on the current date (time the news was sent) to which 30 minutes is added.

- dtStart: retval = Now()

  The message start date is the current date of the Connect-It server.

- msg:

```
retval = [return.location] & " " & "Humidity: " & [return.humidity]
& " " & "Barometric pressure: " & [return.pressure] & " " & "Sky: "
& [return.sky] & " " & "Temperature: " & [return.temp] & " " & "Visi
bility: " & [return.visibility] & " " & "Wind: " & [return.wind]
```

  The message that is sent contains all the information supplied by the web service separated by a space.
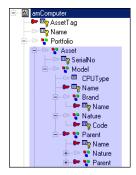
- Name:

```
retval = "WEATHER" & [$ParentDoc$.arg0]
```

  The name of the message is a concatenation of the "WEATHER" character string and the ICAO name (arg0 element) that is defined in the data generator and retrieved using the $ParentDoc$ variable.

- Topic: "Weather"

  A category named "Weather" is created in AssetCenter.

- Author.BarCode: RetVal = "CIT"

  The barcode of the message's author is the Connect-It application.

- Author.IDNo: RetVal = "CIT"

  The ID number of the message's author is the Connect-It application.

- Author.Name: RetVal = "Connect-It"

  The name of the message's author is the Connect-It application.

- Check that your mapping is similar to the following illustration and validate.

The following reconciliation keys are defined:
- Element Name: A reconciliation key is defined for this element because the concatenated name is invariable.
- Author.BarCode: A reconciliation key is defined for this element because the ID of the message's author is invariable.

Defining the mapping behavior:
- For the root element: An **Update or insert** reconciliation type is defined. The first message is created when the scenario is put into production and then the messages are updated.
- For the *Author* structure: An **Update or insert** reconciliation type is defined. The author is created when the scenario is put into production and its information no longer changes.

4 Validate the mapping and back up the scenario.

This mapping:
- Creates a record in the **amNews** table. It contains weather data provided by the web service (included in the amMsg field) and the display format of the message (title, category, groups, sub-groups, display blink option, display start and end dates).
- Creates a record in the **amEmplDept** Employees and Departments table for the message's author.

If a record already exists with the same author and the same category, it is updated with the most recent information; old information is discarded.

Viewing the data of a produced document-type

Connect-It enables you to view the data that corresponds to the elements in a produced document-type. This is the same data that is recorded in the external application linked to your connector.

The data displayed depends on both the document type and connector type:

- For all produced document types and for all source connectors, the data comes from the source database
- For all consumed document types and for all destination connectors, the data comes from the document type available for the destination connector, that is to say the data available in destination database. This functionality is not available for certain connectors.

Note:
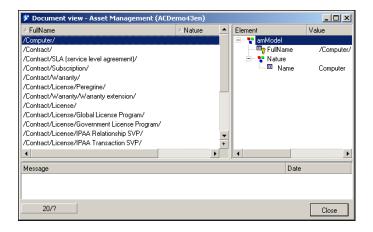
If it takes too long to load the data for viewing purposes, a window will appear asking if you want to stop the current action. If you stop loading the data, you will only be able to view the data that has already been loaded. You can continue the action, though, by clicking ⬆.

Example :

The data in a document type produced by a database connector corresponds to the records of one of the Database tables.

Figure 4.9. Produced-document-type data viewing window

The number of documents loaded in the view window can be set by the **Number of documents recovered in the block during the preview** option (**Display** section), accessible via the **Edit/Options** menu. By default, the first 20 documents are recovered. Click ⊞ to reload a new series of documents. When all the documents are visible, the ⊞ button is grayed out and deactivated.

🖊 Note:

The ⊞ button is not available for all connectors.

Viewing the data in produced document-type

1 Create or edit a produced document-type.
2 Perform one of the following actions:
   - Click 🐾.
   - Position your pointer on the work area, right-click and select **View data** from the shortcut menu that appears.

A document-type data viewing window is composed of two panes:

- The left pane in which appears all the records of the external application that correspond to elements in the document type.

   Each of this pane's columns corresponds to one of the document type's elements.
- The right pane representing the produced document-type, accompanied by the values of the records selected in the left pane.

🖊 Note:

The number of documents displayed depends on the display parameters. For more information on the display parameters, refer to the section Number of documents recovered in a block during the preview [page 61].

To pause the collection of data

1 Perform one of the following actions:
   - Wait for a dialog box to appear. It will ask if you want to interrupt the current query. Click OK.
   - Press the escape button on your keyboard (**Esc**).

Finding an element in the document type

There is a search function that enables you to find elements in a document type without having to unfold all the nodes of this document type.

To find an element in a document type

1. Select a document-type element in the workspace or in the zone of published document-types.
2. Perform one of the following actions:
   - Right-click and select Find (Ctrl + F) from the shortcut menu.
   - Click 🔲.
3. Enter the name of the element that you want to find in the **Find what** field and select the options of your choice:
   - Case sensitive
   - Select the direction of your search: up or down.
4. Click **Find**.

To search through previous elements or following elements:

1. Perform one of the following actions:
   - Right-click and select **Find next** (**F3**) from the shortcut menu.
   - Right-click and select **Find prev.** (**Ctrl + F3**) from the shortcut menu.

🖉 Note:

The search is limited to the elements displayed in the windows of the published document types. If you want your search to apply to the sub-nodes of a given node, you must unfold the node before the search.

Shortcut menus of the document-type edit window

In the document-type edit window, two shortcut menus are available.

- When you place your pointer in the published document-types zone.
- When you place your cursor in the work area.

When you place your pointer in the published document-types zone.

Command	Function
Find	Enables you to display the dialog box for finding a character string.
Find next	Enables you to relaunch the search of the last character string indicated near the bottom of the window.
Find prev.	Enables you to relaunch the search of the last character string indicated near the top of the window.
Copy this XML DTD	Copies the document's XML DTD on your computer's clipboard.
Copy path	Copies the path of the selected element to your Clipboard.
Add this element	Adds the selected element to the produced or consumed document-type that you create.

Command	Function
Filter unused document types	Displays only the produced or consumed document types that are edited in the work area.

Command	Function
Find	Enables you to display the dialog box for finding a character string.
Find next	Enables you to relaunch the search of the last character string indicated near the bottom of the window.
Find prev.	Enables you to relaunch the search of the last character string indicated near the top of the window.
Copy this XML DTD	Copies the document's XML DTD to your Clipboard.
Copy path	Copies the path of the selected element to your Clipboard.
Remove this element	Removes the selected element from the produced or consumed document-type that you create.
View data	Enables you to view the data corresponding to the elements of a produced document-type.
Filter unused document types	Displays only the produced or consumed document types that are edited in the work area.

Define the mapping of document types

Mapping document types enables a destination connector to consume documents produced by a source connector. In the Scenario builder, a mapping box positioned between two connectors enables you to edit the mappings between a produced source document-type and a consumed destination document-type.

Creating a mapping involves associating elements of a source document-type to those of a destination document-type.

In order to simplify this chapter, we use the concepts **source element** and **destination element** as well as the source and destination of the field, structure and collection.

When no source document-type elements can be directly associated to a destination document-type element, you must write mapping scripts.

To learn about mapping scripts, refer to the chapter Define mapping scripts [page 117].

The mapping box

The mapping box enables you to edit the mappings between the document types produced by a source connector and those consumed by a destination connector.

Configuring the mapping box

Configuring a mapping box in Connect-It is performed with the help of a wizard.

To launch a mapping box's configuration wizard, you can use any of the following methods:

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

There are two pages in the mapping-box configuration wizard, which are documented in the following sections.

🖊 Note:

The second page only appears if you clicked 🖬 in the Scenario builder toolbar.

Naming and describing the mapping box

The first page of the configuration wizard enables you to specify:

- The name of the mapping box.
- The description of the mapping box.

Name

Name your connector. By default, the value of this field is *Mapping*. No two tools can share the same name in a single scenario; all names must be unique.

Description

Enter text to describe the connector.

Advanced configuration

This page enables you to:

- Specify the behavior of the *[..]* operator in the mapping scripts (**Verify the existence of the element under which the [..] operator applies**).

  When this option is:

- selected: for each document type node in square brackets [...], if this node is empty then the document is rejected.
- not selected (default option): the existence of a document type element is not verified.

  If you wish to verify the existence of a document type, you must replace the square brackets [...] with curly brackets {..}.

  In Connect-It version 3.2 and higher, the *{..}* operator can be used instead of the *[..]* operator. This syntax checks for the existence of the element for the document produced.

- Force the explicit declaration of variables in Basic scripts. All declared variables must be preceded by the **Dim** instruction.

  This option is selected by default.

Example of using the *[..]* and *{..}*

The scripts from the following table test the existence and the value of the *FamilyName* field. If this field has a value, the return value is *Doe* in all cases.

Depending on whether you select the **Check the existence of the element to which the [..] operator applies** or not and depending on the type of operator used in the script, the value sent in the consumed document is different.

📌 Important:

The notion of an empty value differs depending on the type of field concerned:
- For a text-type field, the empty value is equal to "".
- For a numeric-type field, the empty value is equal to 0.
- Etc.

Option unselected

	FamilyName = ""	FamilyName does not exist.	FamilyName = "*value*"

`RetVal = "Doe"`
`If [FamilyName] = "" Then`
`RetVal = [FamilyName]`
`End If`
	Empty value	Empty value	*Doe*

`RetVal = "Doe"`
`If {FamilyName} = "" Then`
`RetVal = [FamilyName]`
`End If`
	Empty value	Empty value	*Doe*

`RetVal = "Doe"`
`If {FamilyName} = "" T`
`hen`
`RetVal = {FamilyName}`
`End If`
	Empty value	Empty value	*Doe*

Option selected			

`RetVal = "Doe"`
`If [FamilyName] = "" T`
`hen`
`RetVal = [FamilyName]`
`End If`
	Empty value	The node concerned by the script is rejected by the Scenario builder. The message "" appears in the Document log.	*Doe*

`RetVal = "Doe"`
`If {FamilyName} = "" T`
`hen`
`RetVal = [FamilyName]`
`End If`
	Empty value	The node concerned by the script is rejected by the Scenario builder. The message "" appears in the Document log.	*Doe*

`RetVal = "Doe"`
`If {FamilyName} = "" T`
`hen`
`RetVal = {FamilyName}`
`End If`
	Empty value	NULL value	*Doe*

Editing mappings
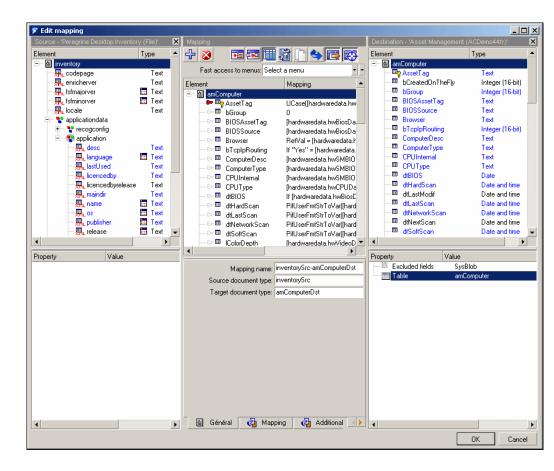
This section presents the different procedures enabling you to edit the mappings of your integration scenarios.

Mapping edit window

The mapping edit window is divided into three distinct zones:

- A zone showing the document types published by the selected source tool.

  If you selected an existing produced document-type in the **Select source and destination** dialog box, the unfolded root node of this document type is shown in this zone.

- A zone showing the document types published by the selected destination tool.

- A mapping zone containing:
  - A work area in which source and destination elements are mapped together.
  - A pane showing the directives tabs of the destination tool and a tab in which to describe each mapping node.

Figure 4.10. Mapping edit window
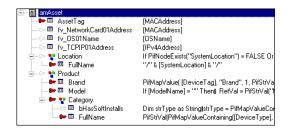
The mapping-box toolbar allows you to perform the most frequent mapping actions.

Icon	Function
	Places a destination element on the work area.
	Deletes the current selection from the work area.
	Associates a destination element with a source element.
	Filters the unused source and destination document-types in the mapping.
	Duplicates a destination element in the work area.
	Finds the mapped destination element.
	Displays or hides the document types published by the source connector.
	Displays or hides the document types published by the destination connector.

Icon	Function
▣	Enables you to display a find box in the zone where an element is selected.
▣	Enables you to display a find and replace box in the mapping's workspace.

The mapping window shows the detail of the mapping between destination and source elements. When a script is used, it appears in summarized form next to the destination element with which it is associated.

Figure 4.11. Summary of the mapping scripts in the work area

By exposing all the document types published by a source and destination connector, the edit window of a mapping enables you to simultaneously create the document type produced by the source connector and the document type consumed by the destination connector.

Note:

In the same mapping, it is not possible to map elements from more than one source document-type or to more than one destination document-type. If you change one of the document types while editing a mapping, you are effectively starting a new mapping, and you lose all prior mappings. To map a source element to more than one destination document-type, you must create several mappings using the same source document-type.

On the other hand, when you use different mappings, using the same source document enables you to minimize the number of requests to perform on the source connector.

You can also select a destination document and drag one of its nodes onto your work area without having to associate it with a source document: The fields located under this node will appear in the work area.

> ✎ Note:
>
> You can view the source data, and sometimes the destination data, by using the **View data** shortcut menu on a source or destination element.

To mask or display the source zone

1 Perform the following actions:
   - Click 📇.
   - Right-click in the mapping edit window and select or unselect **Show source**.

To mask or display the destination zone

1 Perform the following actions:
   - Click 📇.
   - Right-click in the mapping edit window and select or unselect **Show destination**.

To move a zone of the mapping edit window

1 Place the pointer on the upper border of the zone.
2 Click on this dark gray border and wait for a black rectangle in the shape of the zone appears.
3 Move the black rectangle where you want it and right-click.
4 Release the mouse button after you move the black rectangle to the desired location.

To create a mapping

1 Select a mapping box in a Scenario diagram.
2 Perform the following actions:
   - Select the **Mappings** tab.
   - Double-click.

     Double-clicking displays a dialog box containing the same information as found in the **Mappings** tab.
3 Click 🗋.

When you create a mapping, the **Select source and destination** dialog box is displayed in which you select the:

- Source tool
- Source document-type
- Destination tool

Figure 4.12. Updating the collection in AssetCenter

> **Note:**
>
> This dialog box is not displayed when:
> - The mapping box already has a single source tool and a single destination tool.
> - No produced document-type has been created for the source tool.

To edit an existing mapping

1 Select a mapping box in a Scenario diagram.
2 Perform the following actions:
   - Select the **Mappings** tab.
   - Double-click.

     Double-clicking displays a dialog box that contains the same information as found in the **Mappings** tab.
3 Perform the following actions:
   - Click 🔍.
   - Double-click the mapping that you want to edit.

To deactivate a mapping

Deactivating a mapping prohibits the connector from producing the documents necessary for the mapping. For example: With a mapping, you can delete the records in a destination application corresponding to assets no longer existing in the source application. If you only want this deletion to occur once, you need to deactivate the mapping after it has finished.

To deactivate a mapping, clear the box corresponding to this mapping in the **Mappings** tab.

To delete a mapping

1 Select a mapping box in a Scenario diagram.
2 Perform the following actions:
   ▪ Select the **Mappings** tab.
   ▪ Double-click.

      Double-clicking displays a dialog box that contains the same information as found in the **Mappings** tab.

3 Click 🔖.

Identification of complex elements and automatic mapping

When you create a mapping between two document types, the application identifies similar fields, i.e. those with the same name. These fields are mapped automatically.

For example, when you map the **location** document type of the ServiceCenter connector to the **amLocation** document type of the Asset Management connector, the **Address** fields are automatically mapped.

Path of document-type elements

To locate a document-type element, in particular in the **Source for the collection mapping** field, you must indicate the path of this element in the document type. Parent-child relations are shown by a period. The following schema represents a set of linked elements.

To indicate the path of collection X, you must write:

`Structure_A.Structure_B.Collection_X`

This syntax means that collection X is a child of structure B, which in turn is a child of parent structure A.

You can also use the following syntax:

```
'Structure A'.'Structure B'.'Collection X'
```

Example :

In the **inventorySrc-amComputerDst** mapping of the Desktop Discovery to AssetCenter scenario, the **Document** collection of the **amComputer** destination document type is mapped to the **storedfile** collection. Since this collection is child of the **storedfiles** structure of the **inventory** document type, the **Source collection** field indicates the following path:

```
storedfiles.storedfile.name
```

If an element name includes a period, square brackets or parentheses '.[]()', you must place the element in single quotes:

```
'Element.A'
```

Example :

Many elements of document types published by the ServiceCenter connector include periods. To reference the **sw.vendor** field of the **sw.vendor** collection in the **ICMDevice** document-type published by the ServiceCenter connector, you must indicate the following path:

```
'sw.vendor'.'sw.vendor'
```

For fields that do not include periods in their names, single quotes are optional. You can write both:

```
ElementA or 'ElementA'
```

Path of document elements

The notation used for the path of document elements is the same as is used for document types. The only exception concerns members of a collection.

Members of a collection are numbered incrementally starting from 0: The first member is marked 0, the second member 1, etc.

To indicate a specific member of a collection in a path, refer to its rank using the following syntax:

Name of the collection [rank of the member]

Example: The path concerning the **VersionName** field of the first member of the **Software** collection is indicated as follows:

`Software(0).VersionName`. For the second member of the collection, the path is `Software(1).VersionName`, etc.

You can also use a variable to designate a given element:

```
Dim lId as Long
lId = 3
[Software(lId).VersionName]
```

This syntax is particularly useful to browse the members of a collection. The following example tests for a member (in this example the software Connect-It) in the **Software** collection:

```
' Get number of members in the "Sofware" collection
Dim lCount As Long
Dim lCur   As Long
lCount = PifGetItemCount("Software")
lCur   = 0

Dim bFound As Integer
bFound = 0

While lCur < lCount AND iFound = 0
If [Software(i).Name] = "Connect-It" Then
bFound = 1
End If

lCur = lCur + 1
Wend
```

Path of a collection of attributes

To specify a given member of a collection of attributes (collection with only one field), don't forget to specify the name of this attribute in your path. Example: To specify the members of an *Address* collection with an *address* attribute, you must write: Address(1).Address, Address(2).Address, Address(3).Address, etc.

✎ Note:

When you perform the collection-to-collection mapping (**Source for the collection mapping** field is populated), it is no longer necessary to specify the numbers of the collection members.

Example:

```
[Software.Version]
```

Executing mapping scripts on opening and closing a session

Mapping scripts can be supplemented by additional scripts, executed on opening and closing a session.

These additional scripts are most often global variables, called by the *OpenSession()* and *CloseSession()* procedures.

These procedures must use the following syntax:

```
Sub OpenSession()
rem add your code
End Sub

Sub CloseSession()
```

```
rem add your code
End Sub
```

All scripts declared after the *OpenSession()* procedure are executed when the session is opened, when the first document is produced.

All scripts declared after the *CloseSession()* procedure are executed when the session is closed, after the last document is consumed.

For further information on additional scripts and global variables, refer to the chapter Define mapping scripts [page 117], section Additional scripts [page ?].

To order the mappings

The mappings that you edit are ordered by document-type produced by the source connector. When you have several mappings for the same document type, you can classify them to indicate the order in which they will be executed during the document-production phase. In certain cases, the first mapping enables you to create records in the destination application while the following mappings enable you to update the records.

1 Select a mapping.
2 Click ▲ or ▼ to move up or down in the list of mappings.

Order of document production and mapping execution

The following rules apply to the:

- Execute the scenario without writing to the connectors (test mode)

  The order in which the documents produced by the source connector, as shown in the **Produced document types** tab, applies.

- Scheduled mode

  The order that is defined in the Scheduling edit window is applied. In this window, a scheduler can be associated with several document types: The order in which documents are associated with a scheduler is the order to which we are referring here. In this case, the order defined in the **Produced document types** tab has no impact.

To describe a mapping

The Scenario builder enables you to describe each node of your mapping.

Example: a description of a mapping script used to test for the presence of an element in a structure or collection.

1 Edit a mapping.
2 Select a node in your mapping.
3 Select the **Mapping** tab in the workspace.
4 Enter your description in the **Description** text zone.
5 Save your mapping.

The description is saved in the `.scn` file.

You can copy a mapping - or part of a mapping - of a source document-type to a destination document-type. Then, you can add or merge this copied source-to-destination mapping into another mapping that shares the same destination document-type as the first mapping. (For example: In a scenario using the Asset Management connector, you can copy a part of the mapping where the destination document-type is **amAsset** and then you can paste it into another mapping whose destination document type is also **amAsset**.)

This function is particularly useful if you need to create a large number of identical, or very similar, mappings.

1 Open a mapping in an existing scenario.
2 In the mapping work area, place your pointer on one of the following destination document-type elements:
   ▪ Root node
   ▪ Structure
   ▪ Collection
   ▪ Field
3 Right-click.
4 Select **Copy mapping** in the shortcut menu that appears.
5 Open another mapping where the copied mapping's destination document-type is present (either as a root node, a structure or collection).
6 In the work area, place your cursor in the spot where you want to add or merge your mapping (a root node, structure or collection).
7 Right-click and select **Paste mapping** in the shortcut menu that appears.
8 Depending on your needs:
   ▪ Click **Yes** to merge the mapping: The selected element is modified by the copied mapping (the two mappings are merged into one mapping).
   ▪ Click **No** to add the mapping: The copied mapping is added to the existing mapping in the form of a new structure or new collection.

In the mapping edit window, you can preview the data of the source and destination applications. Note: Previewing the destination data is only possible for certain connectors.

1 Edit a mapping.
2 Position your pointer in the source or destination document-type window.
3 Right-click.
4 Select **View data** from the shortcut menu.

Wait for the data viewing window to appear.

Edit the data preview

You can edit the source document-type of your mapping in the edit window in order to preview the data of your choice.

1   Edit a mapping.
2   Position your pointer in the source document-type window.
3   Select **Edit a document type to preview data**.
4   Wait for the source document-type edit window to appear.
5   Add or delete elements to the workspace.
6   Click ✎.
7   View your source data.
8   Click **Close** to go back to the mapping edit window.

Important: Do not use this function to edit the source document-type.

All modifications made to the source document-type are lost when you close the preview window.

To find an element in the mapping edit window

1   Select an element in the source or destination document-types zone or in the workspace.
2   Perform one of the following actions:
    - Right-click and select **Find** (Ctrl + F) from the shortcut menu.
    - Click ▦.
3   Enter the name of the element that you want to find in the **Find what** field and select the options of your choice:
    - Case sensitive
    - Select the direction of your search: up or down.
4   Click **Find**.

To search through previous elements or following elements:

1   Perform one of the following actions:
    - Right-click and select **Find next** (**F3**) from the shortcut menu.
    - Right-click and select **Find prev.** (**Shift + F3**) from the shortcut menu.

Find - Rules

Finding an element in the hierarchy of document types is governed by the following rules:

- The search is limited to the elements displayed in the window of published document-types. If you want your search to apply to the sub-nodes of a given node, you must unfold the node before the search.
- A search in the workspace will include destination document-types as well as mapping scripts.

To find and replace an element in the mapping scripts

1 Select an element in the workspace.
2 Perform one of the following actions:
- Right-click and select **Replace** (Ctrl + H) from the shortcut menu.
- Click ▦.
3 Enter the name of the element that you want to find in the **Find what** field and select the options of your choice:
4 Enter the name of the replacement for this element in the **Replace with** field.
5 Select the options of your choice:
- Case sensitive
- Select the direction of your search: up or down.
6 Click **Find**, **Replace** or **Replace all**.

Mapping types

This section details the Scenario builder's different mapping types.

Direct field-to-field mapping

A direct field-to-field mapping is simply a direct mapping of a source field to a destination field. When data is processed, the value of the source field is used to populate the destination field in the document produced by the mapping box. This operation is performed without the involvement of the Basic engine.

Example :
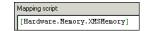
The **Name** field in a source document-type is mapped to the **Name** field in a destination document-type.

Different methods can be used to perform a direct field-to-field mapping:
1 Drag-and-drop method

- Select a field in the **Source** or **Destination** window.
- Drag it onto a field in the other window.

2 Icon method
- Select the field in the source zone and the field in the destination zone that you want to map together.
- Click ▦.

3 Two-part drag-and-drop method
- Select a destination field.
- Drag and drop it onto the workspace.
- Select a source field.
- Drag it onto the work area at the same line as the destination field.

4 Drag-and-drop with mapping script method
- Select a destination field.
- Drag and drop it onto the workspace.
- Enter the name of the source field directly in the **Mapping script** field.

  Each direct field-to-field mapping appears automatically in the **Mapping script** field. If you enter the name of the source element manually, you must place it in square brackets.

Figure 4.13. Source element in the mapping script

```
Mapping script:
[Hardware.Memory.XMSMemory]
```

Collection-to-collection mapping

A collection is made up of an indefinite number of elements (fields, structures, collections).

Each element of a collection is a member of the collection.

Example: For a given document type, the software that is installed on a computer is represented by a collection. Each member of this collection contains the description of a software application (name, publisher, version...).

Figure 4.14. Collection

When you map a source collection to a destination collection, Connect-It calculates the number of members in the source collection when processing the data, and it creates the corresponding number of members in the destination collection.

Example :

In the Desktop Discovery to AssetCenter scenario, elements in the **application** (software) collection of the **inventory** document type are associated with elements in the **Softinstall** (installed software) collection of the **amAsset** document type.

Different methods can be used to perform a collection-to-collection mapping:

- Drag-and-drop method
- Keyboard method

Drag-and-drop method

1 Drag a destination collection onto your work area.
2 Select a source collection. Drag it onto the work area at the same line as the destination collection while holding down both the left mouse button and the **Ctrl** key on your keyboard.
3 Release the mouse button when the triple document symbol is displayed underneath the mouse pointer.

The **Source for the collection mapping** field is automatically populated with the path of this collection.

📝 Note:

A warning message appears each time that a mapping of a collection element is performed. This is to remind you that, in most cases, you need to populate the **Source for the collection mapping** field. The **Display a warning every time a collection-to-collection mapping is needed** option (**Confirmations** section), accessible via the **Edit/Options** menu, enables you to disable this warning message.

Keyboard method
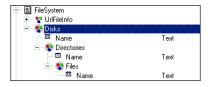
1 Drag a destination collection onto your work area.
2 Enter the path of the source collection directly in the **Source for the collection mapping** field of the mapping script. Since you are not calling on the value of this collection, you do not need to include square brackets.

Mapping sub-collections
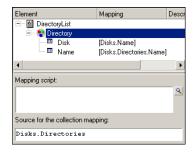
A produced document type may contain collections that in turn contain other collections.

If you wish to map sub-collection elements from the produced document type to a first-level collection in the destination document type, you can specify the path of the sub-collection of the produced document type as the source.

For example, the produced document type **FileSystem**, contains the collection **Disks**, which in turn contains the collection **Directories**, which contains the collection **Files**.

We wish to create a mapping for the following consumed document type, in order to recover the list of all the folders:

The mapping used for the **Directory** collection is:

```
Disks.Directories
```

All the collections contained in the parent element (a collection) create the same number as first-level collections (without parent documents) in the destination.
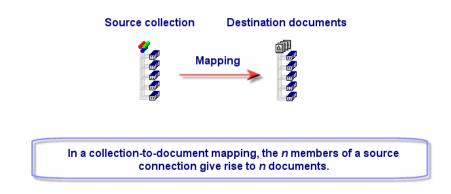
Collection-to-document mapping

With Connect-It, you can also associate a source collection with the root node of a destination document-type (▤). While processing data, Connect-It creates as many destination documents as there are members in the source collection.

A collection-to-document mapping is defined in the same way as a collection-to-collection mapping. The root node of the document type replaces the destination collection.

Figure 4.15. Collection-to-document mapping

Example :

In the `scac\sc4ac41\scac.scn` scenario, the **Software** collection (which describes the software installed on a computer), is mapped to the **pcsoftware** document-type. When the data is processed, the n members of the software collection give rise to n **pcsoftware** documents, which in turn, send n input events to ServiceCenter.

Syntax

The mapping script between a collection and document type must adhere to the following syntax:

```
[Collection.Element(valeur)]
```

The value given in parentheses is that of the element of the collection.

For example:

```
[Machine.Name(0)]
```

Field-to-collection mapping

In a field-to-collection mapping, one or more source fields that do not belong to a collection are mapped to a destination field belonging to a collection. The structure containing this field is dealt with as a collection with one single member.

Example :

The **InstalledCards** field regroups all the cards installed on a computer: motherboard, graphics adapter, sound card. In the source document-type, this information is represented by separate fields (**Motherboard**, **GraphicsCard**, **SoundCard** fields).
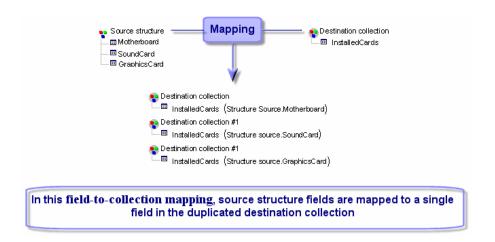
After dragging your collection onto the work area, duplicate it as many times as is necessary and map the source fields to the **InstalledCards** fields of these duplicates.

