



Peregrine | Connect-It
Studio



© Copyright 2003 Peregrine Systems, Inc. or its subsidiaries.

All Rights Reserved.

Information contained in this document is proprietary to Peregrine Systems, Incorporated, and may be used or disclosed only with written permission from Peregrine Systems, Inc. This manual, or any part thereof, may not be reproduced without the prior written permission of Peregrine Systems, Inc. This document refers to numerous products by their trade names. In most, if not all, cases these designations are claimed as Trademarks or Registered Trademarks by their respective companies.

Peregrine Systems ® and Connect-It ® are trademarks of Peregrine Systems, Inc. or its subsidiaries.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

This document and the related software described in this manual are supplied under license or nondisclosure agreement and may be used or copied only in accordance with the terms of the agreement. The information in this document is subject to change without notice and does not represent a commitment on the part of Peregrine Systems, Inc. Contact Peregrine Systems, Inc., Customer Support to verify the date of the latest version of this document.

The names of companies and individuals used in the sample database and in examples in the manuals are fictitious and are intended to illustrate the use of the software. Any resemblance to actual companies or individuals, whether past or present, is purely coincidental.

If you need technical support for this product, or would like to request documentation for a product for which you are licensed, contact Peregrine Systems, Inc. Customer Support by email at support@peregrine.com.

If you have comments or suggestions about this documentation, contact Peregrine Systems, Inc. Technical Publications by email at doc_comments@peregrine.com.

This edition applies to version 3.2.1 of the licensed program

Connect-It

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



Table of Contents

Introduction	9
What does Studio do?	9
Who is Studio intended for?	10
How to use this guide	10
Chapter 1. Installation	13
Required environment	13
Installing for the first time	14
Chapter 2. General layout of Studio	17
Introduction to the Studio interface	17
Customizing Studio	18
Working with files	28
Studio workspace	31
Chapter 3. Extending the definition of a connector	41
Presentation of sample data	42
Creating a reference scenario	46
Creating a Studio project linked to the reference scenario	46
Updating the raw schema	49

Definition of customization rules	50
Building and updating customized document-types	71
Testing customized document-type	71
Synchronization between Scenario builder and Studio	73
A. Keyboard shortcuts	77
B. Glossary of terms	79
Build	79
DSD	79
Elements	79
Workspace	80
Studio file	80
Project Explorer	80
Link	80
Project	80
Property	80
Index	81

Studio

List of Figures

2.1. Detail of a toolbar	21
2.2. Studio interface - Options screen	23
2.3. Color selection screen	26
2.4. Explorer window	32
2.5. Window of a project file	33

List of Tables

2.1. General tab	25
2.2. Colors tab	25
2.3. Keywords tab	26
3.1. Customizing document types - sample database	43
3.2. Add or modify a node - wizard pages	51
3.3. Add or modify a link - wizard pages	53
3.4. Create a filter - wizard pages	66



Introduction

PREFACE

What does Studio do?

Studio is a fully customizable graphical editor of configuration files. It is used by various Peregrine Systems applications.

Studio provides you a rich work environment, to which has just been added a wide range of conceptual help tools: multiple views of the same element, contextually based data-entry features, wizards dedicated to executing systematic tasks, etc.

Advanced users can define their own tools and make them available for other users of Studio.

The graphical user interface includes, among other things, a workspace that can be fully tailored to the needs of the user, configurable windows and customizable toolbars. The explorer offers a modular and hierarchical view of all elements composing a project, which makes it easy to navigate and to select and reorganize components by simple drag-and-drop operations.

We developed Studio's capacity to be customized in order to maximize user productivity while minimizing the volume of data entry. All Studio functions are available in just a few clicks of the mouse.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>)

Who is Studio intended for?

Studio is intended for IT technicians in charge of integrating the different applications their company uses.

To use Studio, you should have an expertise in:

- XML
- The application for which you are editing configuration files.

How to use this guide

Chapter Installation

This chapter takes you through the Studio installation process.

Chapter General layout of Studio

This chapter presents the layout of Studio.

Chapter Extending the definition of a connector

This chapter explains how to extend the resource-connector-description XML file. This extension enables you to customize the document types published by this connector.

Annex Keyboard shortcuts

This annex contains the keyboard shortcuts available in Studio.

Annex Glossary of terms

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

Conventions used in this guide

The following conventions are used throughout this guide:

Convention	Description
Code	Example of code or command
Fixed font	DOS command, function parameter or data formatting
...	Portion of code or command omitted
Note: Additional information...	Note for informational purposes
Important: Be careful...	Important information for the user
Tip: A tip for using...	Tip
Warning: Warning	Extremely important information for the user
Object	Graphical interface object in Studio: a menu, menu entry, tab or button.

The following conventions are also used:

- The steps that need to be followed in an established order are presented as a numbered list. For example:
 - 1 First step
 - 2 Second step
 - 3 Third and last step
- All figures and tables are numbered according to the chapter in which they belong and their order of appearance within the chapter. For example, the title of the fourth table of chapter two will be prefixed by **Table 2-4**.

1 Installation

CHAPTER

This chapter explains how to install Studio.

Required environment

Studio uses Java language extensively. To use certain Studio tools, you must install a JRE (Java Runtime Environment) version 1.3.0 or higher, provided with Studio.

 **Tip:**

If you don't have a JRE, you will need to download it from the Sun Microsystems Web site. The version 1.4.1 used for this Studio version can be downloaded from the following address: <http://java.sun.com/j2se>.

Refer to the JRE documentation and release notes for further information on installing and using the JRE.

XWP and WSP file association

A Studio project corresponds to the creation of two types of files:

- An **.xwp** file (eXtended Workbench Project)

This file references all the XML files used by the project.

- A **.wsp** file (WorkSPace)

This file contains the:

- The reference of the **.xwp** file.
- The memory of which XML files were open when the last opened project was closed.

We recommend that you associate these two file name extensions to the Studio executable (**studio.exe**). This way, when you double-click such a file, Studio is automatically launched by the operating system. The designated file is then automatically loaded.

 **Note:**

This association is automatically performed when Connect-It is installed.