To duplicate a collection in the work area:
1 Select this collection.
2 Click 📄
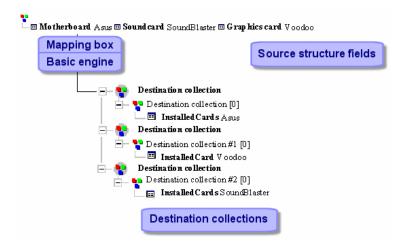
The first duplication of this collection is indicated by the suffix #1, the second duplication by #2, etc.

Figure 4.16. Field-to-collection mapping

In this **field-to-collection mapping**, source structure fields are mapped to a single field in the duplicated destination collection

When processing the data, the mapping box produces as many destination collections with single members as there are duplicates of this destination collection. Each member has the value of one of the source fields.

Blob-type field mapping

Blob-type fields (binary-large-object-type fields) are those fields corresponding to binary data: sound files, video files, image files.

A Blob field can only be mapped to another Blob type field.

Two cases are possible:

1 You map the destination Blob-field to the source Blob-field.

   In this case, the binary files are transmitted from the source application to the destination application via the scenario connectors.

2 You map the destination Blob-field to a field indicating the name of a binary field located in a file, which is shared by Connect-It and the destination application.

   In this case, you must use the **Load Blob from file** option, available in the mapping script editing zone.

Create a document or a collection item even if the source collection does not contain items

If you select this option, a collection that does not contain any members produced by the source connector enables you to create:

- A document consumed by the destination connector if you perform a collection-to-document mapping.

  Consult this section's sub-section: Collection-to-document mapping [page 109].

- A collection member in the document consumed by the destination connector if you perform a collection-to-collection mapping.

  Consult this section's sub-section: Collection-to-collection mapping [page 106].

Forcing the creation of a member of a collection or a document enables you to signal the absence of this value in the destination application.

Load Blob from file

This option enables you to indicate that:

- The Blob file is located in the folder shared by Connect-It and the destination application.

  Example: [name of the shared folder on the network]/blob/pictures.

- The destination application loads the Blob file from this shared folder.

In order for the destination application to load the file from the shared folder:

1 In the mapping, associate the name of the *Blob* file from the source to a Blob-type file from the destination application.

   The file name of the source application can either be the field name linked to the Blob file in the source application (icon.name), or it can be the full path of the file in the shared folder: [name of shared folder on network]/blob/pictures/monimage.png.

2 Select the **Load Blob from file** option.

   While the scenario is running, the destination application recovers the **Blob** file from the shared folder.

Convert a hexadecimal string to a Blob

This option enables you to convert a hexadecimal string to a Blob.

When this option is enabled, you must specify the code page to use.

Exporting a Blob to a file

You may wish to export the data contained in a Blob field in order to verify the contents.

To export the contents of a Blob field:

1 Preview the data produced.
2 Right-click the Blob field and then select **Copy this Blob to a file** from the shortcut menu.
3 Enter the name of the file to save and its nature.

Mapping options

Depending on which connectors are used, different tabs in the mapping box will be available or not.

The following tabs are common to all connectors:

1 General
2 Mapping
3 Blob

   This tab is available for all connectors handling Blob files. For further information on advanced reconciliation, refer to chapter Define mapping scripts [page 117], section Additional scripts [page 150].

4 Additional scripts

   This tab is available for all connectors. For further information on advanced reconciliation, refer to chapter Define mapping scripts [page 117], section Additional scripts [page 150].

The following tabs are specific tabs:

- Reconciliation

   This tab is available for Database-type connectors. For further information on advanced reconciliation, refer to the *Connector Guide*, chapter *Connector directives*, section *Reconciliation*.

- Advanced reconciliation

   This tab is specific to the AssetManagement connector. For further information on advanced reconciliation, refer to the *Connector Guide*, chapter *Connector directives*, section *Reconciliation*.

- Reconciliation script

This tab is specific to the AssetManagement connector. For further information on advanced reconciliation, refer to the *Connector Guide*, chapter *Connector directives*, section *Reconciliation*.

Reconciliation keys

You can access the reconciliation tools in the **Reconciliation** tab of the detail of a mapping.
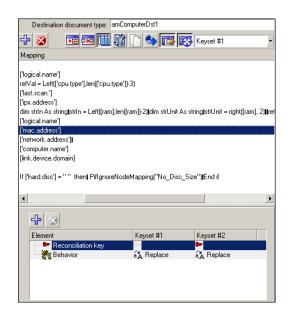
Performing a reconciliation operation means bringing together old and new data.

Reconciliation consists of defining the fields that enable the unique identification of the records in the tables that a connector creates or updates.

The fields used to identify the data to be reconciled are designated as reconciliation keys.

Complex mapping elements may have multiple reconciliation keys. Each key belongs to a key set.

For example, for a given reconciliation, *AssetTag* belongs to key set #1 and *PhysicalAddress* belongs to key set #2.

For each reconciliation and key set, we choose one of the following behaviors:

- Update or insert
- Update only

- Insert only
- Delete

Also refer to the *Connectors* Guide, chapter *Connector directives*.

Viewing reconciliation keys

A filter is available in order to distinguish between the two sets of keys.

To filter on key set # 1 or key set # 2, select the number of the key set in displayed in the icon bar of the mapping box.

How key set weighting works

Each key set has a weight allocated to it, 100 by default, which sets its priority in a given reconciliation operation.

The way in which a key sets function depends on their defined behavior and their weight.

Functioning of keys for update or insert

In the case where, all reconciliations are **Update or insert**, there are three behaviors for the first key set:

1 If the first key set does not reconcile a record, then the next key set is used.
2 If the first key set reconciles a record:
   - and the weight of the key set is greater than or equal to 100, then this record is updated
   - otherwise, the record and its weight are stored in the list of potential reconciliation candidates and the following key set is examined.
3 If the first key set reconciles several records, then the records and their weights are stored in the list of potential reconciliation candidates and the next key set is examined.

Behavior of the following key set if none of the previous key sets were able to identify a single record:

1 If the following key set does not reconcile a record, then the next key set is used.
2 If the key set identifies one or more records, then, for each new record:
   - A check is performed to test whether it already has been reconciled by one or more of the previous key sets: A record in the list of candidates is considered to be the same as the new record to be processed if the values of the fields are identical (except in the case of the Asset Management connector, for which we check whether the fields have the same main ID).
   - If the record is already in the list of candidates, the weight of the key set is added to the previous value.

- If the records is not in the list of candidates, it is added with the weight of the key set.
- If for all candidates, there is only one record with a weight greater than or equal to 100, then this record is updated.
- If multiple records have weights greater than or equal to 100, an error message indicates that the key sets do not identify a single record.
- If the weights of all the records are less than 100, then the following key set is used.

Once all the key sets have been examined, if no record has been selected by the process and there is only one candidate left in the list (with its weight less than 100) then this record will be updated (even though its weight is less than 100).

Example :

Three key sets are used, their weight being 40, 70 and 50 respectively.

The first key set reconciles the records A, B and C. Their weights are therefore the following:

A = 40

B = 40

C = 40

The second key set reconciles the records C, D, E and F. The weights of the records become:

A = 40

B = 40

C = 110

D = 70

E = 70

The record C is the only one with a weight greater than or equal to 100, it is updated and without key set #3 being used.

Example :

Three key sets are used, their weight being 40, 70 and 50 respectively.

The first key set reconciles the records A, B and C. Their weights are therefore the following:

A = 40

B = 40

C = 40

The second key set reconciles the records B, C and D. The weights of the records become:

A = 40

B = 110

C = 110

D = 70

Records B and C having the save weight greater than or equal to 100, it is not possible to uniquely identify them and an error message is generated.

Define mapping scripts

When a direct field-to-field mapping is not possible, a mapping script is needed (**Mapping script** field). This script indicates how data or source elements are manipulated so that they give a value to the destination element.

In the mappings of an integration scenario, scripts enable you to:

- Associate a fixed value to a destination element.
- Associate a calculated value to a destination document-type field.
- Subordinate the processing of a structure's or a collection's fields to the verification of a condition.

Example :

An example of this is a Basic script that concatenates two source elements: The return value of this concatenation is mapped to a destination element when data is processed.

Editing associated files

The mapping scripts sometimes necessitate using associated files.

These associated files contain:

- Character string tables

- Maptables
- Global variables and functions

These files are saved independently of the `.scn` file. Their file names correspond to the contents.

File extensions	Description
`.str` (for string)	Character string tables
`.mpt` (for maptables)	Maptables
`.bas` (for Basic)	Global functions and variables
`.scp` (for Script Constant)	Script constants
`.usr` (for User)	User formats

Note:

A scenario (an `.scn` file) that does not have its associated `.str`, `.mpt` and `.bas` files will not function. Make sure that, each time you move a scenario, its attached files are also moved with it. It is also useful keep these files right next to your `.scn` file so that you can easily assure their presence. Moving the associated files prohibits the scenario from obtaining their contents during data processing.

Editing an associated file

You can edit elements using the Connect-It editors as an intermediate tool.

Use the **Scenario** menu to access these editors.

When you use the **Scenario** menu to access an editor, a window is displayed. This window has two panes:

- A pane showing the files (`.str`, `.scp`, `.mpt`, `.usr` or `.bas`) associated with the current scenario.
- A file editor pane (character string tables, maptables, global functions and variables).

Figure 4.17. The maptable editor

These three editors use the same text editor that is used for mapping scripts.

To create a new associated file
1 Click ☐ on the toolbar.
2 Enter the name of your file in the **File name** field.

   By clicking ☐, you can browse your hard drive to find the folder where you want to create this file.
3 Click **Create**.

> ✎ Note:
>
> If the file already exists, it will be opened instead.

To open an existing file
To open a file appearing in the pane showing the associated files of the current scenario, click this file.

To delete an associated file
To delete a file associated with the current scenario:
1 Select this file in the window of the associated files.
2 Click ☐ on the toolbar.

Validating scripts

Connect-It constantly verifies whether your scripts are coherent or not. This validation checks, not only the Basic syntax, but also makes sure that the source document contains those elements that are referenced in the mapping script. (Only elements that are visible in the source document structure are considered in the validation.)

🖊 Note:

The **Do not authorize the validation of a script that includes compilation errors** option (**Confirmations** section) enables you to activate or deactivate the behavior of this validation.

Editing the text of an associated file

You can access the text editor by clicking the 🔲 icon, available in each editor near the **Mapping script** field in the mapping edit window.

This text editor facilitates the editing of Basic scripts and the .str, .mpt, and .bas files. The following class of text-editor functions are available: cut/paste, undo, etc. These functions are available via the **Edit** menu or the toolbar.
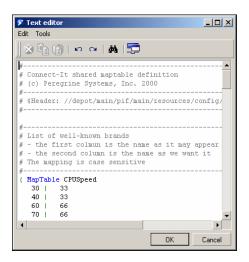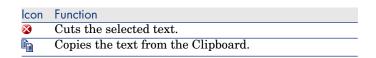
Table 4.1. Text editor - Toolbar

Icon	Function
⊗	Cuts the selected text.
🗋	Copies the text from the Clipboard.

Icon	Function
📋	Pastes the text from the Clipboard.
↶	Cancels the current action.
↷	Redoes the canceled action.
🗗	Displays the configuration window of the text editor.

Configuring the text editor

To configure the text editor:

- Click 🗗

  or

- Select **Options** from the **Tools** menu.

Configuring the text editor allows you to modify the templates on which your files are based.

Warning:

In order to edit one of the templates available in the text editor, verify that the `codeedit.cfg` file located in the `bin` sub-folder of the Connect-It installation folder is not read only.
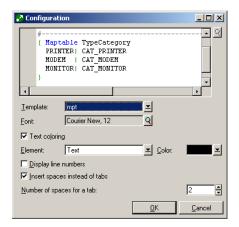
It thus goes without saying that for each text editor there is a template. For example: If you are editing a maptable, it is the `mpt` template that is used by the text editor.

Use the template ...	To edit ...
bas	mapping scripts
	global functions
str	character string tables
mpt	Maptables
sql	SQL queries
scn	Connect-It scenarios
Default	all other file types

For a given template, you can:

- Modify its character font (type, weight, style).
- Select the color of its elements.
- Show or hide its line numbers.
- Replace tabulations with one to eight spaces.

Figure 4.18. Configuring the text editor

Modifying a template

To modify the font used in a template, click the magnifier next to the **Font** field. A dialog box appears in which you can select a font from those fonts installed on the computer.

You can also modify the color of certain elements:

- Text
- Background
- Selected text
- Selected text background
- Line numbers
- Numeric values
- Delimiters
- Comments
- Strings
- Keywords
- Operators

To set a color for an element, select it in the **Element** field and then select a color in the **Color** field.

You can also choose not to color the text by clearing the **Text coloring** option.

To display line numbers and insert spaces instead of tabs, select the appropriate check boxes.

To access the character-string editor, use the **Edit** menu.

In a character string table, each string is associated with a unique identifier. When the data is processed, the Basic scripts replace the identifiers with their respective character string.

To use a character string in your mapping, you need to use the **PifStrVal** function followed by the identifier of the string in quotes and parentheses.

The following example shows how part of the Category.str character string table is used in Desktop Discovery to Asset Management scenario.

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

Example :

Thus, to obtain the "/Hardware/Unix Workstation", use the following line of code in your mapping script: PifStrVal("CAT_UNIX")

In this table, the CAT_UNIX identifier corresponds to the "Hardware/Unix Workstation" value. If you change this character string, the scenario accounts for this change without having to recompile the Basic script that uses this identifier. You can also use several character string tables corresponding to a given language: FrCategory.str for French, DeCategory.str for German, etc.

Concatenating character strings

The & operator enables you to concatenate character strings.

If one of the operands is not a character string type of operand, a character-string conversion is automatically performed.

Example

```
RetVal = "Current date: " & Date()
```

This script displays a date according to the following format:

```
Current Date: 2002/12/26
```

Creating a character string from a character code

The **Chr()** enables you to generate a character according to the code page used by Connect-It.

This function is useful for generating characters such as the following types:

- Single quotation marks: Chr(39)
- Double quotation marks: Chr(34)
- Carriage return: Chr(13)
- Line skip: Chr(10)

Example

```
Dim lVal As Long
lVal = 5
RetVal = "Value: " & Chr(39) & lVal & Chr(39)
```

This script returns the value:

```
Value: '5'
```

UNIX/Windows character string

The line return is not managed the same way in UNIX as it is in Windows.

In Windows, the line return is managed by the characters Chr(13) and Chr(10). In UNIX, only Chr(10) is used.

Example

```
RetVal = Replace([WindowsText], Chr(13) & Chr(10), Chr(10))
```

This script replaces all the Windows line returns by UNIX ones in a character string recovered in the *WindowsText* of the source document.

Maptables

To access the maptable editor, use the **Scenario** menu.

In a maptable, the first column contains a key and the following columns contain values. Each key corresponds to a value in a given column.

To obtain the value of a given column in your mapping, you associate the **PifMapValue()** function with the key as a parameter, a default value if the key is not defined in the maptable, the name of the maptable and the column number.

At runtime, each time that a key is found in a script, it is replaced by the value of the column indicated in your script.

For example: In this extract of the **Brand** maptable, the **Apple** and **Macintosh** keys return "Apple" if column 1 is specified in the script.

```
{ MapTable Brand
Compaq | Compaq
IBM -Lexmark| IBM - Lexmark
Hewlett Packard| Hewlett Packard
```

```
HP-UX| Hewlett Packard
Toshiba | Toshiba
Apple | Apple
Macintosh | Apple}
```

Example :

Thus, to obtain "HP-UX", use the following line of code in your mapping script:

```
PifMapValue([Name of field in the
mapping],"Brand",0,"DefaultValue")
```

For a complete description of the functions linked to the maptables, refer to the following functions in the Programmer's Reference:

- PifCreateDynaMaptableFromFmtName
- PifIsInMap
- PifMapValue
- PifMapValueEx
- PifMapValueContaining
- PifMapValueContainingEx

Creating maptables for different languages

To obtain maptables that return strings in different languages:

1 Create character string files in which each line maps an identifier to a localized string (French, Polish, etc.) as [Identifier], ["localized string"]. Example: a file category.str, for English, contains CAT_UNIX, "UNIX Workstation"), another file fcategory.str, for French, contains CAT_UNIX, "Station de travail UNIX"), etc.

2 Include this file in your maptables file using the following syntax: #include_str "[File name]". Example: #include_str "category.str"

3 Reference each string in the maptable using an identifier, the syntax of which is [dollar]([Identifier]).

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

Global functions and variables

To access the global functions and variables editor, use the **Scenario** menu.

Your mapping scripts use functions and variables that you may want to use in several different places. These global functions and variables can be saved in the same .bas file and can then be accessed by the mapping script. At runtime,

the mapping box uses these functions and variables located in the `.bas` files associated with the scenario.

For example: The global function **CPUSpeed()** is contained in the `gen.bas` file. This function references a maptable, **CPUSpeed()**, and makes it possible to specify the generic value of the CPU speed.

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

Global variables

Variables declared in a mapping script are local.

For example, if you declare a variable in a script associated with a field, you cannot use this variable for another element in the document.

If the same variable name is used in a mapping script performed on another element, a new local variable is created, and the value of the variable in the first script cannot be recovered.

Declaring a global variable

For the variable to be global, it must be declared at the root of a document type, in the **Additional scripts** tab, in order for it to be calculated and used in the different scripts applied to the different complex elements in the same mapping.

Note:

In order to differentiate between global and local variables, we recommend prefixing global variables by *g_*.

For example:

```
Dim g_lCounter As Long
```

You must declare a global variable before defining the Basic script. The variable is taken into account when you open (**OpenSession()**) or close (**CloseSession()**) a session.

For example:

```
Dim g_lCounter As Long
(...)
```

```
Sub OpenSession()
rem add your code here
End Sub

Sub CloseSession()
rem add your code here
End Sub
```

Creating a document counter

To create a document counter, you must first declare a global variable as described in the section Global variables [page 126].

By default, this variable's value is 0.

Example 4.1. Counter

```
g_lCounter = g_lCounter + 1
PifLogInfoMsg(g_lCounter)
```

Each time that a document is produced by a source connector, the counter increases by 1 and the script returns the value of the counter to the Document log.

As long as the scenario is being executed, the counter is not reinitialized.

To reinitialize the value of the counter each time a session is opened, you must perform an additional script.

```
Dim g_lCounter As Long

Sub OpenSession()
g_lCounter = 0
End Sub
```

By keeping the counter's incrementing script at the root of the document type, the values of the global variables will be the following (for example):

```
Session 1
1
2
3
...

Session 2
1
2
3
...
```

Saving a global variable in a file

If a scenario is stopped while it is running in service mode, the current value of the global variable is lost.

To keep the current value of the global variable, you must save it in a file and reload it from this file whenever necessary.

Example 4.2. Global variable

In this example:

- A counter counts the number of documents processed without ever being reinitialized.
- This counter will be reloaded from a file if the application is stopped.
- On a root element, if the global variable equals 0, the additional script loads the "C:/tmp/counter.txt" file and verifies it if exits and if it contains data.

  If the global variable equals 0, it implies that:

  - The scenario is executed for the first time.
  - The scenario is interrupted.

```
Dim g_lCounter As Long

Sub OpenSession()
' Counter equals to 0, means that it has not been initialized.
' The application has been stopped and we reload the counter
' value
If g_lCounter = 0 Then

' If the file does not exist, it must be the first run. Check the
' existence of the file to avoid error on opening.
If FileExists("c:/tmp/counter.txt") Then

' Open the file in read mode
Open "c:/tmp/counter.txt" For Input As #1

' Check the file contains data and read the first line of the file.
If Not Eof(1) Then
Line Input #1, g_lCounter
End If
Close #1
End If
End If
End Sub
```

The following script enables you to save the value of the counter in a file each time that the counter is increased on the root element.

```
g_lCounter = g_lCounter + 1
Open "c:/tmp/counter.txt" For Output As #1
Print #1, g_lCounter
Close #1

PifLogInfoMsg(g_lCounter)
```

This script can be applied to any element of a document type.

Constants

You can define constants for your mapping scripts.

These constants are declared and saved in a file that can be accessed via the **Scenario/ Script constants** menu. This file has the extension `.scp.`.

Declare a constant as follows:

```
constant = <value of the constant>
```

A constant is replaced with its value during preprocessing of a Basic script (similar to the declaration of a #define macro in C or C++).

Declaration of the constant depends on its type (integer, string, etc.).

For example:

- For example, double quotes ("") must be used for a string.

```
login = "Maggie Smith"
```

- Numeric constants must be declared as follows:

```
idmax = 5
```

A constant is identified by the declaration **PifConstant(name of the constant)**.

Note:

Constants are replaced when the Basic script is processed. **PifConstant** is not considered as a Basic function but as a simple marker.

For example:

```
RetVal = PifConstant(login)
```

This script will be translated as follows when the script is executed:

```
RetVal = "Maggie Smith"
```

It is assumed that the **Script constant editor** was populated as follows:

```
login = "Maggie Smith"
```

Programmer's Reference

To help you write your scripts, an online Programmer's Reference is available for Connect-It.

This Programmer's Reference is available by pressing **F1** on your keyboard when your pointer is inside the:

- **Mapping script** field of the mapping edit window.
- Entry zone of the different editors.

Execute scripts when a session opens/closes

To execute scripts when a session opens, you must declare the code to be executed in the **Additional scripts** tab using the function:

```
Sub OpenSession()
rem add your code here
End Sub
```

To execute scripts when a session closes, you must declare the code to be executed in the **Additional scripts** tab using the function:

```
Sub CloseSession()
rem add your code here
End Sub
```

User formats

User formats are date-type or numeric-type formats that the user can define for use with the mapping scripts. These formats can be used exclusively with the **PifUserFmtVarToStr** and **PifUserFmtStrToVar** functions. To use these functions, refer to the Programmer's Reference guide available online by pressing F1 whenever you write a script in Connect-It.

Creating a date-type user format

To create a date-type user format:
1 Select **User formats** in the **Scenario** menu.
2 Click **Next** on the wizard page that appears.

   This page shows the date-type formats that cannot be defined by the user.
3 Click 🔁 on the second page of the wizard.
4 Click in the **Name** column under the name of the last format in the list of existing formats.
5 Name your format in the editable zone that appears.

   This name must not have any spaces.
6 Click in the **Format** column corresponding to the name that you just entered.
7 Enter a formula using the attributes from the **Date-type format symbol** pane.
8 Click ✓.

   An example of the created format appears in the **Result of date-type format** field.

Creating a numeric-type user format

To create a numeric-type user format:

1 Select the **Scenario/User formats** menu.

2 Click **Next** on the wizard page that appears.

   This page shows the numeric-type formats that cannot be defined by the user.

3 Click ⬑ on the second page of the wizard.

4 Click in the **Name** column under the name of the last format in the list of existing formats.

5 Name your format in the editable zone that appears.

   This name must not have any spaces.

6 Click in the **Format** column corresponding to the name that you just entered.

7 Enter a formula using the attributes from the **Numeric-type format symbol** pane.

8 Click ✓.

   An example of the created format appears in the **Result in numeric format** field.

Syntax to use for format creation

You must respect the following rules when you write format formulas:

- The symbols used for date-type or numeric-type formats is identical to those used in Windows operating systems.
- String files appearing in the formats must be surrounded by single quotes.
- The space between two values must be entered as if it were a string, surrounded by single quotes.

Table 4.2. Examples of date-type formats

Format formula	Example
yyyy'-'mm'-'dd	2002-02-07
hh':'nn':'ss	11:55:29
h':'nn':'ss	11:55:41
hh':'nn	14:18
hh" 'h' "nn	15:54:53
h t' 'nn' 'tt	2 29 pm
dd'/'mm'/'yy	07/02/02
dd'.'mm'.'yy	07.02.02
dd'-'mm'-'yy	07-02-02
dd'/'mm'/'yyyy	07/02/2002
dddd' 'd' 'mmmm' 'yyyy	Thursday, February 7, 2002

Table 4.3. Examples of numeric-type formats

Format formula	Example
n' 'n',''dd-'USD'	1 0 2 0 3,41-USD
-'USD'n'.'nnn',''ddd	-USD10.203,408
-n'.'nnn',''dd'USD'	-10.203,41USD
-n'.'nnn',''ddd'USD'	-10.203,408USD

Examples of scripts

This section presents sample scripts sorted according to the different elements that they use.

Basic functions

If, Then, Else, Else If, End If

Syntax

If <condition> **Then**

<Instructions>

Else If <condition> **Then**

<Instructions>

Else

<Instructions>

End If

> Note:
>
> About logical fields (Boolean):
>
> Logical fields are represented as 8-bit integers. The value "true" in Basic is equal to -1.
>
> Certain scripts concerning logical fields can pose problems:
>
> ```
> if [logicalfield] = true Then
> ```
>
> If the value "true" defined for your database is 1 and the value "false" is 0, then, for this script, the value returned will be 1, and "false" as understood in Basic.

Example

```
Dim strVal As String
(...)
```

```
If strVal = "" Then
RetVal = "Empty"
ElseIf strVal = "Default" Then
RetVal = "Default"
Else
RetVal = "Unknown"
End If
```

This script returns the value:

- *Empty* if the text field of a produced document doesn't contain any information.
- *Default* if the text field of a produced document contains the *default* information.
- *Unknown* if the text field of a produced document contains any other information.

For Loop

This function enables you to create a loop.

Syntax

For <counter variable> = <start> **to** <end>

<Instructions>

Next

Example

```
For i=0 To 10 Step 2
PifLogInfoMsg(i)
Next
```

This script returns the value *i* in the Document log.

You will see the following in the Document log:

```
0
2
4
6
8
10
```

While Loop

This instruction enables you to create a loop.

Syntax

While loop

While <conditions>

<instructions>

WEnd

Example

```
Dim i As Integer
i = 0
While i < 10
i = i + 2
PifLogInfoMsg(i)
WEnd
```

This script returns the value *i* in the Document log if this value is less than 10.

You will see the following in the Document log:

```
0
2
4
6
8
10
```

Return

In the script, if the conditions defined before this function are not respected then the rest of the script is ignored.

Syntax

<conditions>

Return

<conditions>

Example

```
If [MacAddress] = "" And [IPAddress] = "" Then
PifIgnoreNodemapping
Return
End If

If [MacAdress] <> "" Then
RetVal = [MacAdress]
Else
RetVal = [IPAdress]
End If
```

This script tests whether the **MacAdress** and **IPAdress** fields of a produced document have not got an empty value. If this condition is fulfilled:

- the current node is ignored

- the end of the script is not executed

Select

This function enables you to execute a block of instructions according to the value of a variable.

Syntax

Select Case <variable to test>

Case <variable 1>

Instruction block

Case <variable 2>

Instruction block

Case <variable 3>

Instruction block

...

Case <variable n>

Instruction block

Case Else

End Select

Example

```
Select Case [seStatus]
Case 0
RetVal = "Opened"
Case 1
RetVal = "Closed"
Case Else
RetVal = "Unknown status"
End Select
```

In this example:
- The source document's **seStatus** field corresponds to the status of a ticket.
- The status of the ticket is:
    - 0 = open ticket
    - 1 = closed ticket

This script associates the character string describing the status of the ticket to the numeric value of the source field. If the status is unknown, the *Unknown Status* value is returned.

Pif functions

The **PIF** functions have been specially developed for Connect-It mapping scripts.

All these specific functions are documented in the online Programmer's Reference, available in the doc sub-folder of the Connect-It installation folder.

PifIgnoreDocumentMapping

This function enables you to ignore the processing of a document.

Syntax

<conditions>

PifIgnoreDocumentMapping("<message>")

<conditions>

("message") enables you to display an error message in the document log for the ignored element.

The specification of a *retval* function implies that the *PifIgnore* function is executed on a field chosen as a reconciliation key.

Example

```
If [MacAddress] = "" Then
PifIgnoreDocumentMapping("Missing MAcAdress")
End If
RetVal = [MacAddress]
```

We use the **MacAddress** field for a reconciliation key. If this field does not contain a value, the document is ignored. The message ● *Missing MacAddress* field is shown in the document log.

PifRejectDocumentMapping

This function enables you to reject a source document and to not send it to the destination connector.

This applies to any element of the document:
- root node
- structure
- collection
- field

Syntax

<instructions>

PifRejectDocumentMapping("message")

("message") enables you to display an error message in the document log for the ignored element.

The specification of a *retval* function implies that the *PifReject* function is executed on a field chosen as a reconciliation key.

Example

```
If [MacAdress] = "" Then
PifRejectDocumentMapping("Missing MAcAdress")
End If
RetVal = [MacAdress]
```

We use the **MacAddress** field for a reconciliation key. If this field does not contain a value, the document is ignored. The message ❶ *Missing MacAddress* field is shown in the document log.

PifIgnoreNodeMapping

This function enables you to ignore any element in a document type.

This element can be:

- The root node of the document
- A structure
- A collection
- A field

The behavior of the *PifIgnoreNodeMapping* function is different depending on whether it concerns a collection or not.

If this instruction concerns a collection, only the current member of the collection is ignored. If you want to ignore all members of the collection, use the *PifIgnoreCollectionMapping* instruction.

Syntax

```
(...)
PifIgnoreNodeMapping("Message")
(...)
```

("message") enables you to display an error message in the document log for the ignored element.

Example

```
If [MacAdress] = "" Then
PifIgnoreNodeMapping
End If
RetVal = [MacAdress]
```

This script enables you to avoid updating with an empty string if the field or the structure containing the **MAC address** field is empty. If the field is populated then the update is performed.

```
If Left([Software.Name], 7)  = "Windows" Then
PifIgnoreNodeMapping
ElseIf Left([Software.Name], 5) = "SunOS" Then
PifIgnoreDocumentMapping
End If
```

This script enables you to ignore the member of a collection if the **Sofware.Name** field of this member is set to *Windows* or *SunOS*.

PifIgnoreCollectionMapping

This function enables you to ignore a collection of a produced document-type during a collection to collection mapping.

For more information about the collection to collection mapping, refer to this guide's section Collection-to-collection mapping [page 106].

Syntax

<instructions>

PifIgnoreCollectionMapping

<instructions>

Example

```
Dim i As Integer
Dim iCount As Integer
Count = PifGetItemCount("Logs")
For i=0 To iCount - 1
If [Logs(i).LogType] = 1 Then
Return
End If
Next
PifIgnoreCollectionMapping
```

For a processing report, this script enables all the members of the **logs** collection to be ignored if there is no error message.

If the document does not contain an error, it is not necessary to carry out such a script. The **ErrorNumber** field contains the number of errors associated to a document.

The previous script can be replaced by the following:

```
If [ErrorNumber] = 0 Then
PifIgnoreCollectionMapping
End If
```

Collections

In this section, you will find different examples of scripts concerning the processing of collections.

Creating members in a collection from a list of values

This section presents a script example enabling you to create a member in a given collection from a list of values from a source document.

In this example:

- This **Software** source field contains a list of values.
- The values are separated by a given separator.

The script:

- Extracts the software names one by one.
- Creates a member in the **SoftInstalled** destination collection.
- Populates the **Name** element with the name of the extracted software.

```
Dim iCount  As Integer
Dim iIndex  As Integer
Dim strSoft As String
Dim lDummy  As Long
Dim strPath As String

' Count of number of values in the "Software" source field
' the software names are separated by the separator (','), for example: "E
xcel, Connect-It,
' AssetCenter"
iCount = CountValues([Software], ",")

' Loops around all the elements in the list to extract them one by one.
For iIndex = 0 To iCount - 1

strSoft = GetListItem([Software], ",", iIndex+1)

' Deletion of spaces around the name of the software
strSoft = Trim(strSoft)

' Creation of the path of the destination collection from the root element
.
' For example, for the third source software, the path "SoftInstalled(3).N
ame" is created
strPath = "SoftInstalled" (& iIndex & ").Name"

' Assigning of the current value of character string software to the path
using
' the function PifSetStringVal.
' The function PifSetStringVal retourns an error code if the path is not v
alid, it is
' necessar to assign in a variable the return value of the function. The f
unction will not
' be applied in the opposite case.
lDummy = PifSetStringVal(strPath, strSoft)
```

```
Next iIndex
```

This mapping script can be applied on any destination-document type element.

To better read the mapping, we recommend that you do the mapping on the collection to which the members must be added.

Warning:

The element indicated by its path when calling on the Basic function **PifSetStringVal** must be present in the destination document-type. In the present example, the **Name** element of the **SoftInstalled** collection must be added by the user in the consumed document-type.

Concatenating members of a collection in a field

In this example:

- The source document contains a collection of values.
- This element's collections are mapped to a destination document-type field.

The source contains the collection of software installed on a computer. The different names of the software must be written in a field containing the list of software, separated by a comma (',').

```
Dim iCollectionCount As Integer
iCollectionCount = PifGetItemCount("SoftInstalled")

Dim strList As String
Dim iItem As Integer

For each element in the collection, recover the name of the software (Elem
ent "Name" of the collection "SoftInstalled") and concatenate it with the
current list.
For iItem = 0 to iCollectionCount - 1

' Add the name separator if the list is not empty
If strList = "" Then
strList = strList & ", "

' Add the name of the software to the current list.
' Note that it is possible to directly use a variable to indicate the numb
er
' of a member in a collection. For example, if the variable of iITem is 3,
the path
' [SoftInstalled(3).Name] will automatically be created from the value of
iItem.
  strList = strList & ", " [SoftInstalled(iItem).Name]
Next iItem

' Assign the variable strList to the target element
RetVal = strList
```

Mapping several fields in a collection

In this example:

- The source document contains several distinct fields.

  Here, the *Address1* and *Address2* have the two possible addresses of a client.

- The value of these fields must be associated to a member of the destination collection.

  Here, the collection *Address*.

For example:

You must therefore:

- Create two members in the destination collection and associated them to the "Adress1" and "Address2" fields.
- Use the collection-duplication function:
  1 Add the *Address* collection to the destination document-type.
  2 Duplicate this collection.

     The *Address#1* collection appears in the destination document-type.
  3 The mapping scripts [Address1] and [Address2] must be applied to the fields *Address.Address* and *Address#1.Address*, respectively.

Ignoring certain members of a collection in a collection-to-collection mapping

To ignore certain members in a collection, you must use the *PifIgnoreCollectionMapping* and *PifIgnoreNodeMapping* instructions.

For more information about these instructions, refer to the section PifIgnoreCollectionMapping [page 139].

Script concerning a connector not included in a mapping

The following example describes the integration in a scenario (which concerns the replication of data between a database and a ServiceCenter database) of an AssetCenter database. The script imports an employee. During the import, the script verifies whether the employee exists in the AssetCenter and changes the mapping accordingly.

1 Add an Asset Management connector to your scenario. This connector is not required to be linked to a mapping box or another connector, just its title is important (**Connector name** field of the connector configuration wizard) because it will be used in the script. Here, the connector is called Asset Management.

2 Create a new document type produced by the Asset Management connector. Select the Departments and Employees table (**amEmplDept**) and call the produced document type (**Document type** field) **amEmplDeptForMapping**. This name will be used in the script.

3 In the mapping box, populate the script field as follows:

```
dim hQuery as long
dim iRc as long

hQuery = pifNewQueryFromFmtName("Asset Management", "amEmplDeptForMappi
ng", "Name like 'A%'")

Dim strValue as string

while (iRc = 0)
iRc = pifQueryNext(hQuery)
if iRc = 0 then
strValue = pifQueryGetStringVal(hQuery, "Name")
piflogInfoMsg strValue
end if
wend

iRc = pifQueryClose(hQuery)
```

Syntax of the pifNewQueryFromFmtName function

This function creates a query on a document type first defined in the list of documents produced by a resource.

The parameters of the function are as follows:

- **strCntrName**: This parameter contains the name of the resource (on which the query is performed).
- **strFmtName**: This parameter contains the name of the document type (that has been defined as the produced document type).
- **strLayer**: This parameter contains the production directives (for example, a WHERE clause).

The function returns a handle (here, the **hQuery** parameter). This handle must be passed as a parameter to the **PifQueryNext** in order to browse the list of returned records.

The data from the current document can then be recovered using one of the following functions (depending on the field type):

- pifQueryGetStringVal
- pifQueryGetDateVal
- pifQueryGetDoubleVal
- pifQueryGetLongVal
- pifQueryGetIntVal

Each of these functions has two parameters:

- The handle (hQuery) of the query to use (32-bit long integer)
- Path of the element for which we want to recover a value. This path must not contain the name of the root element of the document type (**amEmplDept** in this example).

In this example, the function returns the name of the employee:

```
strValue = pifQueryGetStringVal(hQuery, "Name")
```

Production directives of the pifNewQueryFromFmtName function

The **pifNewQueryFromFmtName** function uses simple parameters. You can, however, define complex queries in XML format.

The production directives can be given in XML, using the following syntax:

```
strLayer =              "<Directives>"
strLayer = strLayer +   "<Where>Name = 'Taltek'</Where>"
strLayer = strLayer +   "<OrderBy>BarCode</OrderBy>"
strLayer = strLayer +       "<Where Path='ItemsUsed'>AssetTag like 'A%'</Wh
ere>"
strLayer = strLayer + "</Directives>"

hQuery = pifNewQueryFromFmtName("Asset Management", "amEmplDeptForMapping"
, strLayer)
```

XML syntax

The &, < and > characters are not authorized. You must replace them with &, < and > respectively. The Basic function **GetXmlElementValue** handles the substitution of these characters.

For example:

```
strLayer = strLayer + "<Where Path='ItemsUsed'>" + GetXmlElementValue("Ass
etTag like 'A%'") + "</Where>"
```

Query on fields containing a period or comma

In the following commented example, a query involves the elements **mac.address** and **logical.name** of an OpenView Network Discovery - ServiceCenter scenario. The script validates the MAC address provided by the IND connector before assigning it a reconciliation key. If the MAC address is validated, the information from the **logical.name** field is recovered instead of from the **mac.address** field.

```
dim hQuery as long
dim iRc as long
dim strQuery as String
```

```
strQuery = "mac.address = " & chr(34) & [MACAddress] & chr(34)
"MAC address in the PDI document

hQuery = pifNewQueryFromFmtName("ServiceCenter", "pc1", strQuery)
"pc1 is the document produced for the ServiceCenter Computers table

Dim strValue as String
strValue = [MACAddress]
"strValue by defaut

iRc = pifQueryNext(hQuery)
if iRc = 0 then           "
"query finished because iRc=0

strValue = pifQueryGetStringVal(hQuery, "'logical.name'")
"Single quotes define the parameter logical.name as a field and not a path

pifLogWarningMsg("Matched Asset using query: " & strQuery)
"write to document log

pifLogWarningMsg("Updating Asset " & strValue)
"strValue is not written to the document log
else
pifLogWarningMsg("Could not locate existing asset using MAC address " & [M
acAddress])
end if

iRc = pifQueryClose(hQuery)

If strValue = "" then
"This code is executed when pifQueryNext returns 0.

pifLogWarningMsg("pifQueryGetStringVal returned no data. Logical.name will
be " & [MACAddress])
RetVal = [MACAddress]
Else
RetVal = strValue
End If
```

Help for writing mapping scripts

This section presents certain basic concepts that will help you write your mapping scripts.

Processing order for mapping scripts

When for a given document type, there are several mapping scripts, the scripts are processed in the following order:

1 Script associated with a field
2 Script associated with a structure
3 Script associated with a collection

This order of processing enables faster error handling.

Map a fixed value to a destination element.

To assign a fixed value (a date, string, number, etc.) to a destination field:

1 Drag the destination field on to the work area.

2 Enter the fixed value in the **Mapping script** field.

Example :

In the Desktop Discovery to Asset Management scenario, each scanned computer results in the creation of a new asset in the **amAsset** table. By default, this table requires each asset to have a category name. In your mapping definition, you map the fixed value **"/Hardware/Desktop computer/"** to the **FullName.Category** field of the **amAsset** document-type.
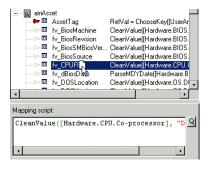
Moving several fields by drag-and-drop

In your Basic scripts, you can use several source fields, for example, to concatenate them using a Basic function.

To move several fields by drag-and-drop:

1 Map a first-source field to a destination field.

2 Drag the other fields while holding down the **Shift** key. The fields appear as a list in the **Mapping script** pane. Simply rewrite the script using these fields.

Figure 4.19. Moving several fields by drag-and-drop

Locating source and destination elements

To find the original location of a destination element that is on the work area:

1 Select this element in the work area.

2 Click ↩

This action places the focus on the destination element in the destination document-type pane. This function is useful when the destination document-type contains a large number of elements.

Mapped source elements (whose names appear in blue) can also be located in the work area.

To do this, double-click a mapped source element in the source document-type pane. All destination elements that involve this source element are displayed in green.

Duplicating a destination element
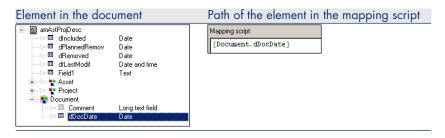
Sometimes you need to duplicate a destination field in order to map several source fields to it. This is particularly the case for collections. For example: You want to map fields X and Y to field A of a collection. To do this, you need to duplicate the collection containing field A, and map field X to field A in the original collection, and then field Y to field A in the duplicated collection.

To duplicate an element:
1   Select an element in the work area.
2   Click

Duplicated elements are numbered: The original element is numbered 0 (this is not displayed), the first duplicate 1, the second 2, and so on. This numbering system is imposed by Connect-It and cannot be modified.

Copying the path of an element

Element in the document	Path of the element in the mapping script

In a mapping script, you must indicate the full path of the elements that you use. Example: In the following image, the full path of the *dDocDate* field is *[Document.dDocDate]*.

To obtain the path of a document-type element:
1   Select this element.
2   Right-click and then choose **Copy path** from the contextual menu (*Ctrl+C*).
3   Place the pointer in the **Mapping script** field.
4   Right-click and then choose Paste from the contextual menu (*Ctrl+V*).

Verifying the syntax of a script

In a mapping script, click the magnifier to access a text editor. This editor enables you to write mapping scripts and validate their syntax.

To verify the syntax of a mapping script:

1 In the **Tools** menu, select **Options**
2 In the configuration window that is displayed, select the script template according to the connector used (**Template** field):
   - **aql** to write an AQL query
   - **bas** for a Visual Basic script
   - **sc** for a ServiceCenter script
   - Etc.
3 Once you have finished entering the script, click **OK**.

Validating a Basic script

A script entered into the **Mapping**, **Additional scripts** or **Reconciliation scripts** tabs is parsed and a error message is shown if they are not valid. In this case, you are able to copy the script but not validate it.

Type-ahead in Basic scripts

To access the dialog box that enables you use type-ahead functionality when entering functions, press CTRL+Spacebar.

Using the contextual menu in mapping scripts

When you right-click on the mapping script, a contextual menu is displayed.

This contextual menu includes the following functions (the keyboard shortcuts are given in parentheses):

- Undo Keystroke
- Repeat Keystroke
- Cut (*Ctrl+X*)
- Copy (*Ctrl+C*)
- Paste (*Ctrl+V*)
- Select all (*Ctrl+A*)
- Filter unused document types
- Show source
- Show destination

Contextual menus in the mapping edit window

In the mappings edit window, the commands in the four contextual menus help you to write your mapping scripts.

Pointer placed on an element of a document type published by a source connector

Command	Function
Copy this XML DTD	Copies the DTD of the document type produced by the source connector.
Copy path	Copies the path of the selected element to the Clipboard.
Show nodes mapped to the selection	Displays in green the nodes of the mappings concerning the selected element (direct or scripted mapping).
View data	Displays a window in which you can view the data from the source document type.
Edit a document type to preview data	Displays a window in which you create a document type produced by the source connector and view its data.
Filter unused document types	Filters the document types that are not used in the current mapping.
Show source	Displays or hides the source document-type pane.
Show destination	Displays or hides the destination document-type pane.

Pointer placed on one of the elements of a document type published by a destination connector

Command	Function
Copy this XML DTD	Copies the DTD of the document type consumed by the connector.
Copy path	Copies the path of the selected element to the Clipboard.
Add this element	Places the element on the work area.
View data	Displays a window in which you can view the data from the destination document-type.
Filter unused document types	Filters the document types that are not used in the current mapping.
Show source	Displays or hides the source document-type pane.
Show destination	Displays or hides the destination document-type pane.

Pointer placed in the work area

Command	Function
Copy this XML DTD	Copies the DTD of the document type consumed by the connector.
Copy path	Copies the path of the selected element to the Clipboard.
Edit the mapping	Enables you to edit the mapping script using the text editor.
Describe mapping	Displays a window in which you can describe the selected mapping node.
Reconciliation key	Specifies that the selected element is used as a reconciliation key in the case of a database-type destination connector.
Remove this element	Removes this element from the work area.
Copy mapping	Enables you to copy all or part of the mapping to the Clipboard.

Command	Function
Paste mapping	Enables you to paste all or part of a mapping from the Clipboard to the current mapping.
Filter unused document types	Filters the document types that are not used in the current mapping.
Show source	Displays or hides the source document-type pane.
Show destination	Displays or hides the destination document-type pane.

Pointer placed in the **Mapping script** edit zone

Command	Function
Undo keystroke	Undoes last keystroke
Repeat keystroke	Repeats the last keystroke.
Cut	Cuts the selected text.
Copy	Copies the selected text to the Clipboard.
Paste	Pastes the Clipboard's contents into the edit zone.
Select all	Selects all the text in the editable zone.
Filter unused document types	Filters the document types that are not used in the current mapping.
Show source	Displays or hides the source document-type pane.
Show destination	Displays or hides the destination document-type pane.

Defining a global variable

You can define global variables for your mapping scripts.

A global variable is a variable that was defined at the root of a document tpe and can be called on by a script associated with any complex element of a document type (collection, structure, field).

You must declare a global variable before defining the Basic script. The variable is taken into account when you open or close a session.

For further information on the functions to use, refer to the section Basic functions [page 133].

You can define a global variable in the **Additional scripts** tab.

For further information on the **Additional scripts** tab, refer to the section Additional scripts [page 150].

It can also be declared in an associated Basic file.

Additional scripts

The additional scripts are defined in the **Additional scripts** tab for a given document type.

This tab is only active for the root element of a document type.

It enables you to:

- Define global variables.
- Execute Basic code at the opening of each session.
- Execute Basic code at the closing of each session.

📝 Note:

A sample of additional script is shown for all created document types.

Define the supervision monitors

The scenario monitoring functions are handled by a system of monitors. These monitors enable and handle the flow of information, and processing history from the application and its connectors.

Monitors - How they work

The monitors are surveillance tools that may be enabled or disabled and make it possible to display and save information related to the processing of a scenario.

The monitors are enabled when creating a scenario.

Monitoring a scenario involves:

- Keeping a graphical trace of the document log
- Keeping the information linked to the documents that have passed through the source and destination connectors:
  - The documents themselves or the documents contains a processing error (warning, rejection in part or in full)
  - The document logs
  - the Connect-It log
  - The parent documents
- Keeping statistics on document processing
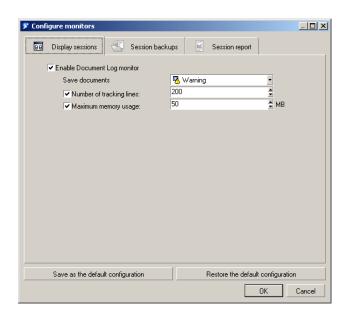
⚠️ Warning:

When you create a new scenario, the default configuration is used. We recommend verifying the configuration of the monitors before starting the first tests and putting the scenarios into production.

Each scenario has its own monitor configuration saved in the .scn scenario file. Additionally, Connect-It has a default configuration saved in the conit.ini

file; This default configuration is used when Connect-It is started up and on creating a new scenario.

Configuring the monitors

The monitors are available via the **Monitors / Configure monitors** menu.

The scenario monitoring settings are defined in each tab.

The monitoring is enabled via the following tabs:

- Display sessions: Display and configure information shown in the **Document log** tab.
- Session backups: Save session-related information (logs, documents, statistics) to file or the Connect-It database.
- Session report: Process session reports in E-mail, Windows event or AssetCenter event formats.

Each tab represents one or more monitors that may or may not be enabled or independent.

The enabling of each monitor depends on the general options validated for each session for all monitors.

> ✎ Note:
>
> The performance of the application when the monitors are enabled depends on the number of documents produced and consumed and saved in the monitors.

Display sessions monitor

This monitor is used to display the information in the **Document log** tab. Enabling this monitor impacts connector performance and is dissociated from the **Session backups / Keep the logs linked to documents** option in the **Session backups** log.

To enable the monitor, select the **Enable Document Log monitor** option in the **Display sessions** tab.
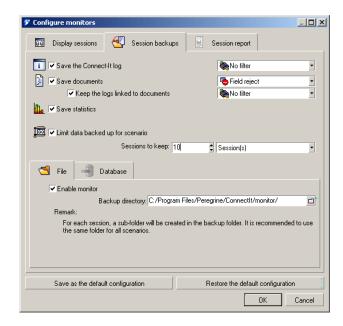
This monitor can be configured. You can specify:

- The condition for displaying the documents
- The number of tracking lines displayed in the document log
- The allocated memory size, between 512 KB and 20 MB

The contents of this monitor can be seen in the **Document log** tab of your scenario editor.

Session options

The options in this tab enable you to define which items of history are handled by the application and are saved to file or the Connect-It database.

In the **Sessions backup** tab, select the options handled by each monitor:
- Save the Connect-It log
- Save documents
- Save statistics

Session options handled by the monitors

Depending on the type of monitor selected, the options defined for the session processing will be available or not.

	Display sessions monitor	File monitor	Database monitor	E-mail monitor
Application log	not available	No	Yes	No
Documents	Yes	Yes	Yes	No
Document log	Yes	Yes	Yes	No
Statistics	No	Yes	Yes	Yes

Save Connect-It logs option

This option enables you to activate the Connect-It log.

For further information, refer to chapter Managing an integration scenario [page 191], section The Connect-It log [page 213]

Save documents option

This option enables you to configure:

- The information that is visible in the **Document log** tab of the scenario editor.
- The information that is saved in the **File** and / or **Database** monitors.

Configuring the documents saved consists of:

- Filtering the saved documents.

  This filtering is obtained using the drop-down list available for the **Save documents** option.

  Filtering the documents saved enables you to save:

  - The documents themselves (**normal processing** filter)

    This option is mainly used when testing your scenarios.

  - The warnings associated with documents (**Warnings** filter)
  - Partial rejections
  - Full rejections

- Filtering the logs associated with the documents you wish to keep.

  This filtering is obtained drop-down list in the **Keep the logs linked to documents** option

For more information about the document log, refer to section Document log [page 214]

Save statistics option

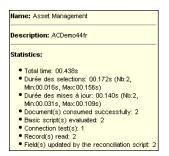This option enables to activate the production of statistics.

The following statistics are produced:

- Document statistics: Documents produced and documents consumed
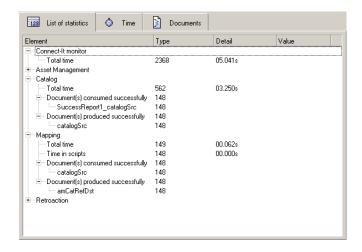- Time statistics: Length of the session, length of processing by the connector

These statistics highlight the key elements in a scenario (insertion, deletion, or updating of documents).

These statistics are:

- visible in the scenario builder when the pointer is on one of the mapping elements.

Name: Asset Management

Description: ACDemo44fr

Statistics:

- Total time: 00.438s
- Durée des selections: 00.172s (Nb:2, Min:00.016s, Max:00.156s)
- Durée des mises à jour: 00.140s (Nb:2, Min:00.031s, Max:00.109s)
- Document(s) consumed successfully: 2
- Basic script(s) evaluated: 2
- Connection test(s): 1
- Record(s) read: 2
- Field(s) updated by the reconciliation script: 2

- visible in the screen for managing saved sessions (**Monitors/Manage saved sessions** menu) when the **File** monitor has been enabled.

Element	Type	Detail	Value
Connect-It monitor			
— Total time	2368	05.041s	
Asset Management			
Catalog			
— Total time	562	03.250s	
— Document(s) consumed successfully	148		
— SuccessReport1_catalogSrc	148		
— Document(s) produced successfully	148		
— catalogSrc	148		
Mapping			
— Total time	149	00.062s	
— Time in scripts	148	00.000s	
— Document(s) consumed successfully	148		
— catalogSrc	148		
— Document(s) produced successfully	148		
— amCatRefDst	148		
Retroaction			

- sorted by element type (connector, mapping box) and by value (duration, success, failure).

Limit data backed up for scenario option

This option enables you to set maximum operating values (time, number) for the monitors:

- For each execution of a scenario, i.e from the time the application is opened until the time it is closed
- For each session, i.e. the number of times a scenario has been launched

The newest sessions replace the oldest ones.

Saving the general options

The configuration of your monitoring tools is saved for each scenario. You can save a default configuration, which will be saved in the `conit.ini` file. This configuration will be loaded each time the application is launched.

To save your configuration, click **Save the default configuration**.

The default configuration is applied to all new scenarios created. It is also applied to all the scenarios shipped with Connect-It.

Note:

The default configuration will not be applied to scenarios created or updated using previous versions of Connect-It. In this case, the scenario configuration defined in the previous version of Connect-It will be kept.

This configuration can be restored at any moment by clicking **Restore the default configuration**. This button allows you to apply the default configuration to the scenario being edited.

Session options - File monitor

This monitor saves the contents of the documents and the associated document logs to file.

To enable this monitor, select the **Enable monitor** option in the **File** tab.

This monitor enables you to specify:

- The folder in which the general settings are saved.

Note:

The **Backup directory** field allows you to enter either the absolute or relative paths of your files. For example:

- `c:\test\<Generic name>`
- `..\<Generic name>`

This folder contains all the sessions defined for the **Limit data backed up for scenario** option. Each session has a corresponding number.

Note:

This folder is used to import saved sessions (**Monitor** menu).

There are as many session folders as there are scenarios executed. The maximum number of session folders for each scenario is set by the **Sessions to keep** field. Each sesion folder has a session number and is saved in a folder bearing the name of the scenario.

The main folder containing the scenarion sessions can be named as you like. This makes it possible to save the sessions pertaining to a given scenario in a new folder. We do however recommend saving your sessions to the same main folder.

All sessions are listed in the `seslist.xml` file and all scenarios are listed in the `scnlist.xml` file.

Each session folder contains the processing history in the form of multiple files:
- The `doclog` and `document` files in the `xml` format
- The `doctrace` and `stat` files in text format

Warning:

If you have set the **Save documents** option to **Normal processing (no filter)**, all processed documents are saved in the database; In this case, performance may be impacted and you may run the risk of saturating your backup media.

Session options - Database monitor

To enable the monitor, select the **Enable monitor** option in the **Database** tab.

This monitor enables you to specify:
- The Connect-It database used to store processing history (**Connection** field)

  The database connection must first be declared in your database manager.

  If the connection does not exist, you can create it using the database administration tool.
- The database login
- The database password
- The temporary disabling of history for tracking lines

Note:

You can test the validity of your connection by clicking **Test**.

For further information on administering a Connect-It database, refer to the chapter entitled Monitors database administration [page 261].

Warning:

If you have set the **Save documents** option to **Normal processing (no filter)**, all processed documents are saved in the database; In this case, performance may be impacted and you may run the risk of saturating the database.

We recommend saving documents with errors.

Because of changes to the database structure, it is not possible to load sessions prior to Connect-It version 3.4.

Temporary disabling of tracking-line groups

The option **Disable temporarily if the number of tracking-line groups to save exceeds:** enables you to define an upper limit for saving tracking-line groups. This option is particularly useful when a large amount of errors are encountered that are obviously the result of a mapping error.

A tracking-line group corresponds to a set of information on the processing of documents. This information includes: The document produced by the source connector, by the mapping box, the document consumed by the destination connector, and the parents documents at the origin of the document.

If the value defined for the option is exceeded, the monitor is disabled for the rest of the session and an error is entered in the Connect-It log.

Session report - E-mail monitor

This monitor enables you to send an e-mail at the end of each session, containing formatted statistics concerning the session.

To enable this monitor, select the **Enable monitor** option in the **E-mail** tab.

This monitor enables you to retrieve and format the processing history.

You can specify:

- When an e-mail should be sent:
  - Normal processing (no filter)
  - Warning
  - Partial rejection
  - Total rejection
- The messaging protocol: MAPI, SMTP, VIM
- The name of the messaging server, its port number
- Information related to the message: Return address, recipient, subject

  The subject of the message can be automated using the following variables:

Table 4.4. E-mail / Variables monitor

%Status%	Status of the session
%StatusDesc%	Description of the status of the session
%DocAll%	Total number of documents processed
%DocOk%	Number of documents successfully processed
%DocWarning%	Number of documents processed with warnings
%DocError%	Number of documents processed with errors
%DocRejected%	Number of documents processed and rejected

%DocIgnored%	Number of documents processed and ignored
%DocConsumed%	Number of documents consumed
%DocConsumedOk%	Number of successfully consumed documents
%DocProduced%	Number of documents produced
%DocProducedOk%	Number of documents produced with success
%DocConsumedWarning%	Number of documents consumed with warnings
%DocConsumedError%	Number of documents consumed with errors
%DocConsumedRejected%	Number of documents consumed and rejected
%DocConsumedIgnored%	Number of documents consumed and ignored
%DocProducedWarning%	Number of documents produced with warnings
%DocProducedError%	Number of documents produced with errors
%DocProducedRejected%	Number of documents produced and rejected
%DocProducedIgnored%	Number of documents produced and ignored
%ScnName%	Scenario name
%ScnFullName%	Full path of the scenario
%SessionLength%	Length of the session
%StartTime%	Start time of the session
%EndTime%	End time of the session

The names of the variables are not case-sensitive

- The format of the message (**E-mail type** field): *Text*, *HTML* or *XML*. The format is defined via the **Modify** button.