To perform this operation:

- 1 Open a Windows explorer.
- 2 Select the **Tools/Folder Options** menu.
- 3 Select the **File Types** tab. The operating system looks for all declared file extensions. If **.adw** does not appear in the list displayed, add it by clicking **New** and populating the **File Extension** field.
- 4 Select **.adw** from the list.
- 5 Click **Load** and select the program associated to the **.adw** files (in this case Studio).
- 6 Click **Apply**.
- 7 Repeat this operation with the **.wsp** file name extension.

Installing for the first time

You can install Studio in Windows.

Studio is part of the Connect-It installation package.

 **Note:**

For more information about installing Connect-It, refer to the chapter **Installation** of the Connect-It installation guide.

To install Connect-It's Studio package

- 1 Launch Add or Remove Programs option from the Windows Control Panel.
- 2 Select Connect-It from the list of installed programs.
- 3 Click **Modify/Delete**.
- 4 Wait for the Connect-It installation program to be launched.
- 5 Select the option that enables you to modify the previous installation by adding or removing a package.
- 6 Click **Continue**.
- 7 Select the Studio package in the customized installation window.
- 8 Select the option **Reinstall the previous packages**.
- 9 Click **Next**.
A window opens showing you the installation's progress bar.
- 10 Click OK when the installation has terminated.

2 | General layout of Studio

CHAPTER

This chapter describes the interface and work area of Studio. It is not necessary to read the entire chapter all at once. Instead, you might find it more useful to refer to it if and when you need information on certain parts of the application's layout.

Introduction to the Studio interface

Studio's user interface enables you to perform numerous tasks - from the most simple, like opening a project, to the most complex, like writing JavaScripts. The interface was designed to facilitate project editing for all user levels.

Studio's interface can be easily customized. This chapter, and this guide in general, explains in detail this interface as it looks when first installed. The Studio version that you are using might differ slightly depending on the functions that you might customized.

Studio's interface is composed of:

- Menus
- Toolbars
- Dockable windows

- Main edit windows that support multiwindowing

Customizing Studio

Studio windows

When working with Studio, you will encounter two types of windows:

- Dockable windows, which are attached to the edges of the workspace.
- Floating windows, which you can move around inside the workspace.

 Note:

Dockable windows take priority over floating windows, and as such, they are always displayed in the foreground.

Dockable windows

Dockable windows are used for the functions of editing, navigation and for the Studio help systems:

- **Project**
- **Help**
- **Advanced information on selection**
- **Messages**

Moving a dockable window

To move a dockable window:

- 1 Click the window's title bar and hold down the mouse button.
- 2 Move the mouse pointer toward the edge of the workspace. Once the window can be docked, Studio will display a rectangle that represents the window's future position.
- 3 Release the mouse button to validate the move. The window docks itself to the chosen location.

 **Tip:**

If several windows are docked on the same edge of the workspace, Studio will optimize the space by resizing each dockable window to fit within the available space.

Resizing a dockable window

You can resize a dockable window by moving the mouse pointer over one of its edges: The pointer then turns into a split pointer. Click and resize the window while holding down the mouse button. Release the button when the window has reached the desired size.

Closing a dockable window

To close a dockable window, click the button in the upper-right corner of the window.

Re-opening a dockable window





To re-open a recently closed dockable window, use the **View** menu.

Main editing windows

Main editing windows are used to view the detail of a project element.

Buttons of the main editing windows

The table below describes the buttons available in the title bar of floating windows:

Button	Function
	Minimizes the window.
	Maximizes the window: The window occupies the entire workspace.
	Closes the window.
	Restores the minimized window's original size. This button is only available for minimized windows.

Moving a main editing window

To move a main editing window:

- 1 Click the window's title bar and hold down the mouse button.
- 2 Move the window to the desired location.

Resizing a main editing window

You can resize a floating window by moving the mouse pointer over one of its edges: The pointer then turns into a split pointer. Click and resize the window while holding down the mouse button. Release the button when the window has reached the desired size.

Organizing floating windows

Studio offers several automatic methods of organization for the workspace. If you work with several windows open at the same time, it can sometimes be difficult to arrange your workspace to optimize the space. The **Windows** menu contains the following organization options:

Organization method	Result
Cascade	Organizes the windows so they overlap: The window in the foreground covers the other windows.
Arrange	Organizes the windows without covering them by creating a mosaic of windows in the workspace.
Reorganize icons	Reorganizes the icons (minimized windows) so that they occupy as little space as possible in the workspace.
Close all	Closes all the open windows.
Minimize all	Minimizes all the open windows.

Toolbar

You can access the main functions of Studio using the toolbar. When the cursor pauses over a toolbar icon, a ToolTip appears containing a brief description of that tool.

Description of a toolbar

Figure 2.1. Detail of a toolbar



A toolbar is composed of:

- A movement handle (represented by two vertical lines).
- A series of tools represented by icons.

Activating and deactivating a toolbar

You can activate or deactivate any toolbar, keeping only the toolbars necessary to carry out any given Studio project. The **View/Toolbars** menu controls the display of the toolbars.

To activate or deactivate a toolbar:

- 1 Select the **View** menu.
- 2 Select the name of the toolbar that you want to activate or deactivate from the given list.


 **Note:**

An active toolbar has a check next to it.

Moving a toolbar

You can dock a toolbar wherever you want inside the different panes of the main window, except for in the workspace.

To move a toolbar:





- Move the cursor over the toolbar's handle () . Click and hold down the mouse button.
- Move the toolbar to the desired location and release the mouse button.

Available toolbars

 **Note:**

This section does not describe each icon of each toolbar; they are presented along with the function with which they are associated.

The following table describes the available toolbars:

Name of the toolbar	Description
Edit	This toolbar contains the standard text and project-element editing tools (such as Copy , Cut , and Paste).
	
Studio	This toolbar enables you to manage the different views of the Studio workspace.
	
Build	This toolbar contains the tools necessary to build a project or one of its elements.
	
Connect-It	This toolbar enables you to synchronize a the Connect-It Scenario Builder with a project dealing with extending a resource connector.
	

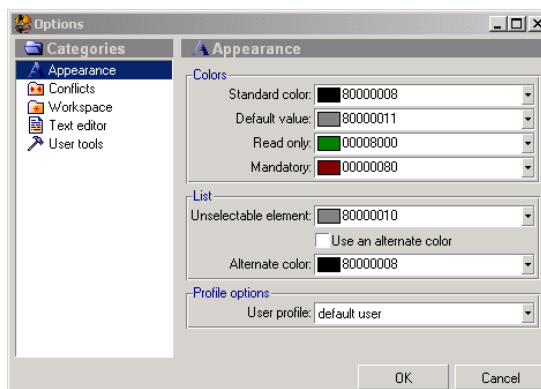
Options

The options giving you access to diverse project parameters. Example: the colors of different text types, their fonts, etc.