  For the text file, specify the number of character per line (**Maximum text width** field).

  For the HTML format, select the **Use a custom stylesheet** option if you want to use a style sheet other than the default one.

  The default template is the following:

```
BODY {
FONT-FAMILY: helvetica;
BACKGROUND-COLOR: #ffffff
}
H1 {
PADDING-RIGHT: 5px;
PADDING-LEFT: 5px;
FONT-WEIGHT: bold;
FONT-SIZE: 150%;
PADDING-BOTTOM: 5px;
COLOR: #ffffff;
PADDING-TOP: 5px;
BACKGROUND-COLOR: #000066
}
TD.ItemDesc {
FONT-WEIGHT: bold
}
SPAN.ItemDesc {
FONT-WEIGHT: bold
}
P.NotAvailable {
```

```
FONT-STYLE: italic;
TEXT-ALIGN: center
}
TR.Header {
FONT-WEIGHT: bold;
COLOR: #ffffff;
BACKGROUND-COLOR: #000066
}
TD.Group {
FONT-WEIGHT: bold;
FONT-SIZE: 125%
}
TR.GraphDoc {
BACKGROUND-COLOR: #e0e0e0
}
TD.GraphDoc {
BACKGROUND-COLOR: #000066
}
TR.GraphTime {
BACKGROUND-COLOR: #e0e0e0
}
TD.GraphTime {
BACKGROUND-COLOR: #000066
}
TD.DocOK {
BACKGROUND-COLOR: #00e000
}
TD.DocWarning {
BACKGROUND-COLOR: #e0a800
}
TD.DocError {
BACKGROUND-COLOR: #e07000
}
TD.DocRejected {
BACKGROUND-COLOR: #e07000
}
TD.DocIgnored {
BACKGROUND-COLOR: #0000e0
}
```

For the XML format, the message body is written in XML using the following tags:

```
<Report><Section><Start><End><Status><Docs><Time><TotalTime><Group>
```

- The contents of the e-mail message:
  - Execution report
  - Statistics on the documents in the scenario
  - Time-related statistics on the scenario
  - List of statistics sorted by element
  - List of statistics ordered by value type

The contents of the e-mail message is presented in the form of options that can be enabled or not. This options can be sorted by order of preference

(**Move up** and **Move down** arrows). The number of options selected is shown in the **Contents of the e-mail** field in the **E-mail** tab.

Session report - ServiceCenter monitor

This monitor enables you to create a record in the ServiceCenter Incidents table.
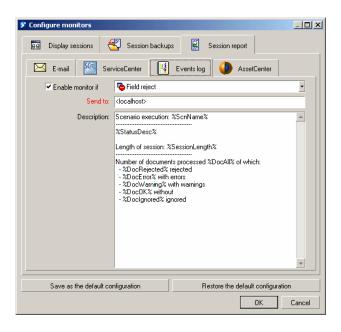
The monitor parameters are the following:

- **Connection** field: ServiceCenter database. For example, Test:12670
- **Login** field: connection identifier
- **Password** field: User password
- **Category**: incident category
- **Severity** field: incident severity
- **Problem type**: Problem type giving rise the incident
- **Description** field: description of the session as shown in the **Description** field of a ServiceCenter work order.

The syntax of the template is the same as that used for the E-mail monitor (E-mail / Variables monitor [page 159]).

Session report - Events log monitor

This monitor enables you to create an event that you can seen in the **Event Viewer** in Windows.

To enable this monitor, select the **Enable monitor if** option in the **Session report** tab and then define the filter associated with this monitor (Normal processing, Field reject, Warning, Reject).

The monitor parameters are the following:

- **Send to** field: Name of the computer (or group of computers) to receive the event
- **Description** field: Description of the event as shown in the **Description** field of the Windows event

  The default template is the following:

```
Scenario execution: %ScnName%
-------------------------------------------
%StatusDesc%

Length of session: %SessionLength%
-------------------------------------------
Number of documents processed %DocAll% of which:
- %DocRejected% rejected
- %DocError% with errors
- %DocWarning% with warnings
- %DocOK% without errors
- %DocIgnored% ignored
```

The syntax of the template is the same as that used for the E-mail monitor (E-mail / Variables monitor [page 159]).

Session report - AssetCenter monitor

This monitor enables you to create a record in the AssetCenter Work Orders (amWorkOrder) table.

We recommend creating a specify work order model in AssetCenter in order to handle and categorize the records inserted by Connect-It.

To enable this monitor, select the **Enable monitor if** option in the **Session report** tab and then define the filter associated with this monitor (Normal processing, Field reject, Warning, Reject).

The monitor parameters are the following:

- **Connection** field: AssetCenter database declared in the DBMS
- **Login** field: connection identifier
- **Password** field: User password
- **Model** field: Name if the existing model of nature **Work order**
- **Title** field: Name of the scenario populating the **Label** field in the **Work orders** table
- **Group** field: Name of the group in the **Departments and employees** table (amEmplDept)

- **Problem** field: description of the session as shown in the **Problem** field of an AssetCenter work order.
- **Status** field (not visible): This field returns the value 0 (Notified status) for the **Status** (seStatus) field of a work order.

The syntax of the template is the same as that used for the E-mail monitor (E-mail / Variables monitor [page 159]).

To test the validity of the connection, click **Test the connection**.

Management of saved sessions

History for each session can be seen in the session manager.

Each saved session has a tab for each declared monitor.

Click a session to view its history in each tab (list of statistics, durations, document, database, e-mail).

Each selected session can be:
- Opened in a new scenario editor (**Open** button): The scenario is shown with its logs (Connect-It log, document log)
- Loaded in the current window of the scenario editor (**Load** button): The current scenario is shown with its logs (Connect-It log, document log)
- Deleted (**Delete** button): The session history is deleted
- Closed (**Close**): The session management window is closed

Session options

The session options, which are defined when configuring the monitors (**Monitors/ Configure monitors** menu, **Session backups**), impact the data that is saved.

For example, if you choose to save only those documents whose status is **Warning**, when you load the session (**Monitors/ Managed saved sessions** menu), you will not see the documents that have been processed normally.

Configuring saved sessions

You may apply a display filter to your sessions in order to reduce the number displayed. A session is displayed according to the following conditions:
- scenario name
- date of modification of the session
- status of the session
- number of sessions to display

To access the filter parameters, click **Configure**. In the window that is displayed you can:

- create a new filter
- modify an existing filter
- delete an existing filter

📝 Note:

The pre-existing system filters cannot be edited.

Session-filter conditions

Each filter is an option that can be enabled or left disabled.

- **Filter name**: Name of the filter displayed in the list available in the **List of available sessions** field in the **Manage saved sessions** window. If no name is specified, a name is automatically assigned describing the options used.
- **Display the sessions of the scenario**: When this option is selected and a scenario name specified, only the session related to the scenario is displayed.

📝 Note:

If this option is selected but without a scenario name specified, the filter will be applied to the current Connect-It scenario session.

- **Display the sessions modified since**:
  - Option not selected: The filter does not take the last date of modification of the session into account.
  - Option selected: The filter enables you to select a duration. For example, for 12 hours, only the sessions modified or completed in the last 12 hours are displayed.
- **Display the sessions with the status**: This option enables you to filter the status of the session. When this option is selected, the filter is applied according to a hierarchic order. Filtering on a **Warning** status implies that all sessions with status **Warning**, **Field reject** and **Reject** are displayed and only sessions without errors are not displayed.
- **Maximum number of session displayed**: This option enables you to define the maximum number of sessions displayed.

📝 Note:

The most recent sessions cannot be displayed if the maximum number has been overrun.

Loading saved sessions

The **Load** button enables you to load a session and apply a filter to it. The information available in the **Document log** tab is filtered according to a given filter:

- Normal processing (no filter)
- Warning
- Partial rejection
- Total rejection

The loaded sessions come from:

- the session backup folder
- the Connect-It database

You can choose to display the first or last N tracking lines only (**Only reload a portion of the tracking lines** option).

Filtering according to the processing status of documents and restricting the number of tracking lines displayed enables you to reduce the time taken to process the lines displayed.

> Note:
>
> If there is a large number of documents to be loaded (500 for the file monitor, 1000 for the database monitor), a warning message is displayed.

Importing saved sessions

You can import session files from other scenarios into the current scenario (**Monitors/ Import saved sessions**).

Doing this is usefule when transferring information from one workstation to another. The principle is as follows:

1 The session folder and related files from the first workstation (`.scn`, `.dat`, `.xml`, `.dtd`, `.ini`, etc.) are archived to a `.zip` file.
2 This folder is copied and unzipped to a folder such as c:\tmp\test on the target workstation; the related files are copied to their original locations (`.ini` to the user settings, the scenario at the same level as the session folder, etc.).
3 Connect-It is started and the scenario loaded
4 The **File** folder is enabled and populated with the path of the unzipped session folder (**Monitors/ Configure monitors**, tab **Session backups**, sub-tab **File**)

5   The session import is used (**Monitors/ Import saved sessions**)

The link tying together the sessions on the original workstation is restored
and the new sessions are now saved in the specified folder.

Define a Connect-It service (Windows environment)

Connect-It enables you to associate a Windows service to a scenario. This service
enables your Connect-It server to start the data-processing procedure as a
background task, depending on the scheduler associated to your scenario.

 Warning:

If one of your scenario's connectors uses an ODBC connection, it absolutely must
use a system data source (system DNS). If this connection uses a user data source
(user DNS), it will not be visible to the service to which it is associated.

A service under Windows enables you to use an application that runs as a
background task. Connect-It enables you to create and launch as many services
as there are scenarios.

To create, start, stop or delete a service, you must have local administration
rights in Windows.

 Note:

If you modified an environment variable in a 32-bit Windows (the search path, for
example), this modification is not applied to the service until you restart the computer.

 Important:

If a connector specifies a folder or a file on another computer, it cannot indicate the
drive mapped by the user, instead it should reference the remote computer. For
example:

```
\\XSFStore\Scan
```
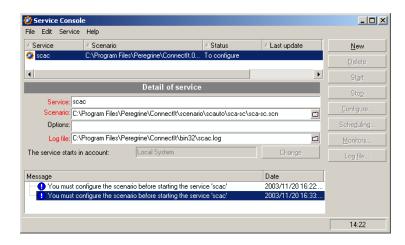
and not:

```
Z:\Scan
```

if the Z drive is mapped to \\XSFStore.

The Service Console

The Service Console is a component of Connect-It. It is a graphical interface that enables you to manage your scenarios and create services to associate to them.

Figure 4.20. The Service Console

To launch the console, select **Service Console** in the HP OpenView \ Connect-It program group. You can also launch the console.exe executable file in the bin folder of the Connect-It installation folder.

Menus

The console has four menus:

File menu

Command	Function
Exit	Enables you to exit the Connect-It console.

Edit menu

Command	Function
Cut, copy, paste	Provides the standard editing features for the console.
Delete messages	Deletes the messages and alerts shown in the console

Command	Function
Refresh	Refreshes the list of services

Command	Function
New	Enables you to create a new Connect-It service
Delete	Enables you to delete a Connect-It service.
Install	Enables you to install a Connect-It service.
Uninstall	Enables you to uninstall a Connect-It service.
Start	Enables you to start a Connect-It service.
Stop	Enables you to stop a Connect-It service.
Configure	Enables you to configure a scenario's connectors by launching the **Configure the connector** wizard.
Scheduling	Enables you to modify the schedulers that are used in the scenario associated to the service.
Monitors	Enables you to launch the Scenario builder by displaying the **Connect-It log** and the **Document log** tabs.
Log file	Enables you to view the log file that accompanies each Connect-It service.

Help menu

Command	Function
About...	Enables you to display the **About** box.
Online help	Enables you to display the online Help

Steps in creating and using a Connect-It service in Windows

Creating a Windows service using the service console does not necessarily imply using it.

You must adhere to the following steps:

1 Create the service
2 Configure the service
3 Install the service
4 Start the service

Creating a Connect-It service in Windows

You can create a Connect-It service using the following tools:

- The Service Console
- From the command line

To create a Connect-It service from the command line

1 Open a command prompt in the **bin** folder of the Connect-It installation folder
2 Execute the csvctool command using the syntax defined in chapter

To create a Connect-It service using the service console

1 Launch the Service Console.
2 Click **New**.
3 Name your service in the **Service** field.
4 Enter the path and the name of your scenario by clicking 📁.
5 Specify a command option if necessary (**Options** field). For example, you can specify the Forcecnx option to require an automatic reconnection.
6 Enter application log file (.log).

   This field is automatically populated if you do not specify a log file.
7 Click **Create**.

   The created service appears in the upper pane of the console. Its status is **To configure**.

   🖉 Note:

   The service is only properly created once each scenario connector is configured (**Configure** button).

Service connection properties

In certain cases, you must modify the service connection properties in Windows and use another user account so that it can function correctly.

The properties of the default connection to the service are saved in the **LocalSystem** account.

Examples

- The source connector of the scenario linked to the service reads the files in the folder on the network drive to which the service's local account does not have access.
- The E-mail connector of a scenario only works with one particular account.

In these two cases, you must specify a particular account in the service connection properties.

You can specify an account directly via the operating system or by using the **Change** option in the Connect-It Service Console (**The service starts in account** field).

To modify a service's connection properties in Windows XP

1 Start Windows XP

2 Select the **Start/ Settings/ Control Panel/ Administrative Tools/ Services** menu.

3 Double-click the service in the list.

4 Select the **Log On** tab in the dialog box that appears.

5 Select the **This account** option.

6 Enter the values that enable you to properly use the scenario linked to your service in the **This account** and **Password** fields.

7 Click **Apply**.

To modify the connection properties of a service usng the Service Console

1 Click **Change**

2 Select the new account to use (**This account** option) and populate the **Password** field

3 Click **Validate**

Scenario path

You must respect certain rules when indicating a scenario's path in the **Scenario** field:

1 The scenario is saved on the Connect-It server.

If the scenario is located in a folder associated with a network drive, do not indicate the letter of this drive in the **Scenario** field.

Example: The scenario\myscenario.scn scenario is located in the folder C:\Program Files\ConnectIt associated with the drive R:. In this case, you would enter C:\Program Files\ConnectIt\scenario\myscenario.scn in the **Scenario** field instead of R:\scenario\myscenario.scn. This is because the service usually uses the local system account or a specific account, which have no associations between the folder C:\Program Files\ConnectIt and the network drive R:.

2 The scenario is installed on the network.

Verify that the service can use the network drive indicated in the **Scenario** field.

In order for the service to be able to use a network drive, you must associate this service with a user name and password that can use this network drive.

Starting or stopping a Connect-It service

To start a Connect-It service:

- Launch the Connect-It console.
- Select the service in the main pane of Connect-It.
- Click **Start**.

Note:

You need to have already configured a scenario in order to start it.

To stop a Connect-It service:

- Launch the Connect-It console.
- Select the service in the main pane of Connect-It.
- Click **Stop**.

Deleting a Connect-It service

To delete a Connect-It service:

- Launch the Connect-It console.
- Select the service in the main pane of Connect-It.
- Click **Delete**.

Define a command line service

Creating a service from the command line

To create a service from the command line:

```
csvctool -create [name of the service]
```

For further information on command lines, refer to chapter Connect-It from the command line [page 193], section Command options used [page 194].

Launching a service from the command line

To start a service from the command line:

1   Go to the *bin* folder in the Connect-It installation folder

   Example: `C:\HP OpenView\Connect-It\bin\`

2   Execute the following at the command line:

   `net start <Name of the service>`

Example :

If you launch the **Asset Management-ServiceCenter** service, your command line is:

```
net start Asset Management-ServiceCenter
```

Deleting a service from the command line

You can also delete a Connect-It service using the following command line:

```
csvctool -delete [name of the service]
```

For futher information on command lines, refer to the chapter Connect-It from the command line [page 193].

Creating a schedule

Scheduling allows you to set the times at which connectors produce documents. A scenario is not complete without a schedule.

Scheduling consists of associating produced document-types with schedulers.

A scheduler plays the role of a timer that "wakes up" a connector:

- At regular intervals during or outside a given period of time.
- At a specified time (March 6, 2003, for example).

Rules can be used to modify the behavior of a scheduler. For example: A scheduler wakes up a connector every day. A rule can be applied changing the scheduler for a given day.

Example :

The Desktop Discovery connector of the Desktop Discovery to Asset Management scenario produces **inventory** document types every day from 9 A.M. to 10 P.M. at intervals of five minutes. Outside of this period, the Desktop Discovery connector produces documents every hour. You can add a rule to change these parameters for the days of your choice.

Editing a scheduler

To access the scheduler editor, select **Scenario/Schedulers**.
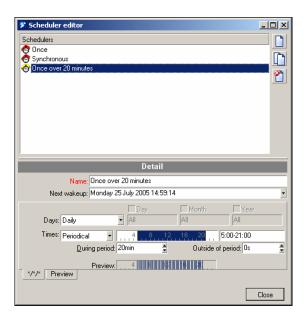
Two non modifiable schedulers are provided:

- The **Once** scheduler

This scheduler enables you to start the production of a document one time. When the produced document type is created, it is associated with a connector by default.

- The **Synchronous** scheduler

    This scheduler enables you to start producing documents every second from 00h00-24h00.

Figure 4.21. The text editor

Icon	Function
	Launches the creation of a new scheduler.
	Duplicates the selected scheduler.
	Deletes the selected scheduler.

Days field

This field specifies the days on which a document type is produced.

You have two options for this field:

- **Daily**

    Every day of the year, without exception.

- **Day of the year**

A day or selection of days, defined by selecting the Day, Month, and Year fields. For example: January 6, 2002.

The first, second, third, fourth, next to last, last day

Define a day of the week using the Day field. This applies on a regular basis or is restricted by the following Month and/or Year fields.

For example: the first Friday of each month.

Times field

You have two options for this field:

- Periodical

    By selecting this option, you set a time during the day for which you define a period in the **During period** field.

    For example: every five minutes from 4 A.M. to 10 P.M.

    You can also define a period outside of this one in the **Outside of period** field.

    To define a time range, enter a value in the text field. You must use the following syntax:

    ```
    <Start> -
    <End>, <Start> - <End>, etc.
    ```

    The date format depends on your computer's regional settings (via the Windows Control Panel). If the option parameter [AM | PM] is not given, the value is considered as being expressed using the 24-hour clock. For example: If you enter "18", 6:00 P.M. is displayed automatically after validation.

    📝 Note:

    The graphical editor is only accurate to the nearest half hour. You can adjust these values manually to the nearest minute.

    The frequency in hours (h), minutes (min) and seconds (s), is defined in the **During period** and **Outside of period** fields. For example:

    ```
    5min 30s
    1h 20min 10s
    30s
    ```

- Itemized list

    In this case, you enter a list of times separated by semi-colons. This allows you to wake up the connector on a selective basis:

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

Creating a scheduler (example)

This section describes how to create a scheduler for a wake-up event:
- Every day of the week
- Every 10 minutes from 8 A.M. to 10 P.M. and then every hour outside this period

  You add the following rule:
  - The first Sunday of the month
  - Every 20 minutes from 6 A.M. to 10 P.M. and then every 30 minutes outside of this period.

To create a schedule, click 🗋.

Populate the fields in the tab displayed below the scheduler.

Daily

In the **Days** field, select **Daily**.

Every 10 minutes from 8 A.M. to 10 P.M. and then every hour outside this period

In the **Times** field, select **Periodic**.

To select the 8 A.M. to 10 P.M. time range, you can:
- Enter it directly in the text field (8:00 A.M.-10:00 P.M. or 8-22).
- Or enter it using the graphical editor.

Next, populate the **During period** and **Outside of period** fields using the values "10 min" and "1 h". These two fields govern the wake-up frequencies during and outside the selected period.

If you have followed the previous instructions correctly, your tab should look like this:

This tab is named automatically using the information from the Days field.

Click **Create** to validate your scheduler.

Creating a rule

To create a rule, right-click one of the tabs in the scheduler editor, and then select **Add a rule**.

A new tab is displayed. Populate it according to the rule to be applied to this scheduler.

In this example, this rule's tab must resemble the following image:

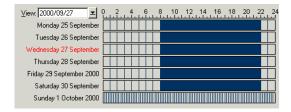Click **Create** to validate your rule.

Deleting a rule

To delete a rule, right-click the tab corresponding to this rule and then select **Delete rule**.

Preview

The **Preview** tab shows the scheduler for the current week.

By changing the day shown in the **View** field, a preview is shown for the week starting on this day.

Modifying a scheduler

To modify a scheduler:

1 Select a scheduler from the list.
2 Modify the required parameters in the tabs.
3 Validate your modifications by clicking **Modify**.

Deleting a scheduler

To delete a scheduler:

▪ Select a scheduler from the list.
▪ Press **Delete** or click 🗑.

Editing the scheduling

To edit the scheduling, select **Scenario/Scheduling**.
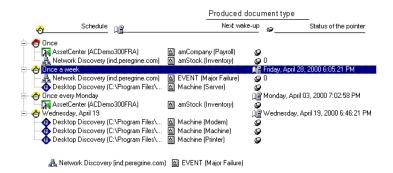
By default, the produced document-types are linked to the *Once* scheduler.

To apply a scheduler to a document type:

- Drag a product type underneath the scheduler.

  or

- Select a produced document-type and use the arrow buttons on the right to move it to the required position.

Your schedulers are shown in the edit window, sorted in alphabetical order. You associate them with document types in the scenario.

Figure 4.22. Schedule edit window

The order of produced document-types in the schedule

Moving a produced document-type to another scheduler	
⇑	Moves the document type to the previous scheduler.
⇓	Moves the document type to the next scheduler.
Moving a document type in the list of the same scheduler	
▲	Moves the document type up a row.
▼	Moves the document type down a row.

The way in which data is processed depends on two factors:

- The order of produced document-types in the schedule.
- The order of mappings within the same produced document-type.

The order in which your document types are arranged determines the order in which they are produced.

The order of the mappings in the **Mappings** tab of the main window of Connect-It is thus important: Connect-It processes the mappings for a given source document-type in the same order you specified here.

To arrange the mappings:

- Select a mapping box in the Scenario diagram pane.
- Select the **Mappings** tab.
- Select a produced document-type.
- Change the order of the mappings by moving them inside the list.

Example: For a given scenario, the **Machine** [Machine] document type has six mappings.

Figure 4.23. Mapping order

Icon	Function
	Launches the creation of a mapping.
	Enables you to edit the selected mapping.
	Deletes the selected mapping.
	Moves the selected mapping up one row.
	Moves the selected mapping down one row.

When a scenario is launched, the priorities of Connect-It for a given scheduler are as follows:

- First, the order of the produced document-types.
- Second, the order of the mappings for the same produced document-type.

Status of the pointers

In most cases, the pointer status refers to a specific time. The information provided by the pointer enables you to place the data processing of an external application in a time-based context.

The type of pointer employed depends on the external application:

- Asset Management connector

  For each mapping used in your scenario, the pointer corresponds to the last date on which the mapping was started. All records whose last modification date (**dtLastModif** field) is more recent are processed.

> **Note:**
>
> Connect-It automatically adjusts the time difference between the computer on which it is installed and the AssetCenter server.

- ServiceCenter connector

  This connector's pointer changes according to the nature of the data coming from the source application:

  - For events, the pointer corresponds to their sequence number (**evsysseq**).

    For each mapping used in your scenario, the pointer corresponds to the sequence number of the last event processed the last time the mapping was launched. The next time it is launched, all events with a sequence number greater than this value are processed.

  - For tables, the pointer corresponds to the date that the mapping was last launched. The next time it is launched, all records whose last modification date (**sysmodtime** field) is more recent than the pointer are processed.

    > **Note:**
    >
    > According to the configuration of your ServiceCenter connector, the time difference between the computer on which Connect-It is installed and the ServiceCenter server is:
    > - Either automatically adjusted.
    > - Or adjusted according to the offset entered by the user.

- Database connector

  The pointer used is that defined by the user.

- NT Security connector

  Each time the scheduler is woken up, the connector recovers all the information from the domain to explore and not only the information concerning the changes.

  > **Note:**
  >
  > No schedule pointer is shown when you create a new scheduler and assign it an NT domain.

- Delimited text, XML connectors

  The pointer used is the most recent creation date of the files already processed.

- E-mail (fetching) connector

  The pointer used is the date received of the e-mail.

- Network Discovery connector

  The pointer used is the event number (for the EVENT document type).

By double-clicking the pointer status line, you can force another date for the pointer status of an `.fsf` file or another event number for a Network Discovery event.

Example:

Several tens of thousands of events are memorized by Network Discovery. By changing the pointer status (which is 0 before you start the scenario for the first time) to 20,000, only events after 20,000 are processed by the scenario.

Refreshing the pointers in the scheduling window

If the scheduling window is open when a scenario is launched, the status of the pointers is not refreshed. In order for the statuses of the pointers to be refreshed in the scheduling window, you must close this window and then reopen it.

Starting the scenario

To start a scenario:

- Select **Start all schedulers** from the **Scenario** menu.

  or

- Click ⏩.

Stopping the scenario

To stop a scenario:
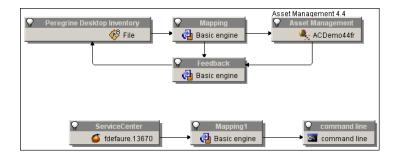
- Choose **Stop** from the **Scenario** menu.
- Or click ⏸.

Using schedulers from the command line

In this example:

- the scenario used is `pdiac.scn`, located in the `pdi` scenario folder
- the scenario is modified and includes a ServiceCenter scenario
- several document types are processed
- the scenario is executed once
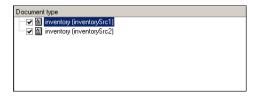
■ the and -once and -scheduler commands are used

-once command

The order of production of the documents corresponds to the order of definition of the produced document types and their mapping to a connector (visible, for a connector selected in the **Document type** tab).

For example, for a source connector producing two document types *Inventory(inventorySrc1)* and *Inventory(inventorySrc2)*, the document *Inventory(inventorySrc1)* will be produced first.

The command line is the following:

```
conitsvc -once 'D:\Program Files\HP OpenView\scenario\pdiac.scn'
```

If multiple connectors produce multiple document types, the order of production will be the order of creation of the connectors.

Note:

It is recommended against using the -once command if multiple connectors in the same scenario produce documents.

The order of production of the document types defined in the schedulers is ignored. The -once command, used alone, corresponds to the ▶ command in the GUI.
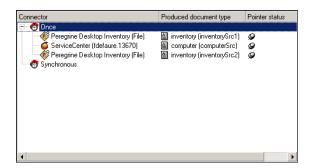
When the path for a scenario is defined as a parameter, the -once (or -scheduler) option must be defined as a parameter. Otherwise, Connect-It exits after the scenario is loaded.

The -once and -scheduler options must not be applied at the same time.

-scheduler command

The order of production of the document types is that defined by the **Once** scheduler. The -scheduler command corresponds to the ⮆ command in the GUI.

For example, the document types will be processed in the following order: inventorySrc1 (PDI connector), computer (ServiceCenter connector), inventorySrc2 (PDI connector). When more than one connector producing documents exists, only the -scheduler option controls the production order of the document types of the different connectors. The scheduling pointers are used and updated after the scenario executes.

The command line is the following:

```
conitsvc -once -scheduler 'D:\Program Files\HP OpenView\scenario\pdiac.scn
```

When the command line is executed, ony the records modified or inserted since the last time are recovered according to the parameters defined for the schedulers.

✎ Note:

The scheduling option makes user of the scheduling pointers. It is possible to disable use of the scheduling pointers in the connection configuration wizard in order to obtain the same behavior as the -once option used alone, the difference being the order or production of the documents.

When the -scheduler option is used as a parameter, the values of the schedule pointers are used and updated after executing a session. There are two different behaviors according to the scheduler definitions:

- If the document types are associated with the Once scheduler, then Connect-It exits after the scenario is executed. This is the same behavior as the -once option and it takes schedule pointers into account.
- If the document types are associated with a scheduler, then Connect-It never exits and wakes up at the frequency defined by the schedulers.

Saving a scenario

To save a scenario

1 Perform one of the following actions:
- Select the **File/Save** menu.
- Click 💾.

To save a scenario under a different name:
1 Select the **File/ Save as** menu.

Backup copy of a saved scenario

When you save a scenario, a backup copy of it is automatically created in the scenario folder. This copy has the file extension .bak.