You access the options screen by way of the **Tools/ Options** menu.

The options in Studio are independent of the project on which you are working. Each time you change an option, it therefore applies to all projects.

Figure 2.2. Studio interface - Options screen



The options screen is divided into two parts:

- 1 On the left is a hierarchical view of the groups of options that are configurable by the user. These options are classified by category.
- 2 On the right is a list of available options that relate to the group of options selected on the left.

 **Tip:**

All Studio options are saved in the **studio.ini** file. If you use Windows NT, 2000 or XP Professional, this file is located in your **Documents and Settings** folder.

Appearance

This category regroups the options that modify how elements are displayed in a project.

Colors

Option	Description
Normal color	This option enables you to define the default display color for working package elements.
Default value color	This option enables you to define the display color for the labels of options whose values have not been modified by the user.

Option	Description
Read-only color	This option enables you define the display color for read-only properties.
Mandatory color	This option enables you to define the display color for mandatory properties.

List

Option	Description
Non-selectable nodes color	This option enables you to define the display color for the project elements that you cannot select.
Use alternate colors	This option enables you to alternate background colors in lists. If you select this option, Studio displays every other line in the color defined in the Alternate color option.
Alternate color	This option defines this alternate color you want to use.

Profile

Option	Description
User profile	This option enables you to hide certain nodes in a project's tree structure. The more advanced the selected profile is, the more visible nodes you will have.

Conflicts

Selecting the **See all properties** option.

Workspace

This category contains the Studio workspace options. Select the **Load the last opened workspace** option if you want Studio to automatically load the last project (.wsp file) open in the previous session.

Text editor

This category contains the file-editing options in Studio. You can edit three types of files:

- **Basic**
- **HelpOnField**

These files are for the Help on fields of the AssetCenter application.

- HTML
- XML
- Java
- XSL

Choose the file type for which you want to define the options using the drop-down list in the top part of the dialog box. The following editing options are the same for all file types:

Table 2.1. General tab

Option	Description
Font	Select the font used when editing files of this type.
Display line numbers	If you check this option, Studio numbers the lines of the edited text.
Expand tabulations	If you select this option, Studio replaces tabs with spaces in the file being edited. The size of tabs is defined in the Tabulation size option described below in this table.
Auto-indentation	If you select this option, Studio automatically indents text to show, for example, the hierarchy of elements in an XML file.
Tabulation size	This option enables you to define the size of tabs. The unit used is the space.

Table 2.2. Colors tab

Option	Description
Use syntax highlighting	If you select this option, Studio uses the syntax highlighting defined in the Colors tab. Syntax color coding enables you to highlight certain elements (HTML or XML tags, strings, etc.).
Background	Defines the background color in the text editor.
Caret	Defines the color of the cursor.
Comment	Defines the color of comments.
Delimiter	Defines the color of delimiters.
Line number	Defines the color of line numbers.
Numeric value	Defines the color of numeric values.
Operator	Defines the color of operators.
Selected background	Defines the background color of selected text.
Selected text	Defines the color of selected text.

Option	Description
Strings	Defines the color of strings.
Text	Defines the color of text.
Word list 1 to 4	Defines the color of word lists.

 **Tip:**

The colors are displayed in hexadecimal form. You can directly enter a hexadecimal value for the color or you can select the color visually from the drop-down list. If you want to define another color, click the **More** button; Studio will display a color-selection window:

Figure 2.3. Color selection screen

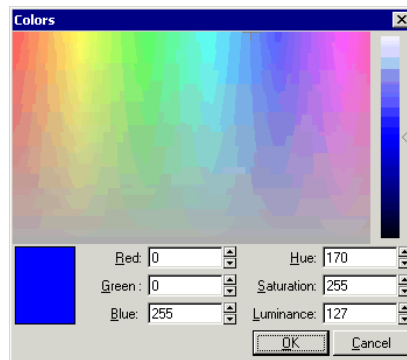


Table 2.3. Keywords tab

Option	Description
Block comments	This option enables you to define the sequence of characters that separate a block of comments. The beginning tag and ending tag are separated by a ' '. The display color is determined by the Comments option in the Colors tab.
Delimiters	This option enables you to define the list of characters used as separators. Each character in this list is separated by a space. The display color is determined by the Word separator option in the Colors tab.

Option	Description
One-line comment	This option enables you to define the sequence of characters that mark the beginning of a comment on a single line (or at the end of the line). The display color is determined by the Comments option of the Colors tab.
Operators	This option enables you to define the list of characters used as operators. Each character in this list is separated by a space. The display color is determined by the Operator option in the Colors tab.
Strings	This option enables you to define the list of characters used to surround strings. Each character in this list is separated by a space. The display color is determined by the String option in the Colors tab.
Word list 1 to 4	This option enables you to define a list of keywords, separated by a space. The display color is determined by the corresponding Word list option (1-4) in the Colors tab.

User tools

This category regroups the tools that the user can customize.

To customize a tool

- 1 Select a tool in the list.

Example: Tool01.

- 2 Populate the following fields:

- **Title**

Example: Notepad.

- **Command**

Example: Notepad.exe.

- **Arguments**

Use this field to specify the arguments for the command indicated in the **Command** field.

In this field, you can use the following variables:

- **%P**

Variable corresponding to the name of a project (full name of the **.xwp** file).

- **%W**

Variable corresponding to the name of the workspace (full name of the **.wsp** file).

- **%F**

Variable corresponding to the name of the **.xml** file open in the active window.

- **%S**

Variable corresponding to the full XML path of the node selected in the open file in the active window.

- **Start in :**

Indicate the path of the folder containing the executable or the command specified in the Command field. **Example: C:\WINDOWS\system32**

- 3 Select the **Use and output window** option so that the executable or the tool's command appear in the window on your workspace.

Example: If a user tool corresponds to a DOS command prompt (**cmd.exe**) with the argument **/c dir**, the list of files returned by the command appears in the output window and not in the DOS command prompt.

Working with files

This section contains information on starting Studio, creating a new project, and changing and saving data in a project.

Starting Studio

To start Studio, refer to the guide of the application with which you use the Studio tool. **Example:** If you use the Studio tool with the Connect-It application, you would start Studio by selecting the **Peregrine\ Connect-It\ Studio** from the Windows **Start\ Programs** menu.

If you have selected the **Load the last opened workspace** option, Studio automatically loads, at startup, the last workspace on which you were working.

Creating a new project

A Studio project is a set of configuration files for a given application: AssetCenter, Connect-It, etc.

For AssetCenter, a project corresponds to editing the database schema.

For Connect-It, a project corresponds to extending document types published by resource connectors. For a full explanation of resource connectors, refer to this guide's chapter [Extending the definition of a connector](#) [page 41].

To configure a new project

- 1 Select the **File/ New** menu.
- 2 Select the **Projects** tab.
- 3 Select a project type from the list of models in the **Projects** tab.
Example: Connect-It project
- 4 Name your file.
This file name must not contain any periods. The project is saved with the extension **.xwp**.
- 5 Specify the folder in which your project is saved in the **Folder** field.
- 6 You can select the **Create workspace** option or you can leave it unselected.
If this option is selected, a .wsp file with the value of the **Name** field is created in the project folder. We recommend leaving this option selected.
- 7 Click **OK**.

Creating a new file (project element)

Each project is composed of several elements (XML files). These files are displayed in the **Project** pane of the Studio workspace.

These elements (files) differ depending on the application with which you use the Studio tool. Example: With Connect-It, the file elements are XML files saved by default in the **cntrs** sub-folder of the project folder.

To create a new XML file

- 1 Select the **File/ New** menu.
- 2 Select the **Files** tab.
- 3 Select a file type from the list of models in the **Files** tab.
Example: Connector
- 4 Name your file.
This file name must not contain any periods. The project is saved with the extension **.xml**.

- 5 Specify the folder in which your project is saved in the **Folder** field.
By default, Studio creates a sub-folder corresponding to the file type created in the folder of the current project. Example: a **Wizard** type file is created in an **actions** sub-folder.
- 6 You can select the **Add to project** option or you can leave it unselected.
If this option is selected, the XML file is referenced in the **.xwp** file that corresponds to the current project.
- 7 Click **OK**.

Opening a file


Studio can open the following files:

- **.xwp**
Files corresponding to a project
- **.wsp**
Files corresponding to a project's workspace
- **.xml**
Files corresponding to a project's elements
Example: an **.xml** file corresponding to the extension of a Connect-It Java resource connector.

To open one of the files, select **File/ Open project**. Studio opens a dialog box. Select a project and then click **Open**.

Depending on the types of files it can open, Studio adopts different components. Thus, when you open a **.wsp** file (workspace), Studio closes all the open windows and opens the windows corresponding to the last time this **.wsp** was saved.

Tip:

The  icon on the toolbar enables you to quickly open a project. The last projects are memorized by Studio and can be accessed from the **File** menu.

Open a file by drag and drop

To open a file you can also drag and drop a file from a Windows explorer to the Studio window. After it opens, you can add this file to the current project by:

- 1 Right-clicking.
- 2 Choosing **Add [name of the file] to the project** in shortcut menu.


Saving a project

Saving a project can correspond to:


- Saving one of the files open on the workspace.
- Saving the project file (**.xwp**).
- Closing and saving the workspace file (**.wsp**)

This save is performed automatically when you close Studio.

To save a file

- 1 Perform one of the following actions:
 - Select **File/Save**.
 - Click .

To save all the open files

- 1 Perform one of the following actions:
 - Select **File/Save all**.
 - Click .

To save a project

- 1 Select **File/Save the project**.

To save and close the workspace

- 1 Perform one of the following actions:
 - Select **File/Close the workspace**.
 - Quit Studio.

Studio workspace

The Studio workspace is composed of:

- A project explorer

This explorer presents all the files composing a project in a tree structure.

- A File editing window

This window presents a tree view of the file selected in the explorer.

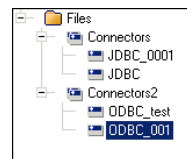
Project Explorer

The project explorer is one of the panes in the open workspace. It presents a tree view of the files in your project.

To display or hide a project explorer, perform one of the following actions:

- Select **Display/Project**.
- Click .

Figure 2.4. Explorer window



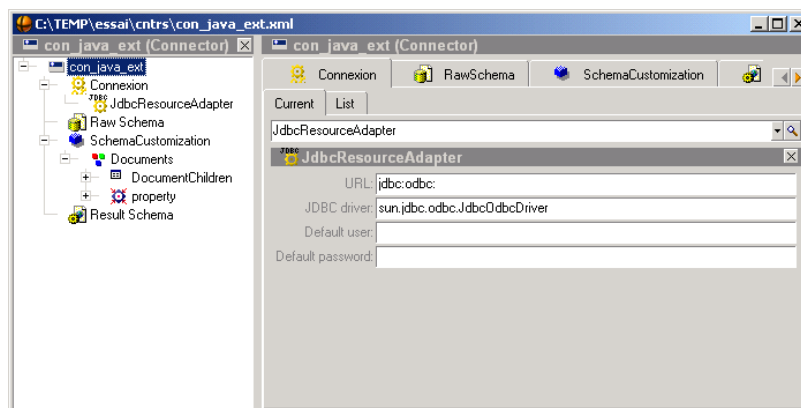
File editing window

- A left-hand pane containing the tree view of this file.
- A right-hand pane containing the detail of the node selected in the tree view.

For each file selected in the project explorer, Studio opens a editing window composed of two panes:

The following screen capture shows the window of a file corresponding to the extension of a connector.

Figure 2.5. Window of a project file




The following table details the function of the main icons of the Studio toolbar.

Icon	Function
	Displays or hides the XML source of the current selection.
	Displays or hides the tree view of the current file.
	Enables you to find the location of the file in the project explorer.
	Displays or hides the output window.

Navigating in the tree structure of a file

The Edit toolbar has icons that enable you to easily navigate through the current file's list of (XML) elements. The navigation functions are similar to those available in a Web browser. If you select an element in the hierarchy, Studio records and numbers each of these selections. You can navigate through your selections and/or your hierarchy by using the following icons:

Icon	Associated function
	Selects the previous element within the selection and displays the previous detail.
	Selects the next element within the selection and displays the next detail.