To deactivate this option:
1 Select **Edit/ Options**.
2 Unfold the **Confirmation** node.
3 Select *No* for the **Keep a backup copy of the scenario** option.
4 Click **OK**.

Moving a scenario

When you move a scenario to another folder do not forget to:
- Either move the files associated with folder.
- Or change the relative paths contained in the sceanrio scripts.

Verify the paths of the files: .bas, .str, .usr, .scp and .mpt.

Scenario documentation

The Scenario builder enables you to view or create documentation containing different information about the current scenario.

The information in the scenario documentation is always up-to-date because it is extracted dynamically from the scenario's .scn file.

Contents of the scenario documentation

A scenario documentation contains the following sections:

- General information

  This section provides the following information about the scenario:

  - Name

    The name of the scenario is preceded by the full path name on your computer.

  - Date of file's last modification.

  - Version and build number of the Connect-It application with which the scenario was last modified.

  - All the information entered by the user in the last scenario history. (**File/Scenario history** menu).

- Information about scenario diagrams

  This section provides the following information about the scenario:

  - Information about views

    The mappings associated with each view are listed.

  - Diagrams associated with views

    The preview of the views is in a .bmp image.

- Connector configuration

  This section provides the configuration parameters of the connectors in the scenario. For more information about connector configuration, refer to the *Connectors* guide, chapter *Configuring connectors*.

- List of WHERE and ORDERBY clauses by document type

  This section deals with the list of WHERE and ORDERBY clauses by document type. These clauses are part of the production directives of the scenario's source connectors.

- Document types defined by the user

  This section presents the elements of each document type defined by the user.

  For each document type, the following information is available:

- Name of the document type.
- Name of each element.
- Nature of the element:
  - field (ATTRIBUTE)
  - structure (STRUCT)
  - collection (ARRAY)
- The data type of each field: string, short, double, timestamp, etc.

For more information about creating document types, refer to the chapter Implementing an integration scenario [page 55], section Define the produced or consumed document-types [page 74].

- Mappings defined by the user

  This section details the mappings between the document types defined by the user.

  For each mapping, the following information is available:
  - The name of the mapping.

    After the name of the mapping, the names of the source and destination document types published by the connectors will be indicated in parentheses.
  - Name of the source-document type.
  - Name of the destination-document type.
  - For each mapping node, the:
    - Destination element
    - Mapping script
    - Source element
    - Possible mapping description

  For more information about how to create mappings, refer to the chapter Define the mapping of document types [page 91].

- Schedulers and their associated files

  This section presents the:
  - Schedulers of the scenario.

    For each scheduler, we provide its:
    - Name
    - Periodicity
    - Connectors and associated document types
  - Maptable files
  - Basic script files
  - Character string files

- Scenario modification history

This section provides information about each modification that can be made to each scenario. This information corresponds to the data that the scenario author enters in the **Scenario history** window (**File/Scenario history** menu).

Viewing a scenario documentation in HTML format

The Scenario builder enables you to display the documentation of the current scenario in an Internet browser (in HTML format).

Viewing a scenario documentation in HTML format

1   Open or create the scenario with the Scenario builder.
2   Select the **File/View scenario documentation** menu.
3   Wait for your Internet browser to open and display the scenario documentation.

📝 Note:

To use the **View scenario documentation** function, verify that your operating system associates this format (HTML) to an Internet browser. Example: Microsoft Internet Explorer or Netscape Navigator.

Creating a scenario documentation

The Scenario builder enables you to create scenario documentations in the following formats:
- HTML format
- DBK format

Creating a scenario documentation

1   Open or create a scenario using the Scenario builder.
2   Select the **File/Create a documentation** menu.
3   Name the documentation file.
4   Select the documentation format in the **Type** field.
5   Click **Save**.

📝 Note:

By default, the scenario documentation is saved in the same folder as your scenario's SCN file.

Documentation in DBK format

Creating a scenario documentation in DBK format creates an XML file that conforms to the DocBook DTD.

For more information about the DocBook DTD, consult the following Web site: http://www.docbook.org

Properties of the scenario documentations

The creation of scenario documentations uses the following elements:

- An XSL processor

  The processor provided with Connect-It is the Xalan processor.

  > ⚠ Warning:
  >
  > The Xalan processor is not compatible with Java RunTime Environment 1.1. Java RunTime Environment or Java Development Kit version 1.2 or higher must be installed.

- A CSS style sheet used for viewing HTML format.
- XSL style sheets enables you to convert SCN files to DBK format.
  - SCN files in DBK format
  - DBK files in HTML format

The conversion of SCN files to DBK format goes through an intermediate conversion of a SCN file to an XML format.

The conversion of an SCN file to HTML format goes through the intermediate conversions to an XML format, then to an HTML format.

Format of the scenario documentation	Conversion order	
DBK	1	*SCN*
	2	XML
	3	*DBK*
HTML	1	*SCN*
	2	XML
	3	DBK
	4	*HTML*

Editing the scenario documentation properties

When you edit scenario documentations, you need to use a wizard, which you launch by selecting the **File/ Documentation properties** menu.

Configure the XSL processor

This wizard page enables you to configure the XSL processor used to convert the SCN files to DBK and HTML format.

The Xalan processor is the default processor provided with Connect-It.

Its file path is: `[Connect-It installation folder]\datakit\doctrans\xalan\xalan.bat`.

You can use other XSL processors if you want. Example: The MSXSL processor can be downloaded for free from different Microsoft Web sites.

XSL processor executable file

In this field, specify the full path of an XSL processor. By default, the path of this file is: `[Connect-It installation file]/datakit/doctrans/xalan/xalan.bat`.

XSL processor parameters

The default parameters of this field are:

- %IN

  This variable corresponds to the full path of the SCN file to convert.

- %XSL

  This variable corresponds to the full path of the XSL file used for the DBK or HTML conversion.

- %OUT

  This variable corresponds to the full path of the output DBK or HTML file.

Use a CSS style sheet to preview in HTML

Select this option to view the scenario documentation in HTML format that uses a CSS style sheet.

The path to the CSS style sheet used by default is: `[Connect-It installation folder]\datakit\doctrans\css\default.css`.

Predefined XSL style sheets

This page indicates the style sheets used for the conversion:

- From an XML file to DBK format.
- From a DBK file to HTML format.

You can change the value of the **Description** and **Alias or XSL style sheet** columns by clicking directly on these values.

The alias of a style sheet is the value entered in the **Extension** column. By indicating an alias before the full path of another style sheet, separated by a

semi-colon, you are indicating the first style sheet that your XSL processor should use.

For the HTM extension, the value `DBK;C:\Program Files\HP OpenView\Connect-It\datakit\doctrans\xsl\dbk2htm.xsl` indicates that the XSL processor converts the XML format of the scenario to DBK format, then to HTM format.

XSL style sheets defined by the user

This page enables you to indicate new XSL style sheets, which enable you to convert a file from one format to another.

Indicating an XSL style sheet defined by a user

1 Click ⚑.
2 Enter a file extension name in the **Extension** column.
3 Click on the same line in the **Description** column and enter a description.
4 Click on the same line in the **Alias or XSL style sheet** column and enter the path of your XSL style sheet.

By indicating the alias of other XSL style sheets, you can specify as many intermediate conversions between the starting format and the final format as you want.

5 Managing an integration scenario

Opening an existing scenario

The scenario editor enables you to open files with the extension `.scn`.

To open an existing scenario

1 Perform one of the following actions:
  - Launch the Scenario Builder and select the **File/ Open** menu.
  - Select the `.scn` file in a browser and double-clik this file.

    The Scenario Builder will open the scenario.
  - Select the `.scn` file in a browser and drag and drop it into the Scenario Builder's main window.

    📝 Note:

    If a scenario is already open, a dialog box appears asking if you want to save this scenario before opening a new one.

Option associated with opening a connector

The **Open all the current scenario's connectors when starting a connector** option enables you to automatically open all the connectors in a

scenario when you start one single connector. This is very useful for scenarios using numerous connectors.

Deploying a scenario (in Unix)

To deploy an existing scenario in Unix, do the following:

- In Windows:
  1 open the scenario and perform the necessary configurations and modifications
  2 create a property file using the *-df* command
  3 export the scenario configuration to the property file using the *-export* command

     For example, for the scenario called `myScenario`, the export command will be the following:

     ```
     conitsvc -export myScenario.scn
     ```
  4 edit the `.prop` property file and modify it for Unix

- In Unix:

  Do not modify the scenario but instead using the following commands to change the behavior:

  - To test the scenario:

    ```
    conitsvc -df:<scenario name>.prop [<scenario properties option>] <name of the scenario>.scn
    ```

    ### Note:

    When the path for a scenario is defined as a parameter, the -once (or -scheduler) option must be defined as a parameter. Otherwise, Connect-It exits after the scenario is loaded.

  - To launch the Connect-It service:

    ```
    conitsvc -svc <name of the service>
    ```
  - To create a Connect-It service:

    ```
    csvctool -create -scn:<name of the scenario>.scn svclog:<name of the scenario>.log -options:"-df:<name of the scenario>.prop" <name of the service>
    ```
  - To start the Unix daemon, execute the following command line:

    ```
    conitsvc -svc indsc &
    ```

For further information on command lines, refer to this chapter, section

Connect-It from the command line

There are two ways to execute Connect-It: Using the graphical interface or from
the command line.

This section presents the different options and the syntax used to execute
certain tasks from the command line, such as opening a connector or putting
a scenario into production.

To start Connect-It from the command line:

1 Open a command prompt
2 Go to the bin folder in the Connect-It installation folder

Commands used

The following commands are used:

- conitsvc: Connect-It in non-graphical mode. This application is executed by
  a service created from the service console.
- conitgui: Connect-It in graphical mode
- csvctool: Connect-It service management utility

Syntax used

- The non-graphical version of Connect-It has the following command-line
  parameters:

```
conitsvc [-verbose] [-?|h|H]
[-log:<Log file name>]
[-d:<connector.property=value>]
[-dc:<connector.property=value>]*
[-dm:<(section|monitor).property=value>]*
[-df:<Properties file>]*
[-once] [-scheduler]
[-port:<Port>] [-activate:<Document types to activate>]
[-updatescn:<Name of the updated scenario file>]
[-gendoc:<Name of the documentation file to generate>] [-openconnectors
] [-svc] [-showstatus] [-sleep:<Length of pause>]
[-export:<Name of the property file>] <scn|svc>
[-forcecnx:<Name of the connection>]
```

- The graphical version of Connect-It has the following command-line
  parameters:

```
conitgui [-verbose] [-?|h|H]
[-log:<Log file name>]
```

```
[-d:<connector.property=value>]
[-dc:<connector.property=value>]*
[-dm:<(section|monitor).property=value>]* [-df:<Property file>]* [-once
] [-scheduler]
[-port:<Port>] [-activate:<Document types to activate>]
[-updatescn:<Name of the updated scenario file>]
[-gendoc:<Name of the documentation file to generate>] [-openconnectors
] [-configscheduler] [-configconnectors]
[-export:<Name of the property file>]   <scn>
```

■ The Service Console has the following parameters:

```
csvctool-<option>[<parameter>:<value of the parameter>]*[-y]<svc>
```

-y: This option disables confirmation of modification

conitsvc.ini

This file contains all services declared for the Connect-It Windows or Unix version.

This file is located in:

■ version prior to version 3.7 (Windows):

```
C:\Document and settings\All users\Application Data\Peregrine)
```

■ version prior to version 3.7 (Unix):

```
~/Peregrine
```

■ version 3.7 and later (Windows):

```
C:\Document and settings\All users\Application Data\HP OpenView\conf
```

■ version 3.7 and later (Unix):

```
~/.ov/conf
```

Command options used

Each command can be specified with one or more options.

The following table describes the command options for the **conitsvc** and **conitgui**.

Option	Value	con-it-gui	con-its-vc	Description
once		Yes	Yes	Start all connectors once only.
				This option must not be used at the same time as the **-scheduler** option.
				Refer to chapter Creating a schedule [page 173] for information on how to use scheduling pointers from the command line.
scheduler		Yes	Yes	Start the schedulers.
				This option must not be used at the same time as the **-once** option.
				Refer to chapter Creating a schedule [page 173] for information on how to use schedulers from the command line.
opencon-nectors		Yes	Yes	Open all connectors.
				In graphical mode, a window containing the messages concerning the actions performed is opened.
forcecnx		No	Yes	Force the automatic reconnection when opening connectors.
				By default, a connection failure when opening a scenario causes Connect-It to stop. If you select this option, the automatic reconnection process is applied even when opening the scenario.
verbose		Yes	Yes	Enables messages and warnings.
gendoc	Name of the documentation file (optional)	Yes	Yes	Generates the scenario documentation.
				If no documentation file is specified, a documentation DBK is generated at the same location as the scenario and includes the same name.
svc		No	Yes	Specifies a Connect-It service to use instead of using the scenario.
				A Connect-It service is a combination (name/ scenario/ log file/ options) defined using the csvctool program or the Service Console.
				Refer to the section Examples of commands [page 199] for an example of a command using the **svc** option.
configsched-uler		Yes	No	Starts the configuration of the schedulers.
configcon-nectors		Yes	No	Starts the configuration of the connectors.

Option	Value	con-it-gui	con-its-vc	Description
port	Port number	Yes	Yes	Assigns a port number for the communication socket between conitgui or conitsvc and cntrsrv. If no port is specified, a number is assigned automatically.
sleep	Value in seconds	No	Yes	Length of time the activity is paused in seconds before production.
log	Log-file name	Yes	Yes	Specifies the name of the log file.
activate	List of document types to activate, separated by commas.	Yes	Yes	List of document types to activate before launching a scenario. Used with the **-updatescn** option, saves the activation in the scenario.
d	Deprecated option	Yes	Yes	This option is maintained for compatibility reasons. This option has been replaced by the -dc option.
dc	<connector name>.<connector property>=<property value>	Yes	Yes	Assigns a value to the connector configuration property. The declared value is valid for the current scenario but is not saved. To save the modifications, use the **-updatescn** option.
dm	<section | monitor>.<parameter>=<value>	Yes	Yes	Assigns a value to a monitor configuration property. The declared value is valid for the current scenario but is not saved. To save the modifications, use the **-updatescn** option.
updatescn	Name of the updated scenario file (optional)	Yes	Yes	Upates the scenario with all the declared values. If no backup file is specified, the new scenario replaces the previous one.
configure	Name of the updated scenario file (optional)	Yes	Yes	Exports the properties and updates the scenario with the exported values.
df	Property file	Yes	Yes	Specifies a property file to update the connector and monitor configurations. The declared values are valid for the current scenario but are not saved. To save the modifications, use the **-updatescn** option.
showstatus		No	Yes	Displays the processing status of the documents.

Option	Value	con-it-gui	con-its-vc	Description
export	Name of the property file (optional)	Yes	Yes	Exports the scenario configuration to a property file (called by the **-df** command option).
				If no property file is specified, the name used is that of the scenario, to which the extension `.prop` is added.

The following table describes the command options for **csvctool**.

Option	Unix	Request for confirmation	Description
list	Yes		Lists the Connect-It services.
info	Yes		Gets information concerning a service: Name, scenario, options, warnings and messages file, service configured or not (Windows only), service installed or not (Windows only).
create	Yes		Creates a new service.
update	Yes	Yes	Updates the service.
delete	Yes	Yes	Deletes a service.
install			Installs a Windows service.
uninstall		Yes	Uninstalls a Windows service.

Paramètre	Value	Unix	Option	Description
name	New name for the service	Yes	update	Name of the service
scn	Scenario name	Yes	update / create	Scenario associated with the service
svclog	Log-file name	Yes	update / create	Log file associated with the service
c			update / create	Service defined as configured
options	Additional service parameters	Yes	update / create	Additional service parameters
account	User account		update / create	User account in which the Windows service is started
password	User password		update / create	User account password
log	Name of the csvctool log file	Yes		Log file

Property file

The configuration parameters of a scenario can be exported to a property file with the `.prop` file extension.

This file is then called by the -df command-line option.

The configuration properties are:

- sorted by connector and monitor
- preceded by a brief description
- commented

Changing the configuration of a scenario consists of changing the value of a property.

Syntax of a property file

The property names follow the following syntax:

```
<type>.<object>.<name>
```

- The type corresponds to the value to be updated. The possible values are **Monitor** and **Connector**.
- The object corresponds to the name of the object whose property is to be updated
- The name corresponds to the name of the property to be updated

To update a configuration property for a connector, the syntax is as follows:

```
Connector.<name of the connector>.<name of the configuration property> = <
value of the configuration property>
```

For example, to change the name of the connector called "My AC connector", the command line is as follows:

```
Connector.My AC connector.Server = MyProductionServer
```

To update a configuration property for a monitor, the syntax is as follows:

```
Monitor.<name of the monitor>.<name of the parameter> = <value of the para
meter>
```

For example, for the name of the SMTP server:

```
Monitor.MailMonitor.SMTPServer = smtp.myserver.com
```

Example of a property file

A property file is structured as follows:

```
# Comment
! Comment
# Blank lines are ignored

# Connector configuration properties are defined using the following synta
x:
# Connector.<name of the connector>.<name of the property>
# The ':' and '=' characters are used to separate the property names and v
alues.
Connector.MyCntr.Property = Value
Connector.MyCntr.Property2 : Value
Connector.MyCntr.Property3= Long value split over \
```

```
multiple lines
Connector.MyCntr.Property4= new lines and tabulations may be inserted \
using the \t and \n parameters

# If the name of the connector contains the '=' or ':' character, the they
are escaped as '\=' and '\:'
# Example for the 'My cntr := test' connector
Connector.My cntr \:\= test.Property=value

# If the name of a character contains a period, an addional period must be
added
# Example for the 'My.cntr' connector
Connector.My..cntr.Property=Value

# Monitors are configured using the following syntax:
# Monitors.<section or monitor name>.<property name>
Monitor.ApplicationLogs.Capture=false
Monitor.MailMonitor.TextPassword = test
```

Examples of commands

This section presents several examples of command lines.

- -dc

  Initialize the ODBC connection of the database connector called **test**

  ```
  conitsvc.exe -dc:test.Odbc_Dsn=AC43fr database_sc.scn
  ```

- -dc

  Using the password unencrypted

  ```
  conitsvc -dc:assetmanagement.textpassword=b42var9
  ```

  ✎ Note:

  If you use the 'password' command option, the password to populate is encrypted.

- -openconnectors

  Open all connectors

  ```
  conitsvc -openconnectors mail_sc.scn
  ```

- -forcecnx

  Force the automatic reconnection to a service

  ```
  conitsvc -svc -forcecnx myService
  ```

- -dm

  Disable Mail monitor

  ```
  conitsvc -dm:MailMonitor.Active=false ldap_sc.scn
  ```

- -dm

Increase the document log size

```
conitgui -dm:MemoryMonitor.MaximumTraceItems=2000 ldap_sc.scn
```

- Start as service

```
conitsvc -svc testqa
```

✎ Note:

The name of the service is that which is saved in the `conitsvc.ini` file (located, in Windows, in the `c:\Documents and Settings\<user account>\Application Data file`)

- Specification of the document type to produce

```
conitsvc -activate:amAssetSrc,amEmplDept test.scn
```

- Using a property file for the scenario

```
conitsvc -once -df:myPropFile.prop test.scn
```

Using Connect-It Web

The Web console is used to manage a set of Connect-It services centrally.

Service and scheduler definitions and monitoring information are stored in a unique database.

How Connect-It Web works

To access the web interface:

1 Start the Tomcat server.
2 Start the service on the Tomcat server.
3 Open an Internet browser and enter the following address:

```
http://<server address>:<server port number>/<name of the .war file>
```

For example:

```
http://localhost:8080/ovcit
```

▶ Install Connect-It Web [page 20].

Login

A login and password are required to use Connect-It Web.

▶ Install Connect-It Web [page 20].

Scenario location

Scenarios can be viewed and executed via the web interface depending on where they are physically located.

The scenarios must be placed in the file system that is accessible by Tomcat web server.

Defining a service

A service is a scenario instance.

Each scenario instance is defined by:

- An identifier or a name
- A scenario file associated with it

When accessing Connect-It Web, the list of defined scenarios is visible on the right-hand side of the page.

To access detailed information about a service, click on the service in the list. Information is displayed in the **Information about the service** frame.

Define a new service

1 On the Connect-It page, click the **Services** menu.
2 Click **New**.
3 Populate the **Service name** field.

   The name of the service must be unique.
4 Provide a description of the service. This field is optional.
5 Populate the path of the scenario file.

   The path must be a local path for the server.

   > Note:
   >
   > Paths are case-sensitive in Unix and Linux.

6 Click **Validate**.

   The newly created service appears in the list of services.

Delete a service

To delete a service, select it in the list via the checkbox and click **Delete**.

You can delete several services at a time.

Configure a service

Connect-It Web lets you define parameters to change the configuration of a scenario's connectors.

Defining parameters is optional.

These parameters will be used in place of the parameters that are defined in the scenario.

Parameters can be accessed on the service's **Parameters** tab in the **Information about the service** frame.

To add parameters:

1   Select a service.
2   Click the **Parameters** tab.
3   Click **New**.
4   Populate:
    ■   The name of the connector
    ■   The name of the parameter
    ■   The value of the parameter

    For example:
    ■   Connector name: AssetManagement
    ■   Parameter name: Server
    ■   Parameter value: ACProd

5   Click **Validate**
6   Repeat this operation as many times as required.

 Tip:

To find out which parameters are used in your scenario, export the parameters using the **conitsvc** command's **-export** option.

For example, to export the parameters of the *catalog* scenario:

```
..\..\..\bin\conitsvc.exe -export catalog.scn
```

The properties file that is created is located in the same folder as the scenario.

To delete parameters:

1   Select the parameters to delete using the checkbox.
2   Click **Delete**.

Scheduling a service

Connect-It Web gives you complete control over how services are scheduled.

Scheduling features are based on schedulers that are separate from scenarios.

A scheduler can be associated with several services.

To access the list of schedulers, select the **Schedulers** menu on the navigator.

How a scheduler works

A scheduler:
- Defines when to execute one or more services with which it is associated
- Defines one or more time ranges for the execution frequency
- Defines an execution frequency outside of a time range

To ensure the correct operation of the scenario that is associated with the service, the document types which are produced by the connectors must be configured as follows:
- Produced document types must be associated with a **Once** scheduler in the Connect-It scenario builder.
- Produced document types must be placed in the order that you want them to be produced.

  ▶ Editing the scheduling [page 178].

📝 Note:

The date used is the date of the server.

Defining a scheduler

To define a scheduler:
1 On the Connect-It page, click the **Schedulers** menu.
2 Click **New**.
3 Populate the **Scheduler name** fields.
4 Provide a description of the scheduler. This field is optional.
5 Define the date of the scheduler's next wakeup.

   The date of the next wakeup is updated after the scheduler has executed.
6 Populate the scheduler's behavior:
   - Occurrence: Used to define the scheduler's execution frequency (daily, once per week, etc.).
   - Time ranges:
     1 Populate the time range fields (**Start** and **End** fields).

        The level of granularity of time ranges is one hour.
     2 Click **Add**.
   - Populate the **Every** field to define the intervals at which the service associated with the scheduler is to be executed.

     For example, for a scenario that uses the E-Mail connector, you can define a frequency of one second. For an inventory scenario, you can define an hourly frequency (3600 seconds) that executes between 7 and 7:59 am.

If necessary, define when to execute the service outside of a given time range.

■ Repeat this operation as many times as required and then click **Validate**.

Deleting a scheduler

To delete a scheduler:

1 Select the scheduler using the checkbox.
2 Click **Delete**.
3 Click **OK**.

Note:

When a scheduler is deleted, services are no longer associated with it.

Associate a scheduler with a service

You can associate a scheduler:

■ From a service via the **Scheduling** tab:
  1 Select the service
  2 On the **Scheduling** tab, click **Configure**
  3 Select the scheduler to assign from the drop-down list.
  4 Click **Validate**
■ From a scheduler via the **Assigned services** tab:
  1 Select the scheduler.
  2 On the **Associated services** tab, click **Modify**.
  3 Select the service(s) to assign from the list of services.
  4 Click **Validate**.

Internal scheduling

The console's schedulers replace the scheduling information defined in the scenarios.

However, it is possible to maintain the scenario's original programming information. To do this:

■ Select the service.
■ On the **Scheduling** tab, click **Configure**.
■ Select **Internal scheduling**.
■ Click **Validate**.

> ✎ Note:
>
> These two scheduling modes are exclusive.

Event-driven connectors

Starting in Connect-It 3.70, scenarios whose source connector is an event-driven connector, such as the XML listening connector, no longer need to be scheduled. The scenario is automatically triggered when an event is received.

For this type of scenario, you do not need to assign a scheduler to the service.

Starting a service

To start or stop a service, select it via the checkbox and then click **Start** or **Stop**.

When a service is started, the associated scenario is opened, and:

- In console scheduling mode, the scenario waits for an event from the associated scheduler.
- In internal scheduling mode, the internal scheduling starts.

In console scheduling mode, the associated scheduler must also be started. The service waits as long as the scheduler has not started.

To start or stop a scheduler, select it via the checkbox and then click **Start** or **Stop**.

> ✎ Note:
>
> Starting and stopping services is independent from starting and stopping schedulers.

Monitoring a service

Connect-It Web lets you view the activities linked with the execution of a service:

- log
- sessions
- statistics

Configuring service monitoring

Connect-It Web lets you configure the desired monitoring level for each service:

1 Select the service.
2 On the **Monitoring** tab, click **Configure**.
3 Select the appropriate options:

- Save service execution log: Select this option to save log information (debug, information, warning, error) in the Connect-It Web database.
- Session information: This option lets you display the start and end dates of each session.
- Save execution statistics: This option lets you save the execution statistics of the scenario's session. The behavior is the same as the one described for the Connect-It document log.

  ▶ The Connect-It log [page 213].

- Limit persistence of monitoring information: This option lets you define how long information about the service is kept.

View monitoring information

The **Monitoring** tab lets you view the scenario's last session and the last 20 lines of the log.

> 🖉 Note:
>
> Modify the value of the **Page size** field to specify how lines from the log are displayed.

The contents on the **Monitoring** tab depends on how scenario items are processed.

The Monitoring tab lets you do the following:
- Refresh log information
- Configure monitoring
- Display all supervised elements (session histories and service log)
- Delete all monitoring information (session histories and service log)

Click **Display all** to view the entire log. This button lets you access the session histories and the service log.

For each of these elements, you can:
- Session history:
    - Refresh
    - Clear the history

      > 🖉 Note:
      >
      > Clearing the history does not affect the contents of the service log.

- Service log:
    - Refresh
    - Clear the log

> **Note:**
>
> Clearing the contents of the service log does not affect the session history.

Favorites

Favorites are preconfigured connectors. They are visible in the **Favorites** tab of the Scenario builder's Toolbox.

When you use a favorite in a scenario (by dragging it and dropping it there), it creates an instance of the connector associated with this favorite. You can then delete the association between this connector instance and the favorite.

When a favorite is associated to one or more connectors:

- Anytime you reconfigure a favorite, it reconfigures the connectors associated with it throughout all scenarios.

  > **Warning:**
  >
  > The name and description of a favorite are unique to that particular favorite. When you modify this name and description in the configuration wizard, the changes are not carried over to the configuration of the associated connectors.

  In order for this reconfiguration to actually take place for each connector in all scenarios, you must manually open each of these scenarios in the Scenario builder.

- Anytime you reconfigure a connector, it might been the reconfiguration of the favorite associated with it, as well as all the other connectors associated with the favorite.

  At the end of the connector's reconfiguration, a dialog box proposes you the following choices:

  - Update the associated favorite.
  - Break the association with the favorite before updating the connector.
  - Cancel the reconfiguration of the connector and the favorite.

Editing the favorites

A favorite enables you to keep a connector's configuration and to reuse it in other scenarios.

To create a favorite associated with a connector

1 Start the Scenario builder.
2 Open an existing scenario or create a new one.
3 Select the connector for which you want to create a favorite.
4 Right-click.
5 Select **Favorites/ Add to the favorites with association** from the shortcut menu.

   The favorite appears in the **Favorites** sub-tab of the **Toolbox** pane.

   📝 Note:

   When a connector is associated with a favorite, the name of the favorite appears above the connector in the Scenario diagram pane.

To create a favorite not associated with a connector

1 Start the Scenario builder.
2 Open an existing scenario or create a new one.
3 Select the connector for which you want to create a favorite.
4 Right-click.
5 Select **Favorites/ Add to the favorites without association** from the shortcut menu.

   The favorite appears in the **Favorites** tab of the **Toolbox** pane.

To use a favorite in a scenario

1 Open and create the scenario in which you want to use the favorite.
2 Move the mouse cursor over the **Favorites** tab of the **Toolbox**.
3 Select the favorite.
4 Perform one of the following actions:
   ▪ Drag and drop this favorite into the Scenario diagram pane.
   ▪ Double-click.