Icon	Associated function
	Enables you to go to the parent node of the current selection.

Export / Import an element

You can import or export an element via a selection's shortcut menu, which is available by right-clicking the selection.

To import an element

- 1 Select **Export** from the shortcut menu to export an element in XML format. This function enables you to:
 - Share XML code between different users.
 - Isolate a portion of XML code causing an error and send it with bug reports.
 - Save a portion of code for archive purposes.
 - Reuse portions of XML code in other projects or files.
- 2 Select the element in your project that you want to export.
- 3 Right-click and select **Export** from the shortcut menu.
- 4 Choose a name for the file that will store the information of the exported element and click **Save**.
- 5 Studio will export the element's definition in the XML file that you selected.

To import an element

- 1 Select **Import** from the shortcut menu to import an XML file describing a project element (and its child elements).
- 2 Select the parent element of the element that you want to import.
- 3 Right-click and select **Import** from the shortcut menu.
- 4 Select the file to import and click **Open**.
- 5 Studio imports the element under the element that was previously selected. Because the data format between two versions of Studio varies, you can only import elements previously exported by the same version.

 **Note:**

An element can only be imported into the file hierarchy where it is valid to do so. In this case, it would have to be imported under its parent element. For example, if you exported a module, you could only import it at the level of a group of modules.

Creating a new object in the file hierarchy

The creation of an object at a given hierarchical level of the file depends on the context. Depending on the selected node in the tree, Studio provides a list of objects that are valid for this context.

To create a new object

- 1 Select the parent node of the object that you want to create and right-click.
- 2 Studio displays a shortcut menu. The **New** menu entry contains the objects that you can create at this level of the hierarchy.
- 3 Select the desired element. Studio creates one or more new objects under the node that you have selected.

 **Tip:**

You can also use the **Insert** menu to create a new object.

Wizards and contextual menus

For certain elements in a file's hierarchy, contextual menus enable you to launch wizards or execute commands (by right-clicking).

Selecting objects

The file's explorer supports multiple selection of explorer objects:

- To select several objects, click on each desired object while holding down the keyboard's **Ctrl** key.
- To select a list of consecutive objects, select the first object, then hold down the **Shift** key while selecting the last object of the list. Studio automatically selects all the objects between the first and the last selected object.





 **Note:**

The multiple selection of objects can only be used to drag and drop these objects. You cannot, for example, edit the value of a common property.

Editing objects

All of Studio's standard editing functions (**Copy**, **Cut**, **Paste**, **Delete**) apply to every editable element. This means that not only your text can be edited, but also all your project's and file's objects (the hierarchical elements displayed in the project explorer).

Studio's Edit toolbar contains all of these editing functions:

Icon	Associated function
	Cuts the selected element. You cannot cut a file element if it is not in the current working package.
	Copies the selected element.
	Pastes the selected element.
	Deletes the selected element. Sometimes this function is not available. The most likely reason for this is because the element that you want to delete is not part of the current working package. In all cases, the window containing information on the selection gives you the existing restrictions for an element.

 **Tip:**

These editing functions are also available in the application's **Edit** menu and via the elements' shortcut menus.

Manipulate a file's elements

In the a file explorer, Studio enables you to:




- Organize the elements of a file by drag and drop.
- Organize the elements of a file by drag and drop.
- Reuse the elements of a file by copy and paste.

 **Tip:**

If you hold down the **Ctrl** key during a drag-and-drop operation, Studio performs a drag-and-paste: The source elements are conserved, and a copy of those elements appear in the new desired position.

 **Note:**

When you move elements, the mouse pointer changes appearance:

- If the drag-and-drop operation is possible, the cursor is shown as follows:

 - If the drag-and-drop operation is possible, the cursor is shown as follows:

 - If the drag-and-drop operation is not possible, the cursor is shown as follows: 
-

Drag and drop an element in a file

- 1 Select the element while holding down the mouse button.
- 2 Drag the element until its desired position in the tree view of a file.
- 3 Release the mouse button.

Drag and drop the elements in a file

- 1 Select the text that you want to move and hold down the mouse button.
- 2 Drag the cursor to the desired position of the text.
- 3 Release the mouse button.

Copy and paste drop an element in a file

- 1 Select the element in the file explorer while holding down the mouse button.
- 2 Press **Ctrl** on the keyboard.
- 3 Drag the element to the desired position in the file's tree structure; it will be automatically copied there when you release the mouse button.
- 4 Release the mouse button.



You can also use the shortcuts **Ctrl + C**, **Ctrl + V** to perform this operation.

Undo and Redo

Studio enables you to undo and redo an unlimited number of modifications, whatever their nature.

The last elements deleted and re-established are saved for each open file and for the project independently. Using deletion (**Ctrl + Z**) and re-establishment (**Ctrl + Y**) only concerns the window in focus.

These two functions are directly accessible via the Edit toolbar:

Icon	Description
	Undoes the last operation performed.
	Redoes the last canceled operation.

Tip:

These functions are also available through the **Edit/Undo** and the **Edit/Redo** menus.

Note:

The information used by Studio to undo or redo modifications is lost when you exit Studio or load a new project.

Performing a search

You can perform a search in a file's XML source or in a set of files of the project.

To perform a search in a file

- 1 Select the **Edit/Find** menu (**Ctrl + F**).
- 2 Specify the character string that you want to find in the **Find what** field.
- 3 Select the search options you desire, if any.
- 4 Click **Search**.

To search in a project or in files

- 1 Select the **Edit/Find in files** menu (**Ctrl + F**).
- 2 Select the type of search:
 - **Project files**

In this type of search, all the files of the projects (stored in the .xwp file) are searched through.

- **Listed files**

In this type of search, only the files whose folder is indicated in the **Folder** field are searched through.

- **Open files**

In this type of search, only the files are searched through.

- **Selected files**

In this type of search, only the current file is searched through.

3 Indicate the character string that you want to find in the **Find what** field.

4 Select the search options you desire:

- Whole word

Search for the whole word (space before and after).

- Match case

Respect the case used when entering the string being searched.

- XML search

Enables you to search the **Value** of the attributes or of the text.