   An instance of the connector associated with this favorite is automatically created in the Scenario diagram.

To break the association between a connector and a favorite

1 Select a connector associated with a favorite in the scenario.

   If the connector is associated with a favorite, the name of this favorite appears in the Scenario diagram.
2 Right-click.

3 Select **Favorites/Break association with the favorite** from the shortcut menu.

As soon as the association with the association is broken, the name of the favorite that appears above the connector in the Scenario diagram disappears.

To reconfigure a connector using a favorite and creating an association

1 Select the favorite.

2 Drag it on a connector similar to the favorite.

Example: an XML favorite onto an XML connector.

3 A dialog box opens asking if you want to update the connector with the configuration of the favorite and to create an association.

4 Click **OK**.

To reconfigure a connector using a favorite without creating an association

1 Select the favorite.

2 Drag it onto a connector similar to the favorite:

- While holding down the left button of the mouse.
- By holding down the **Shift** key on your keyboard.

3 Release the mouse button.

4 A dialog box opens asking if you want to update the connector with the configuration of the favorite without creating an association.

5 Click **OK**.

To reconfigure a favorite from the Favorites sub-tab

1 Start the Scenario builder.

2 Select the **Favorites** tab from the **Toolbox** pane.

3 Select the favorite that you want to reconfigure.

4 Perform one of the following actions:

- Click on the Favorites toolbar.
- Right-click and select **Reconfigure the favorite**.
- Select **Favorites/Reconfigure the favorite**.

5 Modify the configuration of the favorite using the configure wizard that opens.

When the scenario using this favorite is reloaded, the associated connectors are also automatically reconfigured.

To reconfigure a favorite from an associated connector

1 Start the Scenario builder.

2   Open a scenario using a connector associated with the favorite that you want to reconfigure.

3   Select the associated connector.

4   Perform one of the following actions:

   ■   Right-click and select **Configure the favorite** from the shortcut menu.

   ■   Select the **Tools/ Configure** menu (**F2**).

5   Modify the configuration of the favorite using the configure wizard that opens.

6   Click **Yes** in the dialog box that appears.

   If you click **No** in this dialog box, the association with the favorite is broken before the connector is updated.

   When the scenario using this favorite is reloaded, the associated connectors are also automatically reconfigured.

✎ Note:

A favorite associated with a connector is only effectively updated once the scenario is reloaded.

1   Start the Scenario builder.

2   Select the **Favorites** tab from the **Toolbox** pane.

3   Select the favorite that you want to delete.

4   Perform one of the following actions:

   ■   Click 🖼 on the Favorites toolbar.

   ■   Right-click and select **Delete the favorite** from the shortcut menu.

   ■   Select the **Favorites/Delete the favorite** menu.

5   Wait for the favorite to disappear from the **Favorites** tab.

✎ Note:

When a favorite is deleted, the associated connectors lose this association. This loss is symbolized by the disappearance of the favorite's name above the connector in the scenario diagram.

Managing the favorites file

The parameters of the favorites (preconfigured connectors) are kept in the Favorite.fav file containing the Connect-It version number. This file is saved

in the bin folder of your Connect-It installation folder. *Example*: `C:\Program Files\Hewlett-Packard\Connect-It\bin\Favorite330.fav`

> 🖊 **Note:**
>
> The last version of the `favorite.fav` file is saved in a `favorite.bak` file. If a problem comes up during the update, use this file by renaming it.

This file can be exported or imported between two scenario builders.

To import a scenario file

1 Start the Scenario builder.
2 Put your cursor in the zone of the **Favorites** tab.
3 Perform one of the following actions:
   - Click on the Favorites toolbar.
   - Right-click and select **Import favorites** from the shortcut menu.
4 Indicate the path of the `Favorite.fav` file containing the favorites that you want to import into the Scenario builder.

> 🖊 **Note:**
>
> You can perform this operation manually in a Windows browsers by copying and pasting the `Favorite.fav` file in the `bin` folder of the Connect-It installation folder.

To export a scenario file

1 Start the Scenario builder.
2 Perform one of the following actions:
   - Right-click the **Favorites** tab and then select **Export favorites** from the shortcut menu.
   - Click on the Favorites toolbar.
   - You can select multiple favorites by holding down the CTRL key while selecting them with the mouse.
3 Specify the path of the folder in which you want to save the `Favorite.fav` file containing favorites of the scenario builder you are using.

> 🖊 **Note:**
>
> You can perform this operation manually in a Windows browser by copying and pasting the `Favorite.fav` file in the folder of your choice.

Update between favorite and associated connectors

When you open a scenario, the synchronization between a favorite and its associated connectors depends on the date they were updated.

The favorite or the connector having the most recent update imposes its configuration on the others.

Importing favorites from previous versions of Connect-It

The favorites are linked to the version of Connect-It used.

When importing favorites from a previous version to the version used, we recommend checking the names of each favorite to make sure there are no conflicts with the versions used. If there are any conflicts, the application renames them automatically.

Defining a default favorites file

The **Edit/ Options** menu enables you define a default favorites file.

Specify the file to use for the **Connector/ Favorites file** option.

Scenario history

All information related to the creation and modification of a scenario is available via the **File/ Scenario history** menu.

The **General** tab enables you to populate the scenario with your personal information. A summary is available in the **History** tab. A new summary is created each time the **Add** button is used. The existing summary is updated each time the **Update** button is used.

All the differences between two versions of a Connect-It scenario are provided in these tabs.

Testing and debugging an integration scenario

Testing a scenario enables you to know if:
- The source connector correctly produces the documents for which you created produced document-types.
- The mapping box correctly transforms these documents.
- The destination connector correctly consumes these documents after the mapping box transforms them.

> ⚠ Warning:
>
> To perform these tests, we recommend using test data (demonstration files and databases). Untested scenarios that are put into production mode could seriously damage your real data.

Using the logs

Using the Document log enables you to see if any problems occurred while processing the documents produced by a source connector.

To deal with processing problems, we recommend configuring the Document log so that only the problem documents are displayed.

The Scenario builder enables you to consult two logs:

- The Connect-It log
- The Document log

The Connect-It log

This log informs the user of actions performed by Connect-It.

For example: A message is issued when a connector is opened.

To access the Connect-It log, select the *Connect-It Log* tab in the Scenario builder.

In the log, each action is represented by an icon: ❶. An action's message can be composed of several sub-messages that detail the action. These sub-messages can, themselves, be composed of other sub-messages.
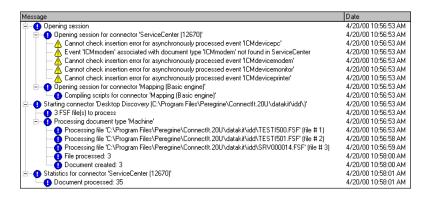
> 📝 Note:
>
> Double-click on a tracking-line to display the complex element (field, structure, or collection) that triggered the message (right window).

Each message is dated according to when the action was launched.

You can unfold or collapse messages by right-clicking and then selecting the appropriate command from the shortcut menu.

Figure 5.1. The Connect-It log

For an example of the messages appearing in the Connect-It log, refer to the *Connectors* guide, chapter *Connector directives*, section *Consumption directives \ Reconciliation \ Error messages concerning the choice of reconciliation keys*.

Icons used in the Connect-It log

Tracking line	Meaning	Message type	Example
●	Information about an action	Describes an action performed by Connect-It.	The opening of a connector.
●	Details of an action	Gives details concerning an action.	Indication that a particular index is used in the reconciliation process.
⚠	Warning	Warns of any problems that could result in a bad data processing.	The field chosen as a reconciliation key does not have a uniqueness constraint.
⊖	Major problem	Gives the reason why an action failed.	The configuration of a connector is not valid. Unable to communicate with the external application.

To clear all messages from the log, select the option **Clear displayed tracking lines** in the **Monitor** menu, or click 🖳.

Purging the Connect-It log also purges the Document log.

Document log

The Document log lets you check the details of documents after they have been:
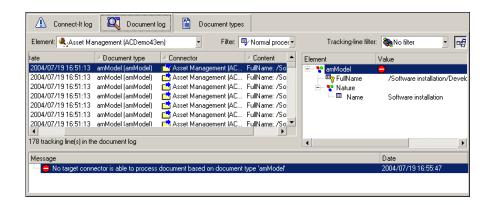
- Produced or consumed by a tool.
- Routed by a link.

The Document log is displayed when you select the **Document log** tab in the Scenario builder.

This tab is organized into three panes:

- A first showing the tracking lines corresponding to documents.
- A second showing the details of the selected document.
- A third showing messages detailing the tracking lines left by elements in the document.

Figure 5.2. Scenario builder - Document log tab

Configuring the Document log

To configure the document log, select **Monitors/ Configure monitors**, **Document log** tab.

This menu enables you to save your documents, their tracking lines and any tracking lines left by scenario elements.

For further information on configuring the document log, refer to chapter Ergonomics [page 39], section Define the supervision monitors [page 151].

Some options of the document log are accessible via the **Edition/Options** menu:

- **Display** category
- **Connector** category
- **Logs** category

For further information on the options for the document log, see chapter Edit the scenario options [page 60].

Clearing memory

A purge command is available in the **Monitors** menu:

❖ **Clear displayed tracking lines**

This command enables you to clear the log entries in the Scenario builder: Connect-It log, Document log. This command does not empty the files that permanently store the tracking lines.

Refresh tracking lines

The **Refresh tracking lines** command in the **Monitors** menu enables you to display all tracking lines in the **Document log** tab. This command is particularly useful if you have restricted the number of tracking lines displayed in the Document log. In this case, this command enables you to view all the problems encountered while processing your documents.

Refresh is performed automatically and also be performed manually by pressing **F9**.

Tracking lines

The Document log contains tracking lines. Each one corresponds to a document processed by a scenario element and includes:

- An identifier.
- An icon symbolizing how the document was processed.
- The data and time of processing.
- The document type that was processed.
- The name of the element that produced or consumed the document.
- The contents of the document.

The identifier corresponds to the order in which the documents were processed; number *1* being the first. This unique identifier allows you to find documents when they have been kept in memory.

You can click on the header of each column in the Document log to apply an ascending or descending sort.

A blue triangle pointing upwards means that the sort is ascending; a blue triangle pointing downwards means that the sort is descending. When no sort is applied, the triangle is grayed out.
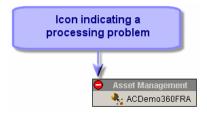
If an element is cut off in the display, you can hold the mouse pointer over the element; it is displayed in full as a ToolTip.

Icons used in the Document log

Each icon represents information of increasing importance. For example: If certain fields only receive warnings and others are completely rejected, only the ● icon appears in the document log. The full rejection of a document is considered more important than a warning.

Icons are also used in the Scenario diagram, as shown below. These icons are displayed at the level of a scenario element if: one of more documents have been rejected by this element; contain fields that have been rejected; or have generated warnings.

Figure 5.3. The processing-problem icon

When an icon appears on a component, if you right-click the tool and then choose **Show tracking lines** from the shortcut menu, the icon disappears, meaning that you have noticed the problem.

Filtering tracking lines

You can filter the tracking lines shown in the document log according to:
- The element that processed the document.
- How the document was processed.

If no elements are selected in the Scenario diagram pane, all available tracking lines are displayed in the Document log.

To only show tracking lines for documents processed by a given element:
- Select this element in the Scenario diagram.
- Or select an entry in the **Element** field.

Figure 5.4. Filtering tracking lines in the document log

To show tracking lines by processing type, select an entry in the **Filter** field.

Only the selected processing type and higher level messages are displayed.

For example: When you select **Field reject**, fully rejected documents are also displayed.

Figure 5.5. Filtering tracking lines

Detail of a document

The detail of a document gives you information about the value of the fields in the document and the way in which the document was processed by a tool or a link.

When you select a tracking line in the Document log, the detail of the document concerned is displayed in the right-hand pane of the Document log.

The document is represented as a tree.

Each tree node shows:
- The name of the corresponding element.
- The value of this node in the document.
- One or more possible tracking lines left by the tool or the link that processed the document.

All the tracking lines in the document detail have messages that explain, for example, why the document was rejected.

Differences between the document detail and the document type detail

The document type detail is displayed when you edit the produced or consumed document-types. Just like documents, document types are represented as a tree. However, there are several differences.

In a document detail, each node has a value. For structures or collections, this value is a description string resuming the contents of the fields in the structure or the collection. Thus, for example, the root node of a document corresponding to an asset in AssetCenter is described by its Asset tag, which is shown in gray. For collections, the number of members is indicated, etc.
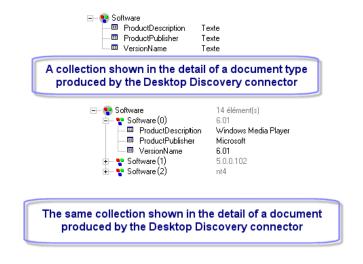
In the detail of a document type, only the name of the node is shown.

Figure 5.6. A description string in the document

In the detail of a document, each member of a collection is represented as a child structure of the collection. Each child node bears the name of the collection followed by a number in square brackets. In a three-member collection, the first member is numbered 0; the second, number 1; and the third, number 2.

In the detail of a document type, only the collection is shown, along with the expected value-types (text, integer, date).

Figure 5.7. A collection shown in a document type detail and in a document detail
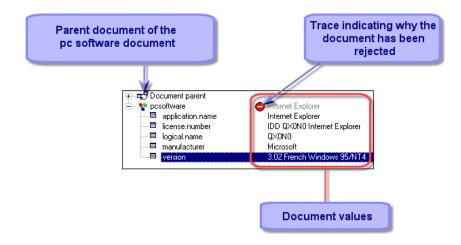
Parent document

The detail of a document can contain a parent document. This parent document contains the data used to create the child document. Thus, the detail of a document produced by the mapping box also includes the detail of the source document from which it was created.

For example: In the pdiac.scn scenario, the detail of the documents produced by the mapping box includes the detail of the documents produced by the Desktop Discovery connector. These documents come from the source document-type shown in the mapping.

If a document field is rejected by the mapping box, you can check the value of this field in the parent document for details as to why the document field was rejected.

Figure 5.8. Detail of the pcsoftware document consumed by the ServiceCenter connector

Manipulating the XML document corresponding to the detail of a document

The detail of a document produced in Connect-It is an XML document. You can:

- Copy this XML document.
    1 Place the pointer on the document detail.
    2 Right-click.
    3 Choose **Copy this XML document** from the shortcut menu that appears.
- Copy this document's DTD.
    1 Place the pointer on the document detail.
    2 Right-click.
    3 Choose **Copy this XML DTD** from the shortcut menu that appears.
- Open this XML document.
    1 Place the pointer on the document detail.

2   Right-click.

3   Choose **Open this XML document** from the shortcut menu that appears.

> 📝 Note:
>
> The XML document is opened by the application that is associated with files
> having an `.xml` extension on the computer on which Connect-It is installed.
> For example: Internet Explorer, Netscape, etc.

Finding an element in the detail of a document

A searching function enables you to find elements in the detail of a Document
log document.

To search for an element in the detail of a document:

1   Select an element from the detail of a document.

2   Right-click.

3   Select **Find** from the shortcut menu.

4   Wait for the dialog box to appear.

5   Enter the name of the element that you want to find in the **Find what** field
    and select the options of your choice:

   ■   Case sensitive

   ■   Select the direction of your search: up or down.

6   Click **Find**.

To search through previous elements or following elements:

1   Perform one of the following actions:

   ■   Right-click and select **Find next** (**F3**) from the shortcut menu.

   ■   Right-click and select **Find prev.** (**Shift + F3**) from the shortcut menu.

> 📝 Note:
>
> The search is limited to the elements displayed in the document detail. If you want
> your search to apply to the sub-nodes of a given node, you must unfold the node
> before the search.

Tracking lines used in document details

A tracking line is a piece of information left by a scenario tool or link, concerning
the processing of a document field.

Tracking lines are symbolized by icons shown in the document detail. Each
tracking line is accompanied by a message in another pane.

There are two categories of tracking lines:

- Informational tracking lines (important information or details concerning a field)
- Warning tracking lines (warnings, errors, etc.)

Only the error tracking line ⊖ signals the full or partial rejection of the document.

The following table summarizes the different tracking lines.

Track-ing line	Meaning	Message type	Example
❶	Detailed inform-ation	Technical information about the processing.	The IND connector downloads data that it converts into documents. The detailed information gives the size of files in KB, the download speed, etc.
❶	Important in-formation	Important information about how the data is used.	The name of the folder and the files cre-ated by the file connector after consuming documents.
⚠	Warning con-cerning a field	Reasons for a which a field could not be pro-cessed correctly.	The ServiceCenter connector has accepted a field but has had to truncate it to maintain the integrity of the ServiceCen-ter data-model.
⊖	Error	Reason for failure.	A field is rejected because it contains an unexpected value. This can lead to the partial or full rejection of the document.
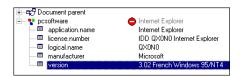
Gray tracking lines next to a node

In the document detail, some tracking lines may be shown in gray.

A gray tracking line simply means that at least one of its sub-nodes has a tracking line.

When there is more than one tracking line in the sub-nodes, only the highest level tracking line is shown in gray next to the node.

Figure 5.9. Grayed-out tracking line indicating a tracking line under the node.

Filtering by tracking-line type

It is possible to filter the document detail by tracking-line type. This allows you to see only those elements that have this tracking line (or tracking lines that are more important).

If you are only interested in the rejected nodes, select ● entry in the **Filter tracking lines** field, found on the top of your document detail pane.

When you select an entry in this drop-down list, all the nodes with tracking lines below this entry are displayed.

The order of information in this list depends on the importance of the information: from detailed information about a field to failure of processing.

Figure 5.10. Filtering tracking lines in the document detail

Explanatory messages

All tracking lines shown in the detail of a document have an accompanying message.

Simply select a tracking line in the document detail; the message is displayed in the pane below the document log.

For example: An error is accompanied by a message explaining that the field has been rejected by the Asset Management connector because its value violates the data model of the connected database.

Display the sub-messages

To display the ensemble of messages contained in the child nodes of your document's parent node, click ▦.

Example

The ⊞ option is not selected: Only the message of the selected node is displayed.

⚠ Text value ('Award Modular BIOS v4.51PG - war,Copyright (C) 1984-98, Award Software, Inc. - Award SoftwareIBM COMPATIBLE 486 BIOS COPYRIGHT Award Software I...

The ⊞ option is selected; all the messages in the child nodes are displayed.
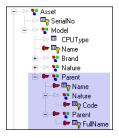
⊟ No reconciliation available for document type 'amAsset (amAsset)'. You need to define 'amAsset (amAsset)' as a consumed document type.
⚠ Text value ('Award Modular BIOS v4.51PG - war,Copyright (C) 1984-98, Award Software, Inc. - Award SoftwareIBM COMPATIBLE 486 BIO!

Use the script debugger

A basic script can be debugged using the **Debug script** option in the mapping box.

Click ▶ to debug a scenario.

When starting a scenario in debug mode, a modal box containing the script to debug opens.

For each line of script, the following debug instructions are used:
- Step over: The instruction is executed.
- Step into: Functions in the instruction are executed one by one. Their value is displayed.
- Step out: Returns to the calling function or exits debug mode if currently in the main function.
- Run to cursor: Functions are executed up to the cursor.

To display the value of a variable used in a script, use the following shortcuts:
- Double-click the variable
- Ctrl-Shift-F9
- Shift-F9

Use the processing reports

A processing report is a document that a connector or a mapping box can produce every time that they produce or consume a document.

Each processing report informs you if a document has been correctly processed or not. A successfully processed document is a document in which no elements have been rejected.

There are two different types of processing reports: Consumption reports and production reports.

A production type processing report is generated by a connector producing documents

A consumption type processing report is generated by a connector consuming documents

The processing report is like a document type that can be produced by each connector and the mapping box. The following table details the information contained in a processing report.

Table 5.1. Information in the processing reports

Element	Information	Field type
ProcessReport	Root node of the processing report.	
DocumentType	Name of the produced or consumed document-type.	Text
DocumentTypeID	Unique ID of the document.	Text
ErrorNumber	Number of errors occurring during the processing of a document.	Integer (32 bit)
Success	Normal processing (value: 1) or abnormal processing (value: 0) of a document.	Boolean
	Abnormal processing means that the document was totally rejected.	
Pending	Document pending or not	Boolean
	Corresponds to a document pending validation by an AssetCenter user	
Ignored	Document ignored (**PifIgnoreDocumentMapping** function)	Integer (32 bit)
WarningNumber	Number of warnings occurring during the processing of a document.	Integer (32 bit)
Logs	Collection corresponding to messages issued when processing a document.	
Date	Date of the message.	Date and time.

Element	Information	Field type
▣ LogType	Type of message.	Integer (32 bit)
	The possible values are:	
	▪ 1 (Error)	
	▪ 2 (Warning)	
	▪ 4 (Information)	
	▪ 16 (Additional information)	
▣ Msg	Contents of the message.	Text
▣ Path	Path in the document of the element concerned by the message.	Text

Processing report - Example of use

In several of the scenarios provided with Connect-It, the processing reports are used to condition an action (deleting, moving, etc.) on the files used by the source connector.

To integrate the processing reports into your scenarios, we recommend that you study the provided scenarios that use them.

The following section details how the Hewlett-Packard Desktop Discovery connector uses processing reports in the `pdi\pdi8ac44\pdiac.scn` scenario. This scenario enables you to insert or update records in the AssetCenter database via `.xsf` files which are processed by the Hewlett-Packard Desktop Discovery connector.
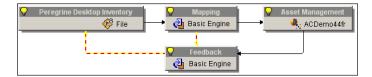
Using processing reports in the `pdi\pdi8ac44\pdiac.scn` scenario

- the Hewlett-Packard Desktop Discovery connector
- the Asset Management connector

In this scenario, a mapping box named **Feedback** enables you to send the processing reports produced by the mapping box Mapping to:

This **Feedback** mapping box contains two mappings:

- A mapping between the mapping box's process report, **ProcessReportFromMapping**, and the Hewlett-Packard Desktop Discovery source connector document type, **DirectoryPoolerAction**.

- A mapping between the Asset Management connector's processing report, **ProcessReportFromUpdate**, and the Hewlett-Packard Desktop Discovery source connector's **SuccessReportMapping** document type.
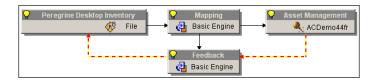
Table 5.2. Detail of the mapping between the ProcessReportFromMapping processing report produced by the Mapping mapping box and the SuccessReportMapping document type consumed by the Hewlett-Packard Desktop Discovery connector

SuccessReportMapping document type element	Processing report or script element
▣ SuccessReportMapping (root node)	
Comment: The processing report is only produced if the document is rejected in part or in full by the mapping box. This behavior is defined at the level of the **SuccessReportMapping** produced document type for the **Feedback** mapping box.	
Path	[$ParentDoc$.UrlFileInfo.Path]
Enables you to obtain the **UrlFileInfo.Path** field from the **Inventory** document type produced by the Hewlett-Packard Desktop Discovery connector. This field corresponds to the name of an .xsf file processed by the connector.	

Success	`if [ErrorNumber] = 0 then`
	 `RetVal = [Success]`
	 `else`
	 `RetVal = 0`
	 `end if`

Comment: Boolean field

- ◆ The value '0' indicates that the document has been rejected in part or in whole by the mapping box.
- ◆ The value '1' indicates that the document has been normally processed by the mapping box.

Table 5.3. Detail of the mapping between the processing report produced by the ProcessReportFromUpdate Asset Management connector and the

SuccessReportMapping document type consumed by the Hewlett-Packard Desktop Discovery connector

SuccessReportUpdate document type element	Processing report or script element
🖳 SuccessReportMapping (root node)	```if [Pending] = 1 then PifIgnoreDocumentMapping end if```
Path	`[$ParentDoc$.$ParentDoc$.UrlFileInfo.Path]`

Comment: Enables you to obtain the **FileInfo.FileName** field from the **Inventory** document type produced by the Hewlett-Packard Desktop Discovery connector. This field corresponds to the name of an .xsf file processed by the connector.

Success	```if [ErrorNumber] = 0 then RetVal = [Success] else RetVal = 0 end if```

Comment: Boolean field

- ❖ The value '0' indicates that the document has been rejected in part or in whole by the Asset Management connector.
- ❖ The value '1' indicates that the document has been normally processed by the Asset Management connector.

> 🖉 Note:
>
> *Use of the $ParentDoc$ variable*
> - ■ The [$ParentDoc$. field] notation enables you to obtain the value of this field in the parent document of a processed document.
> - ■ The [$ParentDoc$.$ParentDoc$. field] notation enables you to obtain the value of this field in the parent document of a parent document of a processed document.
> - ■ Etc.

Effect of the processing reports on the .xsf files processed by the Hewlett-Packard Desktop Discovery connector

Each processing report allows the Hewlett-Packard Desktop Discovery connector to perform a particular action on an .xsf file corresponding to the document concerned by the report. The **FileInfo.FileName** field of each processed document enables you to identify the corresponding .xsf file.

This particular action is selected during configuration of the Hewlett-Packard Desktop Discovery connector.

Example :

The following options are available to manage your .xsf files:

- In the case where the document was successfully processed, the .xsf file that corresponds to it is deleted from its folder of origin.
- In the case where the processing of the document failed, the .xsf file that corresponds to it is moved to a `Failure` folder.

Once the documents have been processed by the scenario, you can open the `Failure` folder to obtain the .xsf files. These files contain the data that could not be correctly transferred to your Asset Management application.

To spot problems arising from the total or partial rejection of a document, consult the Document log in the Scenario builder.

.xsf files whose corresponding documents are successfully processed by the Asset Management connector (**Success** field with the value "1") are deleted by Hewlett-Packard Desktop Discovery from their folder of origin.

Post-processing actions

For certain connectors, there is a predefined policy for managing errors or successes.

This error-management policy can be access in the connector configuration page called **Define post-processing actions** and in the **Manage process reports** wizard.

Not all the connectors have this function available.

In order for a source connector to use a processing report, it must publish a **SuccessReport** document type.

When a source connector uses a mapping between the **ProcessReport** and the **SuccessReport** document type, it can perform a particular action: It can transform the values it extracted from the files into documents.

A **Manage processing reports** wizard enables you to automatically create a mapping between a processing report and the **SuccessReport** document type of a source connector.

Processing report and post-processing actions

For a connector to be able to use a processing report, it must:

- Publish a **SuccessReport** document type.

  This document type is for mapping the elements contained in a processing report (**ProcessReport** document type).

  The following connectors have a **SuccessReport** document type:
  - XML connector
  - Delimited-text connector

- SAP IDOC connector
- Hewlett-Packard Desktop Inventory connector
- E-mail (fetching) connector
- MQSeries connector

- Have a **Define post-processing actions** page in its configuration wizard.

  This page enables you to set a document's post-processing actions after the connector consumes the **SuccessReport** document.

  Example: The **Define post-processing actions** page lets the connector to put all the documents rejected by the destination connector into the **error** folder.

Note:

The connectors not meeting this criteria can use processing reports, but they can use the **Manage processing reports** wizard.

Contents of the SuccessReport document type

The contents of the **SuccessReport** document type vary depending on the connector.

The following tables detail the contents of these document types.

Connectors	Element	Information	Field type
▪ XML connector	▣ SuccessReport	Root node of the processing report.	
	▣ Success	Status of the file processing (success or failure)	Boolean
▪ Delimited-text connector	ᴛ UrlFileInfo	Structure containing information about the processed document.	
▪ PDI connector	▣ Path	Path of the file.	Text
▪ SAP IDOC connector			

Connectors	Element	Information	Field type
◆ E-mail (fetching) connector	▣ SuccessReport	Root node of the processing report.	
	▣ MailInfo	Structure containing information on the processed message	
	▣ UniqueID	ID of the message.	Text

Connectors	Element	Information	Field type
◆ MQSeries connector	▣ SuccessReport	Root node of the processing report.	
	▣ Success	Status of the file processing (success or failure)	Boolean
	ᴛ MessageInfo	Structure containing information on the processed message	
	▣ MsgID	ID of the message.	Integer
	▣ PutDate	Date and time the message was processed.	Date

Using the Define post-processing actions page

The processing report lets a connector producing documents from files perform actions after the successful or unsuccessful processing of these files.

- Leave the files in their original folder.
- Delete the files from their original folder.
- Move the files to another folder.
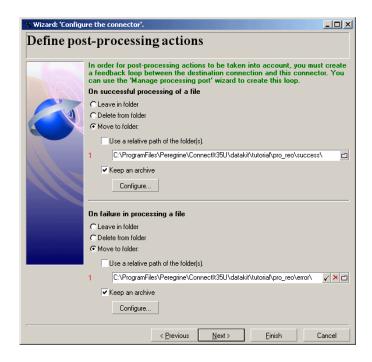
    Example: an **error** file.