Enables you to find the **Name** of tags or attributes.

- Pattern

Enables you to use regular expressions in the searched string.

- Recursive Search

Enables you to search through the elements included in the current search.

5 Click **Search**.

Navigating in Studio wizards

The following table lists the different buttons available in the Studio wizards:

Button	Operation
Previous	Returns to the previous page.
Next	Goes to the next page.
Finish	Finishes the wizard. It is at this stage that Studio performs the operations using the information entered by the user. If any pages in the wizard have not been populated, the default values will be used.

Button	Operation
Cancel	Cancels execution of the wizard. No operations are performed by Studio.

3 | Extending the definition of a connector

CHAPTER

Studio enables you to extend the description file of a Java resource connector.

 **Warning:**

The current version of Studio only lets you extend the instances of the Connect-It Database or JDBC connectors: This are optional connectors. Make sure that your Connect-It authorization certificate enables you to use them.

Extending this definition enables you to customize the document types published by this connector.

Customizing document types enables you to:

- Obtain a set of document types corresponding to only those data containers that interest you in the external application to which your resource connector is connected. Example: the database tables to which your JDBC connector is connected.
- Keep only those published document types whose elements are necessary to your mappings.
- Redefine the type of elements that you want to map.

Example: Transforming a text-type field into a date-type field.

- Redefine the links between the structures and collections of a document type that corresponds to the joins between the data containers in the external application. (Example: tables in a database).
- Etc.

This chapter proposes that you following these steps:

- 1 Create an integration scenario in Connect-It.
This scenario must use one or more resource connectors (JDBC or Database connectors).
- 2 Create a Studio project linked to the integration scenario.
This project enables you to extend the definition of your scenario's resource connectors. It helps to obtain a raw schema of the document types published by your connectors. (Example: the list of tables in a database without their links.)
- 3 Create customization rules in Studio.
Customization rules enable you to define the way in which document types (published by a resource connector instance) can be customized. Example: A specific rule specifies how the joins are exposed in the document types.
- 4 Preview customized document types
This step involves previewing customized document types from rules defined in the previous step. This preview is only available for the JDBC connector.
For the ODBC connector, this preview is not possible in Studio. You must open the extended connector in the Scenario builder to find out which extended document types are published by the connector.
- 5 Test the customized document-types in the Scenario builder.
After building customized document-types in Studio, you must test these document types in the Connect-It Scenario builder.

Presentation of sample data

The sample data provided with Connect-It illustrates how you can extend the definition of a resource connector in Studio.

The following sections use this data to illustrate each step of the extension process.

This sample data includes:

- Connect-It scenarios
- Studio projects associated with these scenarios.
- A Microsoft Access demonstration database (**demoreconc.mdb**).

Sample database

This database contains tables that correspond to the document types published by the JDBC Java source connector in the sample scenario.

Before using the sample integration scenario, you must declare the sample database in your ODBC administrator.

To declare this database:

- 1 Launch your ODBC administrator.
Example: **Start\Settings\Control panel\Administrative tools\ODBC data**
- 2 Add a new user data source.
- 3 Select the Microsoft Access database engine driver.
- 4 Name your data source.
The name used in the sample scenarios is **STUDIODB**.
- 5 Select the sample database whose path is **[Connect-It installation folder]/datakit/ studio/demoreconc.mdb**.
- 6 Validate the addition of your new data source.

The following table presents the tables and fields of the sample, demonstration database.

Table 3.1. Customizing document types - sample database

Table	Field	Type of field	Description
Assets			Assets table
	AssetTag	Text	Asset tag
	BlobField	OLE object	Binary object (picture, sound, etc.)
	CreationDate	Date/Hour	Date of creation
	IAssetID	Number	ID of the asset (primary key)
	IParentID	Number	ID of parent asset (external key)

Table	Field	Type of field	Description
	IProductID	Number	ID of a linked product (external key)
	MemoField	Memo	Important text
	SerialNumber	Text	Serial number of the asset
Products			Products table
	IProductID	Number	ID of the product (primary key)
	Brand	Text	Product brand
	Model	Text	Model of the product
SoftInstalls			Software table
	iSoftInstall	Number	ID of the software (primary key)
	lAssetId	Number	ID of a linked asset (external key)
	Brand	Text	Brand
	Model	Text	Template
	Version	Text	Version
SoftVersion			Software versions table
	versionId	Number	ID of the software version (primary key)
	SoftInstallId	Number	ID of a linked software
	AssetId	Number	ID of a linked asset
	Version	Text	Software version
AutoDesc			Relational table
	Table 1	Text	First table
	Table 2	Text	Second table
	LinkName	Text	Name of the link
	LinkType	Text	Type of link
	Field1	Text	Field 1
	Field1	Text	Field 2

Example scenarios

The following scenarios are provided with Connect-It:

- **datakit/studio/odbc_xml.scn**

This scenario has an ODBC source connector and an XML destination connector. The project corresponding to the scenario enables you to extend

the document types published by the ODBC connector using extended nodes.



- **datakit/studio/jdbc_xml.scn**

This scenario has an JDBC source connector and an XML destination connector. The project corresponding to the scenario enables you to extend the document types published by the JDBC connector using templates

- **datakit/studio/smsdb.scn**

This scenario enables you to have a Studio project in which the configuration file of the SMS connector (**config/sms/smsdb.cfg**) was rewritten to respect the Studio syntax.

Before opening the source connector of one of the sample scenarios:

- 1 Launch the Scenario builder.
- 2 Open the scenario.
- 3 Click .
- 4 Wait for Studio's main window to appear with the project corresponding to the scenario and containing the node of the scenario's connector.
- 5 Click  to go back to the Scenario builder.

This step enables you to create files containing the extension of document types published by the scenario's extended connector.

 **Note:**

This step must be respected each time that the project corresponding to the scenario is modified.

Studio projects

For each of the provided sample scenarios, a corresponding Studio project is available in the sub-folder with the same name as the scenario.

Example: For the scenario **datakit/studio/odbc_xml.scn**, the project **odbc_xml.xwp** is located in the sub-folder **datakit/studio/odbc_xml/**.

Associating a project with a scenario is only a matter of the name and the location of the scenario. For each **[Name of the scenario].scn** scenario, a Studio project is created in a **[Name of the scenario] / [Name of the scenario].xwp** sub-folder.