For all connectors, here are the available post-processing actions:

To display the **Define post-processing actions** page

1 Select one of the connectors having this page.

    Example: the **Delimited text** connector

2 Select the **Tools/Configure** menu (**F2**) to launch the configuration wizard.

3 Click **Next** in the wizard until you get the **Define post-processing actions** page.

The following screen capture shows the **Define post-processing actions** page for the XML connector.

Principle for handling processing reports

Post-processing actions are all applied when closing the session for the connector. Processing is performed as follows:

1   during the session, information from the **SuccessReport** document consumed by the connector is stored and updated progressively,

2   when the session closes, the post-processing actions are applied for all resources (files, e-mails, messages, etc.) for which at least one **SuccessReport** document was received.

Any incomplete **SuccessReport** document that is produced generates an error when it is consumed. In verbose mode, actions performed on resources are recorded in the log file. Note that by default the scenario builder is enabled in verbose mode and that conitsvc (Connect-It application in non-graphic mode) is in non-verbose mode by default.

Manage processing reports wizard

The **Manage processing reports** wizard enables you to create (for a document type produced by the source connector):

- A feedback loop between a destination connector consuming this document type and the source connector.

  This feedback loop maps the information from the processing report produced by the destination connector and the SuccessReport document type of the source connector.

- A possible feedback loop between a mapping box and the source connector.

Using the Manage process reports wizard

1 Start the Scenario builder.
2 Open a scenario in which the source connector publishes a **SuccessReport** document type.

  To find out which connector publish this document type, refer to the section Post-processing actions [page 229] and the *Connectors* guide.
3 Select the source connector producing the document type for which you want to create a feedback loop.
4 Select the **Tools/Wizard/Manage processing reports** menu.
5 On the **Select a source document type** page, indicate the document type for which you want to create a feedback loop.
6 On the **Select a destination connector** page, indicate the destination connector for which you want to recover the processing report of the source document-type selected on the previous page.
7 You can select the option **Force the production of a processing report from a mapping box linked to the source connector** if you want to.
8 Click **Finish**.

Tutorial

The tutorial of this section enables you to use the **Manage processing reports** in a scenario using an XML source connector and an Asset Management destination connector.

The data necessary for this tutorial is contained in the following folder:
`[Connect-Itinstallation folder]\datakit\tutorial\pro_rep`.

Prerequisites

To use this tutorial, you must:

- Open the `pro_rep.scn` scenario, located in the tutorial folder.
- Have access to an AssetCenter 4.1 or 4.2 demonstration database.

Description of the tutorial scenario

In the tutorial scenario:

- An XML source connector produces documents from `.xml` files, contained in the `files` sub-folder of the tutorial folder.
- A mapping box consumes these documents and produces them so that the Asset Management connector can consume them.
- An Asset Management connector creates records in the **amAbsence** table of a test AssetCenter database.

Mapping between the XML connector and the Asset Management connector

The following table details the scenario mapping between the XML connector's **amAbsence** document-type and the Asset Management connector's **amAbsence** document-type.

Element of the destination document-type	Element of the source document-type	Mapping script

amAbsence	amAbsence	`if [Nature] = "" then`
		`PifRejectDocumentMapping("Document re`
		`jected: missing nature")`
		`end if`
		
		This script enables you to reject the document if the **Nature** field of the document type produced by the XML connector is empty.

AccountingType	AccountingType	AccountingType
Nature	Nature	Nature reconciliation key)
PhoneContact	PhoneContact	PhoneContact (reconciliation key)

Tutorial files

The **files** sub-folder of the tutorial folder contains the following `.xml` files:

- `success.xml`

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<amAbsence>
<AccountingType>Convention</AccountingType>
<bArchived>1</bArchived>
<dtBack>2002-06-14 18:00:00</dtBack>
<dtLastModif>2002-07-12 05:30:32</dtLastModif>
<dtOut>2002-06-14 09:00:00</dtOut>
<dtValidation>2002-01-15 18:00:00</dtValidation>
<fDays>1</fDays>
<Field1>Test</Field1>
<Nature>Meeting</Nature>
<PhoneContact>06.12.11.81</PhoneContact>
```

```
<Reason>San Diego Convention</Reason>
<seValidated>1</seValidated>
</amAbsence>
```

This file must be processed successfully by all the scenario's tool and must be moved after the scenario's test into the `success` folder.

- `src_fail.xml`

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<amAbsent>
<AccountingType>Convention</AccountingType>
<bArchived>1</bArchived>
<dtBack>2002-06-14 18:00:00</dtBack>
<dtLastModif>2002-07-12 05:30:32</dtLastModif>
<dtOut>2002-06-14 09:00:00</dtOut>
<dtValidation>2002-01-15 18:00:00</dtValidation>
<fDays>1</fDays>
<Field1>Field1</Field1>
<Nature>Meeting</Nature>
<PhoneContact>06.12.11.81</PhoneContact>
<Reason>San Diego Convention</Reason>
<seValidated>1</seValidated>
</amAbsent>
```

This file must be rejected by the XML connector because its **amAbsent** structure does not correspond to the one defined in the `pro_rep.dtd` DTD, used by the connector.

After testing the scenario, this file must go into the `error` folder.

- `map_fail.xml`

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<amAbsence>
<AccountingType>Convention</AccountingType>
<bArchived>1</bArchived>
<dtBack>2002-06-14 18:00:00</dtBack>
<dtLastModif>2002-07-12 05:30:32</dtLastModif>
<dtOut>2002-06-14 09:00:00</dtOut>
<dtValidation>2002-01-15 18:00:00</dtValidation>
<fDays>1</fDays>
<Field1>Test</Field1>
<Nature></Nature>
<PhoneContact>06.12.11.81</PhoneContact>
<Reason>San Diego Convention</Reason>
<seValidated>1</seValidated>
</amAbsence>
```

This file must be rejected by the mapping box because the **<Nature>** element is empty.

After testing the scenario, this file must go into the `error` folder.

- `dest_fail.xml`

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<amAbsence>
```

```
<AccountingType>Convention</AccountingType>
<bArchived>1</bArchived>
<dtBack>2002-06-14 18:00:00</dtBack>
<dtLastModif>2002-07-12 05:30:32</dtLastModif>
<dtOut>2002-06-14 09:00:00</dtOut>
<dtValidation>2002-01-15 18:00:00</dtValidation>
<fDays>1</fDays>
<Field1>Test</Field1>
<Nature>Meeting</Nature>
<PhoneContact></PhoneContact>
<Reason>San Diego Convention</Reason>
<seValidated>1</seValidated>
</amAbsence>
```

This file must be rejected by the Asset Management connector because the **<PhoneContact>** element selected as the reconciliation key (connector's consumption directives) is empty.

After testing the scenario, this file must go into the error folder.

Step 1 - Configuring the XML connector of the example scenario

1 Select the scenario's XML connector.
2 Start the connector configuration wizard
3 On the **Select a processing mode** page, select the **Read** option.
4 On the **Select a connection protocol** page, select the **Local/network file(s)** option.
5 On the **Choose a file or folder** page:
    1 Select the **Read files** option.
    2 Click ⬢ and indicate the location of the files sub-folders in the tutorial folder.
6 On the **Define post-processing actions** page, select the option **Move to folder** in both cases.

    For the **On successful processing of a file** part of the page, indicate the path of the success sub-folder.

    For the **On failure in processing a file** part of the page, indicate the path of the error sub-folder.
7 On the **Choose a DTD/XSD** page, indicate the path of the pro_rep.dtd DTD in the tutorial folder.
8 Click **Finish**.

Step 2 - Configuring the Asset Management connector of the example scenario

1 Select the scenario's Asset Management connector.
2 Start the connector configuration wizard

3 On the **Define the connection parameters** page, enter the connection parameters of your AssetCenter test database.

4 Click **Finish**.

Step 3 - Using the Manage processing reports wizard

1 Select the XML connector

2 Select the **Tools/Wizard/Manage processing reports** menu.

3 On the **Select a source document type** page, select the **amAbsenceSrc** source document-type.

4 Click **Next**.

5 On the **Select a destination connector** page, select the Asset Management connector.

6 On the **Advanced options** page, select **Force the production of a processing report from a mapping box linked to the source connector**.

7 Click **Finish**.

The wizard creates a mapping box between the XML connector and the Asset Management connector. A mapping is created in the mapping box, which is detailed in the following table.

Element of the **SuccessReport** document type consumed by the XML connector	Element of the processing report produced by the Asset Management connector

Success	`if [ErrorNumber] = 0 then`
	`RetVal = [Success]`
	`else`
	`RetVal = 0`
	`end if`

Comment: Boolean field

❖ The value '0' indicates that the document was rejected in part or in full by the Asset Management connector.

❖ The value '1' indicates that the document has been normally processed by the Asset Management connector.

UrlFileInfo	`[$ParentDoc$.$ParentDoc$.UrlFileInfo.Path]`

Element of the **SuccessReport** document type consumed by the XML connector	Element of the processing report produced by the Asset Management connector

Enables you to recover the **UrlFileInfo.Path** field from the **amAbsence** document type, produced by the XML connector. This field corresponds to the name of the .xml file read by the XML connector.

> 🔔 Warning:
>
> When using the Manage process reports wizard, the fields referenced in the *$ParentDoc$* function are not automatically added to the produced document type. You must edit the document type produced by the source connector and add the elements used in the *$ParentDoc$* function.

Step 4 - Testing the feedback loop created by the Manage processing reports wizard

1 Select the XML connector.
2 Click ▶.

   In the scenario diagram, the ⊝ icons must appear on the scenario tools. These icons signal the failure of processing the .xml files. Select the Document log tab in the Scenario Builder and read the error messages that were generated.
3 Open the success sub-folder and verify that the success.xml file is there.
4 Open the error sub-folder and verify that the dest_fail.xml, map_fail.xml and src_fail.xml are there.

> 🖉 Note:
>
> When using the feedback loop created with the Manage process reports wizard, all the elements used in the feedback mapping (names of the processed files to be moved after processing) are automatically added to the produced document type.

Processing report - How it works
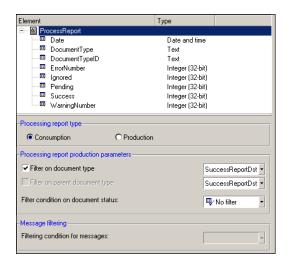
All connectors can produce a processing report as a document type and choose how it will behave. This report will analyze the documents produced or consumed by the connector.

For example, the Asset Management source connector, produces two produced document types: **amAsset** and **ProcessReport**. If the processing report generated is a production report, it analyzes the documents produced by the Asset Management connector.

Conversely, the Asset Management connector consumes a document type, **amBudget** and produces the **ProcessReport** document type. If the processing

report generated is a consumption report, it analyzes the documents consumed by the Asset Management connector.

Filtering a processing report

You can filter the data produced by a processing report according to the document type produced or consumed by the connector.

The filter is available in the **Processing report production parameters** frame.

To filter on a document type:

1 Select the **Filter on document type**
2 In the drop-down list, select the document type consumed or produced on which you want to filter.

   In this case, the processing report will only be produced on consuming or producing the selected document type.
3 Select a filter condition on the document status, if necessary: **Warning**, **Field reject**, **Reject**.

The **Filter on parent document type** option is only enabled for production-type processing reports. This option is particularly useful for the connectors producing documents on consumption (Web Service, SAP BAPI). In this case, the filter does not concern the document type produced but the document type consumed that gave rise to the production of the document.

In addition to the filters applied to a document type, you can filter the messages that appear in the document log.

To filter the messages:

- Add the **Logs** collection to the produced processing report
- Select the collection
- In the **Message filtering** frame, select the filter from among those available (**Filtering condition for messages** field)

  For more information on the meaning of filter conditions, refer to section Document log [page 214].

Syntax

When you want to use the information contained in a remote document as well as a connector for you mapping script, you have the choice of using two variables:

- $ParentDoc$
- $LinkedDoc$

The $ParentDoc$ variable can be used to reference the consumed parent document: For more information on using this variable, refer to the section Processing report - Example of use [page 226] of this chapter.

The $LinkedDoc$ variable can be used to reference the document linked to the consumed processing report: A processing report is produced on processing a document. The linked document, called by this variable, is the document that gave rise to the production of the processing report.

Testing a produced document-type

In a scenario where a connector produces more than one document type, it is necessary to test each one of these produced document-types before going into production mode.

To test a document type produced by a source connector:

1 Open the source connector.
2 Select the **Document types** tab.
3 Clear all document types produced by the connector except the one that you want to test.

  If you want to test several document types, do not clear them before going onto the next step.
4 Launch the production of the document type by clicking ▶ or selecting **Produce** in the **Tools** menu.
5 Consult the Document log to see if any problems were encountered while processing the documents produced by your source connector.

> **Note:**
>
> For the Scenario builder options (Edit/Options), the **Number of documents to produce in a test production of documents** option sets the number of documents to produce during a test.
>
> This option is automatically taken into account when you test a produced document-type. On the other hand, when the scenario is executed as a Windows service, this option is automatically deactivated.

Debug mapping scripts

It is sometimes difficult to locate a syntax error in Basic script (mapping/concerned element). We thus recommend that you test scripts one by one after you write them, rather than write them all and test them all at the end.

Using the cache files

The cache files used by the connectors contain the description of the document types that they publish.

Example :

When you open a database connector, it automatically obtains the description of all the database tables indicated in its configuration. By using a cache file, the connector no longer needs to obtain these descriptions; it uses the descriptions recorded in the cache file that was created when the connector was last opened.

This function is very useful in the testing and debugging phase of a scenario in which you will open and close the connectors numerous times.

The following connectors can use cache files:

- Action Request System connector
- Asset Management connector
- ServiceCenter connector
- Hewlett-Packard Network Discovery connector
- LDAP connector
- Database connector
- Lotus Notes connector
- All inventory connectors

In order for a connector to use a cache file instead of connecting to its external application:

1 Select the **Use cache file** option on the **Configure cache** page of the of the **Configure the connector** wizard.

2 Open your connector for the first time so that its cache file contains the description of the document types that it publishes.

   Note: When you first open the connector, the icon designating the use of the cache does not appear on the connector illustration.

3 Close, then re-open the connector.

   This time, when you open your connector, it will no longer connect to its external application, but instead it will use the cache file to publish its document types.

Note:

Each time you open a connector that uses its cache file, an icon symbolizing a database will replace the icon of the yellow light bulb.

Figure 5.11. Connector - Icon indicating the use of a cache file

Deleting the contents of the cache files

To delete the contents of the cache files:

1 Open the connector.

2 Select **Delete the caches** in the **Scenario** menu.

   You can delete the contents of a single cache file by selecting **Cache/Delete the cache** in the **Tools** menu.

   The next time you open your connectors, they will connect to their external applications to synchronize the contents of their cache files with the contents of the external applications.

Warning:

You can only synchronize the cache files if the value of the **Work offline** option is set to **No** (in the options screen of the Scenario builder).

Synchronizing the contents of the cache files with the contents of the external applications

To synchronize the contents of the cache files with the contents of the external applications:

1 Open your scenario's connectors.

2 Select **Synchronize the caches** in the **Scenario** menu.

You can synchronize the contents of a single cache file by selecting **Cache/Synchronize the cache** in the **Tools** menu.

3 Wait for Connect-It to obtain the description of the document types published by the connector.

Once the synchronization phase is complete, the document types published by the connector correspond to the contents of the external application.

Warning:

You can only synchronize the cache files if the value of the **Work offline** option is set to **No** (in the options screen of the Scenario builder).

Working offline

Working offline enables you to work without needing the scenario connectors to be connected to their external applications.

To work offline, click ⬚ after having opened your connectors for the first time. A dialog box informs you that the mode has changed.

The off-line mode automatically creates a cache file for the scenario connectors.

When you close and reopen your connectors while the 'work offline' mode is still activated, the connectors will use the contents of their cache files to publish document types.

Note:

Each time you open a connector when working offline, and the connector cache is enabled, an icon representing a database will appear to the side of the switched-off light bulb icon.

Figure 5.12. Connector - Icon indicating an offline session

Enabling the test mode

- If the destination connector supports transactions, then the data will be inserted in test mode. However, a rollback will be performed on this transaction (a return to the initial state before data insertion). This is so you can test document consumption without having to modify the data in the destination application.
- If the destination connector does not support transactions, then the data will not be sent to the destination application. This enables you to test the production and the mapping, but not the consumption.

📝 Note:

A dialog box shows that the test mode is enabled.

Managing the tracking of a scenario using the Service console

To manage the tracking of a scenario, use the console to:
- Modify the configuration of your connectors.
- Editing your scenario's schedulers
- Consult the Connect-It logs.
- Consult a LOG file associated with your documents.

Modifying the configuration of your connectors

To modify the configuration of your connectors in a scenario for which you have created a service:

1 Select the service corresponding to your scenario.
2 Click **Configure**.
3 In the list, double click the connector to be configured, or click 🖿
4 Populate the different pages of the **Configure the connector** wizard for each connector, then validate your configuration (**Validate** button).

After you have finished configuring the last connector, the Scenario builder automatically closes.

Edit schedulers

To edit the schedulers in a scenario for which you have created a service:
1 Select the service corresponding to your scenario.
2 Click **Scheduling**.
3 Wait for Connect-It to launch and for the **Edit scheduling** window to appear.
4 Click **Edit schedulers** to configure the scenario schedulers.

Click **Edit scheduling** to define the schedulings associated with each connector.

For more information about how to edit connectors, refer to section Creating a schedule [page 173] in this chapter.

Consulting the Connect-It logs

You can consult two logs in the Scenario builder:
- Connect-It log

  This log describes all actions occurring in Connect-It at the startup of a scenario. For example: the deserializing of the scenario, the connection of the connectors to their external applications, etc.
- Document log

  This log enables you to view the detail of documents produced and consumed by the tools in your scenario. It also enables you to identify the root of certain problems that can occur during document processing.

To display the tabs corresponding to these logs in the Scenario builder:
1 Select the service corresponding to your scenario.
2 Click **Monitors**.

The Scenario builder is launched with the diagram of your scenario and Manage saved sessions window is displayed.

> Note:
>
> You may not modify the list of sessions, which is read-only.

Consulting the log file

Every time a Connect-It service is launched, a LOG file is available.

By default, this file is named after your service (For example: `test.log`). To consult the LOG file:

1 Select the service corresponding to your scenario.

2 Click **Log file**.

  Your scenario's LOG file will open in the application associated to the `.log` files in your operating system.

  > ✎ Note:
  >
  > You may specify another name for the log file of your service (**Log file** field)

The information concerning the configuration of the service are saved in the `cit.svc` file (*SvcLogFile* settings).

Optimizing your scenario's performances

This section explains how to optimize the performances of a scenario by reducing the time it takes for the scenario's tools to process documents.

Evaluating document-processing times using statistics

The Connect-It logs, which you can consult in the graphical interface of the Scenario builder or in the log files, provide statistics about:

- The time it takes the source connectors to produce documents.
- The time it takes the mapping boxes to transform documents.
- The time it takes the destination connectors to consume documents.
- The number of rejected documents.

  Example: The statistics for the Asset Management connector inform you that the documents corresponding to the records in the Products table were not processed.

  For additional information on statistics and how to activate them in monitors, see chapter Implementing an integration scenario [page 55], section Define the supervision monitors [page 151].

To obtain statistics on how a tool processes data

1 Open a scenario

2 Configure the monitors (**Monitors/ Configure monitors** menu)

3 Open the scenario connectors (**Ctrl + F4**).

4 Select the scenario's source connector.

5   Click ▶ (**F5**).

6   Position your pointer on a source connector, a destination connector or a mapping box.

A contextual window appears containing a **Statistics** section.

📝 Note:

This data is also available in the Connect-It log tab, which is located under the scenario diagram.

Analyzing the statistics

The following data - displayed in the Connect-It log - represents the statistics of a scenario using an Action Request System connector as its source, an LDAP connector as its destination, and a mapping box:

```
Statistics for the 'Action Request System (fdcitsrv01)' connector  (sessio
n: 26min  57.310s / API: 26min  04.550s)
Document(s) consumed: 7143
Document(s) rejected: 1
Records(s) inserted: 1199
Records(s) updated: 5943
Statistics for the 'LDAP (mail-sd.Hewlett-Packard.com)' connector (session
: 47.628s / API: 35.765s)
Document(s) produced: 7143
Statistics for the 'Mapping (Basic engine)' connector (Session: 01.404s)
Script(s) analyzed: 14286
Document(s) consumed: 7143
Document(s) produced: 7143
```

The following is an analysis of the produced statistics:

■   Document-processing time of the Action Request System connector = 27 minutes, or *4.4* documents per second.

The processing time breaks down into 26 minutes for the APIs concerned by Action Request System and 1 minute for processing in Connect-It.

APIs include the network response time, the execution of 'commits', respecting the database integrity rules, etc.

■   Document-processing time of the LDAP connector = 48 seconds, or *150* documents per second.

The processing time breaks down into 36 seconds for the LDAP API and 12 seconds for processing in Connect-It.

■   Document-processing time of the mapping box = 2 seconds, or *5100* documents per second.

This duration corresponds to the document-processing time in Connect-It.

Examples of document-processing speeds

Here are a few examples of different document-processing speeds:

- Database connector

  1500 to 2000 documents/second.
- ServiceCenter connector

  450 documents/second
- Asset Management connector

  400 documents/second

Improving the production of documents

Improving a connector's document production involves obtaining the data from a source application as rapidly as possible.

This section explains the different ways you can reduce the time it takes to produce documents.

Optimizing the server of the source data

The production of documents depends largely on the use of the source data's server, as well as its connection.

Connect-It does not use large amounts of resources for production. It does not perform many conversions or data processing during production.

To improve the results, verify:

- The amount of time the server takes to load the source data.
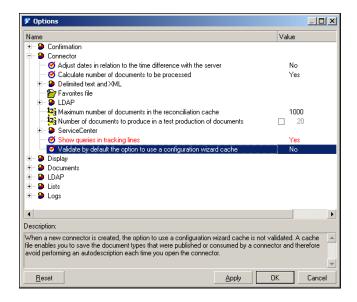
  This amount of time varies depending on the server's technical capacities: processor, available memory, etc.
- The network connection (WAN, LAN).

Checking queries executed by the connector

To track the queries transmitted to the source database:

1 Launch the Scenario builder.
2 Select the **Edit/ Options** menu.
3 Unfold the **Connector** node.
4 Set the option **Show queries in tracking lines** to *Yes*.

5 Click **OK**.

After you configure this option, the Document log will display messages corresponding to the queries executed by your scenario's source connectors.

Example:

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

This information enables you to:
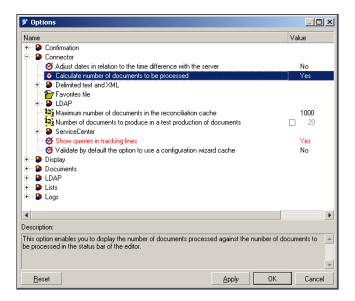
- Verify that the request is generated by conforming to the defined document type and the entered production directives.
- Relaunch the query using another tool and observe the time it takes to execute it.

Calculating the number of documents to produce

To calculate the number of documents that a source connector must produce:

1 Launch the Scenario builder.
2 Select the **Edit/ Options** menu.
3 Unfold the **Connector** node.
4 Set the option **Calculate number of documents to be processed** to *Yes*.

5 Click **OK**.

This option enables you to display a progress bar in the Scenario builder. This bar will indicate the number of documents processed by the source connector.

Figure 5.13. Data processing - Progress bar

To an Asset Management connector, this option triggers a SELECT COUNT type query.

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

Example:

Warning

You must unselect this option when you put your scenario into production phase, because this query takes a lot of time.

Using the connector cache option

For Database-type connectors, we recommend that you use a cache for the structure of the published document types (metadata).

The cache allows Database-type connectors to open more quickly since the published document-type description (metadata) is loaded locally.

For more information about using connector caches, refer to the *Connectors* guide, chapter *Configuring connectors*, section *Configuring the cache*.

If a connector does not use a cache, it then must obtain a data set corresponding to the source database description every time it opens.

For Database-type connectors, this description data includes the:

- List of tables.
- List of fields.
- List of forms.
- List of indexes.
- List of links.
- List of joins.
- Etc.

The more of this data there is, the longer the connector will take to open.

📝 Note:

The time a connector takes to open is also slowed down when the available bandwidth is weak. This is linked to the performance of the network connection (WAN, LAN).

Warning

We recommend using a cache when you put a scenario into production.

However, if the description data of the source database changes, you must synchronize your connector's change.

To synchronize a connector's cache:

1 Open the Scenario builder.
2 Open your scenario.
3 Select the connector whose cache you want to synchronize.
4 Open the connector (F4).
5 Select the **Tools/Cache/Synchronize the cache** menu.

After synchronizing the cache, make sure that your mappings do not contain items that no longer exist in the new source database description.

Automatic reconnection

If you plan on using the connectors whose data source is a remote sever, you must select the automatic server connection option.

Automatic reconnection options have an impact on the network's latency: data-packet transmission.

For more information about the automatic reconnection option, refer to the *Connectors* guide, chapter *Configuring connectors*, section *Configure the reconnection parameters*.

The reconnection information is displayed in the Connect-It log.

Example of the Action Request System connector's attempt to reconnect:

```
Connecting to the server...
Using the dynamic library 'arapi50.dll'.
(ARS ERROR 90) Cannot establish a network connection with the AR System se
rver. fcitsrv01: RPC: Name to address translation failed - No such hostnam
e
The next attempt to reconnect is in 4 s
Attempting to reconnect...
Using the dynamic library 'arapi50.dll'.
(ARS ERROR 90) Cannot establish a network connection with the AR System se
rver. fcitsrv01 : RPC: Name to address translation failed - No such hostna
me
The next attempt to reconnect is in 8 s
Attempting to reconnect...
Using the dynamic library 'arapi50.dll'.
(ARS ERROR 90) Cannot establish a network connection with the AR System se
rver. fcitsrv01 : RPC: Name to address translation failed - No such hostna
me
The next attempt to reconnect is in 16 s
```

This can take a long time if you have an unstable connection to the network.

You must pay special attention to certain connection configurations such as *Router/Firewall/Server*, which close all connections open longer than a given duration. These parameters should be verified by someone in charge of your local network.

Reconnect at each session (specific to the Asset Management connector)

A configuration option specific to the Asset Management connector enables you to close the connection at the end of each scenario session and to reconnect only when you open a new session.

In effect, the less resources the Connect-It server uses, the more rapid the reconnection operations to the server will be.

For more information about this option, refer to the *Connectors* guide, chapter *Hewlett-Packard connectors*, section *Asset Management connector/Configure the Asset Management connector*.

Warning

This option must be unselected if the scenario is executed frequently (example: every five minutes).

For more information about the frequency of a scenario's execution (using schedulers), refer to the chapter Managing an integration scenario [page 191], section Creating a schedule [page 173].

Optimizing the use of the WHERE clause

The Database-type connectors have production directives that enable you to enter WHERE clauses.

For more information on entering WHERE clauses, refer to the *Connectors* guide, chapter *Connector directives*, section *Production directives/WHERE and ORDER BY clauses*.

To optimize the use of your WHERE clauses, verify that the fields concerning these clauses are indexed. If a WHERE clause concerns a non-indexed field, then executing this clause will verify all the fields in this table.

Schedule pointer management

You can use pointers to reduce the quantity of data processed by a scenario.

A pointer corresponding to a record's last-modification date enables a connector to only process the records created or modified since the last time the connector was launched.

Example: A source connector only processes the records from an Employees table corresponding to the newly hired employees.

For more information about using pointers, refer to the *Connectors* guide, chapter *Configuring connectors*, section *Configuring the schedule pointers*.

In most cases, the status of a pointer refers to a specific date and time, which you can modify in a scenario's scheduling window (**Scenario/Scheduling** menu).

If the scenario's pointer uses a non-indexed field, it will take a long time to process data. In order to avoid this inconvenience, we recommend indexing the field as your pointer.

Improving the production of documents

Improving a connector's document consumption involves processing the data to be sent to a destination application as rapidly as possible.

This section explains the different ways you can reduce the time it takes to consume documents.

Choosing reconciliation keys

Reconciliation involves matching old and new data. In the case of a Database-type connector (Asset Management), reconciliation involves defining the fields that enable you to uniquely identify the records in the tables that the connectors create or update.

The field you use as a reconciliation key will greatly influence the consumption of documents by a destination connector.

Consuming documents involves the following two actions:

1 Sending a query that uses the fields selected as reconciliation keys to learn if the record exists in the destination application.
2 Inserting or updating the record.

If the fields selected as reconciliation keys are not indexed, it will take a long time to consume documents. Example: If you select the **memo** or **FeatParam** fields as reconciliation keys for the Asset Management connector, document consumption will be very inefficient.

For further information on reconciliation keys, refer to the section entitled Reconciliation keys [page 114] of this chapter.

Transaction management

By default, the Database-type connectors perform a 'commit' each time they consume a document.

If the bandwidth you have is not very large, you can perform a 'commit' by group of documents.