Creating a reference scenario


Creating a reference scenario is the first step enabling you to extend the definition of a connector.

This scenario must use at least one connector that can be extended. Only the JDBC and Database connectors can be extended.

Creating a Studio project linked to the reference scenario

You can create a Studio project linked to the reference scenario directly in the Connect-It Scenario builder.

To create a Studio project linked to the reference scenario

- 1 Open or create your integration scenario with the Scenario builder.
- 2 Click .


If the project linked to the scenario does not exist, this step automatically creates it.

 **Warning:**

If the scenario is not yet saved, the Scenario builder asks the user to save it. In effect, the created project is linked to the name of the scenario.

- 3 Wait until Studio's main window appears.

Contents of a connector-extension project

A Studio project linked to an integration scenario is represented by a tree structure that contains a set of parent nodes corresponding to the connectors ().

Contents of a node corresponding to a connector

A node corresponding to a connector contains the following elements:

- Connection

This element uses the connection defined when you configure your connector in the integration scenario.

Connections are classified alphabetically. At this time, only the JDBC connections are available. For the ODBC connectors, only the name of the connection appears.

- RawSchema

This element enables you to obtain the raw schema in the application to which the connector is connected. Example: the tables of a database for a JDBC connector. This raw schema serves as a reference for the creation of your customized document-types.

For more information about the raw schema, refer to the section [Updating the raw schema](#) [page 49] of this chapter.

- SchemaCustomization

This element enables you to create customization rules for the document types published by the connector. To learn more about these rules, refer to this chapter's section [Definition of customization rules](#) [page 50].

- Result Schema

This element enables you to build customized document-types. These document types vary depending on your customization rules. To learn more about building document types, refer to this chapter's section [Building and updating customized document-types](#) [page 71].

Editing the Connection element

JDBC connections

The JDBC connections (Java DataBase Connectivity) enable your resource connectors to use the JDBC API provided with Java and to connect to data servers.

Parameters of JDBC connections

The JDBC connection parameters are:

- JDBC driver

This parameter has the default value **sun.jdbc.odbc.JdbcOdbcDriver**. This generic driver is provided with the JRE or JDK. If problems occur with the generic driver, the user must use a driver adapted to the database server.

- URL

With the default driver, you must use the following syntax for this field:
jdbc:odbc:[name of the data source].

Example: jdbc:odbc:STUDIODB

If you use a JDBC driver different than the default driver, refer to its documentation in order to enter the correct syntax.

- Default user

This parameter corresponds to the user connecting to the database.

- Default password

This parameter corresponds to a password associated with a default user.

To create a JDBC connection

- 1 Select the **Connection** element.
- 2 Right-click.
- 3 Select **New\ JDBC connection (Insert\JDBC connection)**.
- 4 Enter your values in the detail window of the JDBC connection.

ODBC connections

Parameters of ODBC connections


The only ODBC connection parameter is the name of the Connection node.

To create an ODBC connection

- 1 Select the **Connection** element.
- 2 Right-click.
- 3 Select **New\ Connection ODBC (Create\Connection ODBC)**.
- 4 Enter the name of your connection in the detail window of this connection.

To declare the connector used by the connector

- 1 Select your connector.
- 2 Select the **Connection** tab.

- 3 Select one of the available connections from the list by clicking .

To change the connection parameters

- 1 Perform one of the following operations:
 - Select your connection from the list.
 - Select the **Connection** node under a connector.
- 2 Modify the connection parameters.

Updating the raw schema

Updating the raw schema of the external application enables you to create customization rules.

 **Note:**

Updating the raw data schema is only available for the JDBC connector.

This raw schema enables you to implement the document types that you want to customize using a drag-and-drop operation.

We recommend that you update this raw schema each time that the data structure changes in your external application or when you use another connection.

Example: A new table has been added to the external application's database.

To update the raw schema

- 1 Perform one of the following actions:
 - Select the root node of a connector in the file.
 - Select the **RawSchema** element located under the connector.
- 2 Right-click.
- 3 Select the **Update the raw schema** entry from the contextual menu.

 **Warning:**

If your external application contains a large data container (a database in an Asset Management application, for example), we recommend deleting the set of sub-nodes from your RawSchema node in order to reduce the size of your project files. The deletion of these sub-nodes also enables you to load your files more rapidly.

Definition of customization rules

Customization rules enable you to modify or add a document type published by a connector in the Scenario builder.

The customization rules are defined under the **SchemaCustomization** node of the node corresponding to your connector (Example: JDBC).

The document types created using these customization rules appear under the **Result Schema** node.

A customization rule enables you to:

- Modify a document type published by the connector.
- Create a new document type published by the connector.
- Modify the properties of the elements (structures, collections, attributes) of the modified or created document types.
- Define or redefine the links between the elements of modified or created document types.

Before creating your customization rules, we recommend updating the raw schema that the connector obtains from the external application to which it is connected. **Example:** The raw schema of a JDBC connector lists the tables not linked to the database to which it is connected.

For more information about updating the raw schema, refer to the section [Updating the raw schema](#) [page 49] of this chapter.

The customization rules that you create depend on your user profile.

To edit your user profile:

- 1 Select the **Tools/Options** menu.
- 2 Select the **Appearance** category.
- 3 Select a value from the **User profile** list.

 **Note:**

Each profile:

- Can create customization rules, to which are added those rules allowed by less advanced profiles.
- Can use the wizards available in this version of Studio.

Modifying or adding a node in the customization rules (SchemaCustomization)

In a project, adding or modifying a node under the **SchemaCustomization** node corresponds to the creation of a customization rule.

The following is a procedure for creating a customization rule:

- 1 Create or modify an extended structure, which corresponds to a document-type root node.
- 2 Create or modify sub-elements of this extended structure, which correspond to different document-type elements (attributes, structures, collections).
- 3 Create or modify links between the document types of the raw schema.

Adding a node - wizard mode

A wizard enables you to add a node corresponding to a customized document-type element.

To launch the **Create an extended mode** wizard:


- 1 Select **SchemaCustomization** or a structure or collection under this node.
- 2 Right-click.
- 3 Select the **Wizard/Create an extended mode** menu.

The following table presents all the pages in this wizard.