To set the number of documents to consume before committing them:

1 Open the Scenario builder.
2 Open your scenario.
3 Select the connector for which you want to modify the transaction parameters.
4 Launch the connector configuration wizard (F2).

   Attention: Verify that you are in the advanced configuration mode (icon is selected in the toolbar).
5 Click **Next** several times until you come to the **Manage transactions** page.
6 Select the **Commit by group of documents** option.
7 Indicate how many documents should be consumed before a 'commit' is triggered.

When a large number of documents is consumed, we recommend that you launch the 'commit by group' function and enter the number of documents you are about to have consumed.

Examples of improving performances of the scenario ldap\ac\complete.scn:

1 Commit after each document (default option)

   Performance: Four minutes for 6,000 documents, or 25 documents per second.
2 Commit by group of 500 documents

   Performance: Three minutes for 6,000 documents, or 33 documents per second.

Synchronous processing vs. asynchronous processing - ServiceCenter connector

Each document consumed by a ServiceCenter connector corresponds to a query sent to a destination application.

To improve the time it takes to consume documents, you can choose from:

- Synchronous data processing.

Each consumed document is sent once the previous document has been processed by the destination application.

- Asynchronous data processing.

Each consumed document is sent even if the previous document has not been consumed by the destination application.

Improving document processing times of the Connect-It engine

This section explains what you can do to reduce the time it takes for the Connect-It engine to process documents.

Configuring the Document log

Configuring Connect-It's Document log enables you to indicate:
- The error type that you want to obtain in the Document log (**Filter** field).
- The maximum number of tracking lines that Connect-It saves in memory.
- A text file in which the messages from the Document log will be saved.

To improve the time it takes to process documents:
1 Launch the Scenario builder.
2 Select **Monitors/ Configure monitors/ General settings** and disable in order of priority :
    - Database
    - Files
3 Limit the error type that you want to save in the Document log by selecting a sub-options for the **Save documents** option.

   Example: Reject
4 Limit the number of tracking lines to be saved (**Document log** tab).

Using the non-graphical interface

If the Scenario builder is displayed while the scenario is running, the Connect-It engine will react more slowly each time you choose a command: refresh, consult the document log, enter a new option, etc.

Setting the scheduling

Setting the Connect-It scheduling function involves associating one or more schedulers to document types produced by a scenario. Example: A scheduler 'wakes up' the Asset Management connector every hour. Each time it wakes up, the connector produces documents corresponding to records in the Asset Management application it consults.

Each time the connector wakes up, Connect-It calculates how much data needs to be processed and does not stop until all the data has been processed. You

must set your schedulers depending on how much data there is to process. When migrating large database, we recommend that you program your scheduler to wake up the connector during periods of little or no activity (at night, for example). For integration scenarios that will process less data, we recommend using the predefined **Synchronous** scheduler (which wakes up the connector every second).

For more information about scheduling, refer to the chapter Creating a schedule [page 173].

Launching parallel scenarios

When one scenario must migrate large amounts of data, you sometimes need to divide it into several scenarios and launch each one on a different service (Windows) or daemon (Unix).

For more information about services and daemons, refer to this chapter's section Define a Connect-It service (Windows environment) [page 167].

Example

You need a scenario to import the list of employees recorded in one database to another database.

To improve the time it takes to migrate this data, you can create two scenarios:

- The first scenario migrates the list of employees whose name starts with a letter between A and J.
- The second scenario migrates the list of employees whose name starts with a letter between K and Z.

The selection of employees is done when you enter production directives (WHERE clause) for your two scenarios' source connectors.

For more information about entering WHERE clauses, refer to the *Connectors* guide, chapter *Connector directives*, section *Production directives/WHERE and ORDER BY clauses*.

Improving performances of a scenario using the Asset Management connector

This section explains the different ways you can reduce the time it takes to process documents in a scenario using the Asset Management connector.

Indexing the **dtLastModif** field

In a scenario using the Asset Management connector, you must add indexes to the **dtLastModif** field in all AssetCenter tables covered by the scenario. This field is also used systematically by Connect-It to verify the records created or modified since the last session.

To create an index on the Assets table (amAsset), Products table (amProduct) or Employees and Departments table (amEmplDept), execute the following command:

```
CREATE INDEX Ast_dtLastModif ON amAsset (dtLastModif)
GO
CREATE INDEX Prod_dtLastModif ON amProduct (dtLastModif)
GO
CREATE INDEX EmplDept_dtLastModif ON amEmplDept (dtLastModif)
GO
```

To verify that the **dtLastModif** fields were all indexed, you can use the adblog executable. This executable enables you to verify if the SQL queries that have a filter on the **dtLastModif** field, in the WHERE clause, was executed.

Settings for an Asset Management database - Sybase ASE engine

If the LOG files indicate that certain queries are taking time because they have numerous tables in the FROM part, we recommend that you use the Sybase Query Optimizer application to reduce this processing time.

Modifying the `amdb.ini` file

To reduce the time it takes to process SQL queries, you must also modify the `amdb.ini` file in your Asset Management application by adding the following line:

```
PostConnectSql= set forceplan on
```

The following lines show the configuration of the `amdb.ini` file for a database called *DB ASE COPPER*:

```
[DB ASE COPPER]
PostConnectSql=set forceplan on
stmtcache=500
LongDesc=
Engine=Sybase
Location=COPPER
EngineLogin=itam
EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC641ED76CDAA17
050EA1A124254200200
ReadOnly=0
CacheDir=
CacheSize=5120000
Base=EDS
Owner=
TableSpace=
TableSpaceIndex=
AmApiDll=aamapi35.dll
UseNTSecurity=0
```

These parameters must be configured in the `amdb.ini` file on the Asset Management database server.

If the client part of your Asset Management application is installed on your Connect-It server, you can establish two different connections linked to the same Sybase database:

■ The first connection uses the `PostConnectSql=set forceplan on` and `stmtcache=500` options.

```
[DB ASE ConnectIt]
PostConnectSql=set forceplan on
stmtcache=500
LongDesc=
Engine=Sybase
Location=COPPER
EngineLogin=itam
EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC641ED76CDA
A17050EA1A124254200200
ReadOnly=0
CacheDir=
CacheSize=5120000
Base=EDS
Owner=
TableSpace=
TableSpaceIndex=
AmApiDll=aamapi35.dll
UseNTSecurity=0
```

■ The second connection does not use either of these parameters.

```
[DB ASE COPPER ACGUI]
LongDesc=
Engine=Sybase
Location=COPPER
EngineLogin=itam
EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC641ED76CDA
A17050EA1A124254200200
ReadOnly=0
CacheDir=
CacheSize=5120000
Base=EDS
Owner=
TableSpace=
TableSpaceIndex=
AmApiDll=aamapi35.dll
UseNTSecurity=0
```

For the classic AssetCenter clients (other than the Sybase ASE clients), the `stmtcache=500` option must not be used.

If you encounter performance problems on the classic AssetCenter clients, you can try one of the following options:

■ PostConnectSql=set forceplan on

■ PostConnectSql=set table count 3

■ PostConnectSql=set table count 2

Improving performances of a scenario using a Database type connector

For the Database type connectors using a native Sybase connection, you can improve the execution of SQL queries by entering the `PostConnectSql=set forceplan on` option in the advanced options.

For more information about advanced options, refer to the *Connectors* guide, chapter *Configuring connectors*, section *Advanced configuration/Advanced options*.

6   Monitors database administration

The information related to the processing of your data may be saved in a database.

Data is backed up according to the parameters that are defined for the monitors (refer to section Define the supervision monitors [page 151] in this guide).

This database must first be created in the form of an empty shell.

A demonstration database in MSDE format is shipped with the application.

Supported DBMSs

The following DBMSs are supported for the Connect-It database:

- Microsoft SQL Server
- Oracle Workgroup Server
- Sybase Adaptive Server
- IBM DB2 UDB

Steps in configuring the Connect-It database

Adhere to the following steps in order to save your information in a database:

- Create an empty shell for your database

- Declare a connection to this database
- Create the Connect-It database
- Add the users

Note:

A demonstration database for Connect-It is available in the dbb folder of the Connect-It installation folder. The administrator is declared without a password.

▸ MSDE monitors database [page 18].

Creating an empty shell

This step depends on the database engine used. Refer to the documentation of your DBMS.

Declaring a connection

To declare a connection:
1 Declare the connection at the database manager level
2 Use the database administration console (**Administration/ Monitors database administration**)
3 Populate the **Connection** field with an existing connection from the list

Note:

If no connection has been declared, you can create a new one by clicking ▦

Creating a connection

To create a connection to the database:
- In the **Connection** field, click ▦
- In the connections editor, click **New**
- Populate the fields required to declare the connection
- Click **Create**.

Connection declared at the level of the .ini file or ODBC

When creating a new connection (**New** button), the information contained in the fields of the **Connection** tab (**Name**, **Description**, **Engine**) are saved in the citdb.ini file. This file describes the connections used by Connect-It.

The physical database connection is declared in the **Data source** (or **Server**), **Database**, **User** and **Owner** fields. If the data source is not in the list of available connections, you can declare a new connection (▣ button).

✎ Note:

When you edit the connection, the password field is emptied. When editing the connection, you must know the password of the ODBC connection.

System connection

This option enables you to define the behavior of the connection:

- Option selected: The connection is created in the `citdb.ini` file of the Windows folder
- Option not selected: The connection is created in the `citdb.ini` file of the user preferences folder.

  In the case where this option is not selected, the behavior is the following:

  - If a user B uses a scenario used by user A, user A must declare the same connection again because they will not be able to read the user A's `citdb.ini` file because of the user rights
  - When executing a scenario is service mode, a service is declared by default in system mode. With this mode, the service can only read system connections.

    ✎ Note:

    You may force a service to log on as a given user (Windows Services utility).

Location of .ini files

All connections created in the database administration window are saved in the `citdb.ini` file.

System connections are saved in the `citdb.ini` file of the Windows folder.

The user connection is saved in the `citdb.ini` file in the user preferences folder of Windows.

✎ Note:

For a quick overview of where the .ini files are located, refer to the information available for theapplication (**Help/ About** menu, **Plus** button).

Creating a database

Creating a Connect-It database means populating an empty shell with the tables that will hold the information related to the processing of your documents by the connectors.
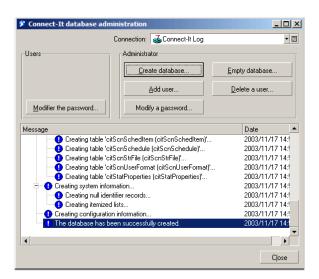
To create a Connect-It database:

1 Use the database administration console (**Administration/ Monitors database administration**)

2 Click **Create database** and populate the following fields:

▪ **Administrator password**: Enter the administrator password of the database

✎ Note:

You may not leave this field empty

▪ **Confirm administrator password**: Enter the password again

The database is created, with the administrator as the first user.

Adding a user

To add users to the database:

1 Use the database administration console (**Administration/ Monitors database administration**)

2 Click **Add user** and populate the following fields:

- **Administrator password**: Enter the password you defined when creating the database
- **User**: Name of the new user
- **New password**: Password for the new user
- **Confirm password**: Confirm the password

3 Click **Add** to add this user and then click **Close**

Note:

The created users do not have administration rights for the database.

Administering users

All operations concerning the database users are performed via the **Administration/ Monitors database administration** menu item.

Managing database users involves:
- Adding a user
- Changing their password
- Deleting a user

Adding a user

Refer to section Adding a user [page 264] of this chapter.

Changing the password of a user

To change the password of a user:
1 Click **Modify password**
2 Select the user from the list (**User** field)
3 Enter the new password
4 Confirm the password
5 Click **Modify**

Deleting a user

To delete a user:
1 Click **Delete a user**

2   Enter your administrator password
3   Select the user from the list (**User** field)
4   Click **Delete**.

Operations performed by a user

Once a user is declared in the database, they can:

◆ Modify their user password

To modify the user password:

1   Use the database administration console (**Administration/ Monitors database administration**)
2   In the **Users** frame, click **Modify password**
3   Populate the following fields:
    ▪ **User**: Name of the user
    ▪ **Previous password**: Enter the current password of the user
    ▪ **New password**: Enter your new password
    ▪ **Confirm new password**: Enter the new password a second time
4   Click **OK**

Operations on the database

Apart from user-administration operations on the Connect-It database, you can also delete all the information in the database.

This function may only be used by the database administrator.

To delete the data from the Connect-It database:

1   Click **Empty database**
2   Enter your administrator password.
    The data is erased

Backwards compatibility

Version 3.4 and version 3.3.x databases are incompatible.

A How to signal a problem

To help us in our aim to constantly improve our products, we strongly encourage you to contact Hewlett-Packard technical support whenever you have a problem using one of our products. In order to identify and solve these problems in the least amount of time, we will need to collect a certain amount of information that will allow us to reproduce and then solve the problem.

The objective of this document is to help you gather the necessary information to provide us and how to report the problem in the most effective manner possible.

General information

Here is the information that we need:

- Functional information [page 267]
- Technical information [page 268]

Functional information

This information enables us to categorize the problem:

- The name of your company.
- The name of the product causing you trouble.
- The functional domain of the problem. For example, the Scenario builder in Connect-It, or the Procurement management feature in AssetCenter, etc.

- The frequency in which the problem occurs: The problem could occur systematically; it could depend on certain things; it only deals with a certain kind of data, etc.
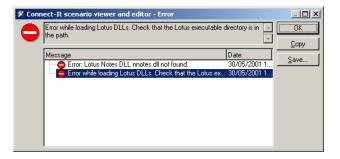
Technical information

Some essential technical information is directly accessible in our products. To find this information:

1 Select the **Help/About** menu in the software.

2 In the dialog box that appears, click **More**. A list of technical information about your system and the product will appear.

3 Click **Copy**. The information will be copied to the Window clipboard.

4 Paste this information in a new file and send it to Hewlett-Packard technical support.

If the problem that you are reporting also has an error message that occurs in the software, perform the following steps:

1 When the error message appears (see the example below), click **Copy**. The information relating to the problem will be copied to the Windows clipboard.

Note:

This screen capture is given as an example. The information in this error message, as well as the name of the software, will probably be different in your case.

2 Paste this information in a new file and send it to Hewlett-Packard technical support.

Specific information about the problem

The first thing we need is a contextual list of steps that enable us to reproduce the problem. Since the problem could depend on any number of factures, please be as precise as possible in your description. For example:

1 Select the Hewlett-Packard Prosigma 850 in the Assets table in AssetCenter. This asset is part of the demonstration database provided with the product.
2 Click on the **Acquis** tab of this asset's detail.
3 Click on the **Procurement** sub-tab of the **Acquis** tab.
4 Change the value of the **Purchase value** field to 123.45.
5 Click **Modify**.
6 Etc.

> 💡 Tip:
>
> As a general rule, you should describe the problem as thoroughly as possible.

Other useful information

If you wish, you can send any additional screen shots or files that might help us identify the problem.

> 📝 Note:
>
> Do not, however, send extremely large files via e-mail. Try to set a limit of about 500 KB for the attachments. If necessary, our support team can open an ftp access to read files going over 500 KB.

Here are some tips to limit the size of your files:

- Try using a compression tool. In most cases, we notice a dramatic decrease in the size of files when these tools are used.
- You can reduce the number of colors in your images to 16; the image will then take up less space.

If you signal a problem about Connect-It

Connect-It systematically keeps document logs, application logs and service logs. The information in these files can be very useful in helping to reproduce the problem.

These files are stored in the following directories:

- `bin\conitgui.log`: application-log file

- `bin\<Name of the service>.log`: Connect-It log file

> ✏ **Note:**
>
> Try to send us these files along with the scenario file (`.scn`) when you report a problem on Connect-It.

B Glossary

The following glossary provides definitions of the main terms used in Connect-It. The glossary also provides a list of terms relating to databases, Basic scripts, Java and external applications.

Connect-It terms

Action after processing

Certain connectors, such as the OpenView Desktop Discovery connector, can manipulate the data that they have produced. In this case, a connector-configuration wizard page enables you to select post-processing actions for the produced documents. For example: Delete source data after successful processing in Connect-It..

External application

An external application designates not only the application itself, but also the data source to which the connector connects. Examples: an Asset Management application, an XML file, a messaging system.

Processing reports

Processing reports are documents that the connectors or the mapping box produce each time they process a document. These reports enable you to know if the document was processed successfully or not, and whether or not any problems were encountered in Connect-It.

Example :

The e-mail connector can consume scenario processing reports and send them to the Connect-It administrator in the form of messages.

Authorization certificate

The authorization certificate is a text file that enables you to activate the base connectors, as well as the optional and additional connectors, included in the Connect-It package that you have chosen.

The certificate is required in order to use Connect-It.

Tool

A tool (or component) is an element defined in the Scenario builder. It may be a connector or an internal tool, such as a mapping box or a data generator.

Reconciliation key

Reconciliation consists of defining the fields that enable the unique identification of the records in the tables that a connector creates or updates.

Fields used for data reconciliation are identified by marking them with a key, called the reconciliation key.

Connector

A connector enables Connect-It to process information coming from or going to external applications. In order to exchange data between themselves, connectors copy this data into documents that they can product or consume.

The connectors are divided up into several categories:

- Inventory connectors

  Inventory connectors process information from databases that scan a company's assets, and more specifically, their IT assets: Intel LanDesk connector, SMS 1.x connector, SMS 2.x connector, etc.
- HP OpenView connectors

These connectors are specifically designed for the HP OpenView applications (Asset Management connector, ServiceCenter connector, OpenView Desktop Discovery connector, Action Request System...)

- Application connectors

  These connectors are dedicated to specific external applications: Lotus Notes connector, NT Security connector, etc.

- Protocol connectors

  These connectors process data formatted by external applications using a specific protocol: XML connector, Delimited text connector, Database connector, **E-mail** connector, etc.

- ERP Connectors (Enterprise Resource Planning)

  These connectors enable you to process data from Enterprise Resource Planning applications.

The role of a connector is to:

- Publish document types corresponding to available data containers in the external application to which it is connected.

  Example :

  The e-mail connector (fetching) publishes a document type that corresponds to an e-mail, including such things as a **From** structure, a **Carbon copy** collection, and an **Attachment** collection.

- Produce or consume documents.

  A connector uses a produced document-type that was created for a specific integration scenario, and from this template, it creates the produced document.

  Example :

  In each of the **Enterprise Discovery to Asset Management** out-of-the-box scenarios (scenario/pdi folder), the Enterprise Discovery connector produces documents that use the **Devices** document type as a template. This document type corresponds to the values obtained during the IT inventory scan performed by Enterprise Discovery.

  To consume a document, a connector extracts the values from a document leaving a mapping box and transmits them to an external application.

  ▶ Consumption [page 274].

When data comes from an external source application and goes to an external destination application, the connectors involved in the scenario are called the

source connector and the destination connector. A source connector produces documents, whereas the destination connector consumes them.

Collection

In a document type, a collection represents the relation between one or more elements to several other elements.

Example: When a document type corresponding to a database table has a collection, this collection corresponds to another table in the database. A collection implies that the link between these two tables is a 1-n or n-n link. This type of link means that a record in the first table can be linked to one or more records in the second table, and vice versa.

Example :

In a document type corresponding to a sent e-mail message, a collection corresponds to the message recipient (a message can have several recipients).

Service Console

The Connect-It console is a graphical user interface that lets you manage your scenarios and create services to link to them.

Consumption

Consumption is the process in which a destination connector writes or sends documents to an external application.

▶ Connector [page 272].

DBK

The DBK files are XML files that correspond to a scenario's documentation. These files conform to the DocBook DTD. For more information about this file, refer to the chapter Scenario documentation [page 185].

Directives

Directives are pieces of information that the Connect-It tools use to produce and consume documents. The directives are categorized by *production directives* and *consumption directives*.

Example :

For a Database-type connector, the WHERE and ORDERBY clauses, which are used to filter records from a source database, are the consumption directives.

To learn more about directives, refer to the *Connectors* guide, chapter *Connector directives*.

Document and document type
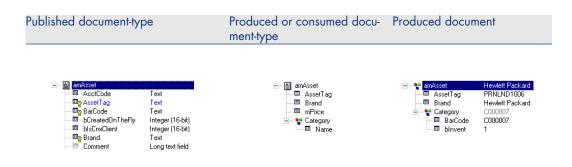
In Connect-It data is exchanged via documents. Each document corresponds to a data container in an external application. Depending on the function of the external application, a data container corresponds to:

- Database tables
- E-mail messages
- Files containing scanned information
- Delimited text files
- XML files
- Security information
- Etc.

When a connector is open in the Scenario builder, it publishes a list of document types. This list of published document-types corresponds to the different data containers available in the external applications.

Produced and consumed document types

In order for a connector to produce or consume documents, it needs to use produced or consumed document-types. To create a produced or consumed document-type, the Connect-It user selects elements from a document type published by the connector. For example: A document type produced by a database connector contains the only fields that the user wants to import in the external destination application.

Published document-type	Produced or consumed document-type	Produced document

Published document-type	Produced or consumed document-type	Produced document
The connector publishes document types. Here is an excerpt of the amAsset document type published by the Asset Management connector.	Certain elements from a published document-type are selected to create a produced or consumed document-type.	The connector produces or consumes document instances.

DSC

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 Delimited-text connector publishes 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.

▶ Connect-It Guide - Connectors, Protocol Connectors, Delimited-text connector, Creation of the description file.

Documentation

All the information extracted dynamically from a scenario and that can be customized by users.

Scenario builder

The Scenario builder is the graphical interface that enables you to create integration scenarios. These scenarios are identifiable by the SCN file extension.

Element

A document type is made of the following elements:
- Structure
- Collection
- Field

  This element is the terminal element of a document type.

These elements are also called complex elements.

▶ Document type elements [page 75].

▶ Collection [page 274].

Favorites

Favorites are preconfigured connectors.

▶ Favorites [page 207].

Configuration file

Each connector publishes a document type in `.xml` format which is an interpretation of a database structure. This document is called a configuration file; it can be edited (using the configuration file editor) for any inventory connector and for the database connector.

User formats

User formats are date-type or numeric-type formats that the user can define for use with the mapping scripts. These formats can be used exclusively with the **PifUserFmtVarToStr** and **PifUserFmtStrToVar** functions. To use these functions, refer to the Programmer's Reference available online by pressing F1 whenever you write a script in Connect-It.

For more information about user formats, refer to the chapter Define mapping scripts [page 117], section User formats [page 131].

Data generator

The data generator is a tool for sending statically-defined data to a connector.

▶ Using the data generator [page 82].

History

All the information about the creation and modification of a scenario.

▶ Scenario history [page 212].
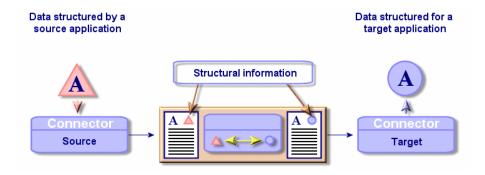
Document log

The Document log is a tab in the Scenario builder that helps you follow up on all the documents produced and consumed by the connectors and the other Connect-It tools.

When a connector rejects a document, the rejected document appears in the Document log along with an icon indicating its status. The Document log also warns you when a the mapping box could not perfectly produce or consume a document.

Mapping

A mapping involves creating correspondences and transformations between elements from a document type that is produced by a source connector and one that is consumed by a destination connector.

In Connect-It, a mapping box enables a destination connector to consume documents produced by the source connector. The mapping box is an internal tool that produces and consumes documents like all other connectors, but which does not communicate with external applications.

▶ Define the mapping of document types [page 91].

Monitors

Monitors are surveillance tools associated with a scenario. Monitors activate and manage the forwarding of information and process history data provided by the application and its connectors.

▶ Define the supervision monitors [page 151].

Schedule pointer

The schedule pointer enables you to reduce the amount of data processed by a scenario.

A pointer corresponds to the date of a record's last modification. It enables a connector to distinguish data it has already processed in a previous session

from the new data. It therefore only processes the records created or updated since the last session.

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 :

During its last session, an Asset Management connector read all records in the Assets table at 9:00:00. During the next session, it will read all records created after 9:00:00.

Production

Production is the process by which a source connector reads or receives data from an external application in order to transform it into documents.

▶ Connector [page 272].

Scheduler

A scheduler is a timer that you associate with one or several document types produced by your scenario's source connectors.

To edit a scenario's schedulers, select the **Scenario/Schedulers** menu.

Two preconfigured schedulers are provided with the Scenario builder:

- The **Once** scheduler.

  This scheduler is the one associated by default to all the document types produced by a scenario.

- The **Synchronous** scheduler.

  This scheduler enables a connector to produce or consume a document every second.

▶ Schedule pointer [page 73].

Schedules

Scheduling in Connect-It involves indicating at what moment the source connectors of an integration scenario will produce documents. Creating a schedule consists of associating produced documents to one or more schedules.

Each schedule plays a role of a timer that makes a connector produce documents either at one specific time (October 4, 2002) or in regular intervals (every two hours for a given period of time).

▶ Creating a schedule [page 173].

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.

- 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.

The process is based on the following question: "Does the information that I would like to reconcile already exist in the target system?"

When the destination system is empty, this processes is no longer a reconciliation, but an import.

Integration scenario

An integration scenario enables information to be passed between different applications.

Writing a scenario consists of:

1 Choosing the source and destination connectors.

   In principle, you can choose an unlimited number of connectors.

2 Configuring the connectors.
3 Linking the connectors.
4 Creating produced document-types for the source connector and consumed document-types for the destination connector.
5 Creating mappings between the source document-types and destination document-types.
6 Choosing a schedule.

   This schedule establishes the intervals at which the scenario's source connectors produce documents.

Connect-It is provided with several example scenarios.

Warning:

For the most part, the example scenarios do not apply to specific user configurations.

Services

A service under Windows provides an application that runs as a background task. .Connect-It lets you create and launch a different service for each of your scenarios.

Statistics

After each session, the Scenario builder enables you to obtain statistics on the number of documents processed by each tool in a scenario. Example: the number of documents processed by the Asset Management connector during one session.

To obtain the statistics of each connector, position your pointer on the top part of a connector in the scenario diagram.

Structure

A structure in a document type indicates a 1:1 relation with the node containing this structure. A node that contains a structure can be the document type's root node, the structure or the collection.

Example :

In a document type that corresponds to a table in a database, each structure corresponds to another table. A structure indicates that the link between the two tables is 1:1. A link of this type means that a record in the first table can only be linked to one record in the other table.

Maptable

A maptable is a table that maps elements from one container to elements of another data container.

▶ Maptables [page 124].

Views

Views of a scenario diagram can be created in order to make viewing the scenario easier. Example: In a scenario with a dozen or so different tools, you can create a view where only some of those tools are visible.

To learn more about views, refer to the chapter Ergonomics [page 39], section The Scenario builder [page 39]/ Scenario-diagram views [page 43].

Common terms

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.

Blob

Blob-type fields (binary-large-object-type fields) are those fields corresponding to binary data: sound files, video files, image files.

DSE

The DSEs (Directory Entry Services) are the entries making up the hierarchy of the LDAP directory. This information is represented in the form of branches.

Each entry in the LDAP directory corresponds to an abstract or real object (a person, material object, parameters, etc.).

JCA

Perfected by Sun, JCA regroups a set of Java specifications that describe an interface: "Java Connector Architecture".

Its main objective is to define a layer of standards enabling a Java server to dialog with applications based on third-party technologies.

JCA aims to improve connection interfaces and standard connectors.

XML schema

An XML schema is the definition of the contents used in an XML document. The XML schema is a superset of the DTD, which is itself a standard SGML.

Unlike a DTD, the XML schemas are written in XML syntax, which is more verbose than a DTD.

The XML schemas can be created with any XML tool.

Transaction

A transaction is an element from the dialog between a computer an a user or between two computers - such as in an information request and the response.

Global variables

A global variable is a variable whose value can be accessed and modified by any program instruction; not just by the module that defined it.

XSL

XSL (eXtensible StyleSheet Language) is a style sheet language developed especially for XML and which can be extended. It is not recognized as an official standard.

Index

.cfg, 67
.ini
    Declared connection, 262
$LinkedDoc
    Processing reports, 240
$ParentDoc
    Processing reports, 240
$ParentDoc$
    Mapping script, 237
    Using the variable, 228
-activate, 200
-dc, 199
-df, 200
-dm, 199
-forcecnx, 199
-openconnectors, 199
-svc, 200

A

Additional scripts, 150
AIX, 15
    Dynamic libraries, 20
AQL
    Definition, 281
Associated file

Configuring the text editor, 121
    Creating, 119
    Deleting, 119
    Opening, 119
    Text editor, 120
    Validating scripts, 120
Authorization certificate
    Definition, 272
    Entry, 29

B

Blob
    Conversion, 113
    Definition, 282
    Export, 113
    Mapping, 111

C

Cache, 241
Cache files, 241
Character string
    Concatenating, 123 , 123
    Conversion to Blob, 113
Character string tables, 123
Chr(), 123