Table 3.2. Add or modify a node - wizard pages

 _____ Add or modify a node [page 52]

 _____ Identifying the node [page 52]

 _____ Associate a native name [page 52]

 _____ Define the reference string [page 53]

Add or modify a node

This page enables you to choose between adding a node or modifying an existing node.

Note:

If you launch the wizard after having selected the **SchemaCustomization** node, this page will not appear. In this case, selecting the option **Add** is implicit (you cannot modify the node **SchemaCustomization**).

Identifying the node

This page enables you to identify the node type that you want to add or modify.

Add a node

Select the node type of your choice (attribute, structure, collection) and enter its name in the field **Node name**.

Modify a node

This page is prepopulated according to the selected node. You can change its type and its name.

Associate a native name

This page enables you to associate a **'name'** attribute to your node. This attribute is from one of the document types present in the raw schema.

To associate the native name of a document type to your node:

- 1 Select the option **Indicate a native name**.
- 2 Enter the path of the **'name'** attribute in the text zone on the bottom of the page.

This path's syntax is the following: **SchemaCustomization.[Name of the document type].[path of the 'name' attribute of the node of your choice]**.

Example: SchemaCustomization.Assets.name

By selecting the option **Select from the list**, you can directly choose the node in the raw schema for which you want to recover the **'name'** attribute.

Example: To indicate the native name of the **AssetTag** field of an **Assets** document type:

- 1 Select the option **Select from the list**.

- 2 Select the **Assets** document type from the list.
- 3 Click the **Display children** button to display the attributes of the **Assets** document-type.

Use the **Display this level** button to redisplay the set of document types of the raw schema.

- 4 Select the **AssetTag** attribute from among the other document-type attributes.

The native name that appears in the text zone is the following:
SchemaCustomization.Assets.AssetTag.name.

Define the reference string

This page enables you to specify the path of the node of another document type defined in the customization rules (**CustomizationSchema**).

Using this reference path, the element that you add can display the children of a node in the raw schema as sub-elements.

By default, the reference path points to the node for which you have specified the '**name**' attribute during the previous step.

Example: If the native name specified on the previous page is **SchemaCustomization.Products.name**, the default reference path will be **SchemaCustomization.Products**.

Adding or modifying a link - wizard mode

A wizard enables you to add a link between the document types of a raw schema.




Example: For the JDBC connector, this wizard enables you to link the tables of the ODBC data source.

To launch the **Add or modify a link** wizard:

- 1 Select a document type, a structure or a collection under the **SchemaCustomization** node.
- 2 Right-click.
- 3 Select the **Wizard/Add a modify a link** menu.

The following table presents all the pages in this wizard.

Table 3.3. Add or modify a link - wizard pages

	Select link type [page 54]
	Select the source document type [page 54]
	Define or modify the parameters of the link [page 54]

Select link type

This page enables you to select the link type that you want to edit:

- A structure enables you to create a 1:1 type link.
Example: a relation between a product and a manufacturer.
- A collection enables you to create a 1:n type link.
Example: a relation between a computer and its installed software.

 Note:

This page does not appear if the node selected is the one created during a previous use of the wizard.

Select the source document type

This page enables you to choose the document type of the raw schema for which you want to create or modify a link.

If you select the **Select from the list** option, you can choose a document type of the raw schema.

Define or modify the parameters of the link

This page enables you to specify the following parameters:

- **Destination document type**
Enter in the field the name of the document type for which you want to define a link.
If you select the option **Select from the list**, you can choose a document type of the raw schema.
- **Link name**
Name your link.
- **Source attribute**

Enter in this field the name of the source document-type's attribute that enables you to create the link with the destination document type.

If you select the option **Select from the list**, you can choose an attribute from the source document type selected on the previous page.

- **Destination attribute**

Enter in this field the name of the destination document-type's attribute that enables you to create the link with the source document type.

If you select the option **Select from the list**, you can choose an attribute from the destination document type specified in the **Destination document type** field.

Click **Add** to validate the creation of the link.

You can then define a second link in this same page without having to relaunch the wizard.


To delete an existing link, select this link in the window and click **Delete**.

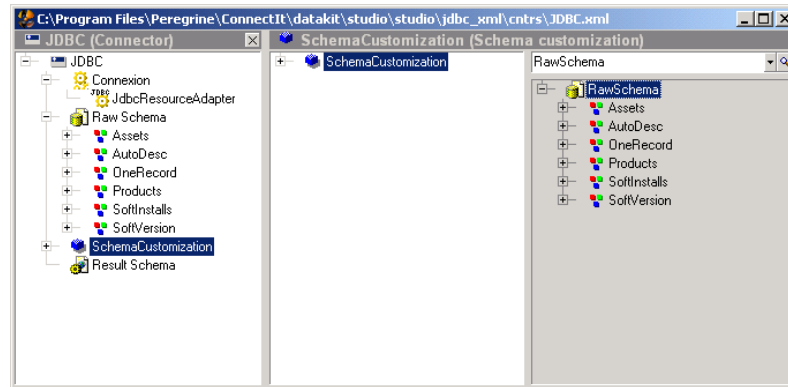
Creating customized document-types by drag-and-drop


Studio enables you to create a customized document-type by dragging and dropping a document type from a **RawSchema** node.

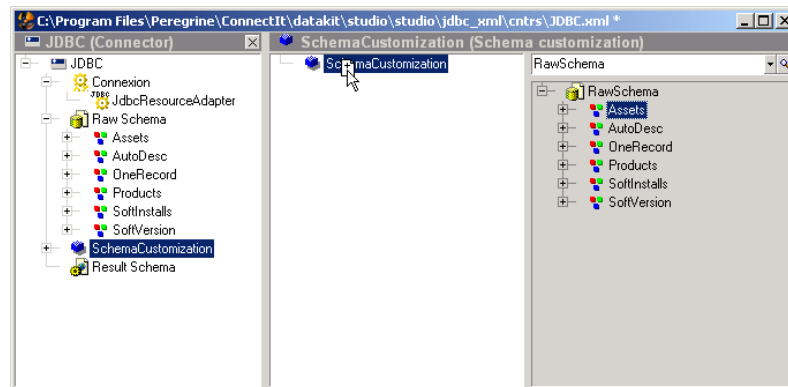
To obtain the set of document types of the raw schema

- 1 Verify that you have updated the raw schema of the resource connector.
To update the raw schema, refer to this chapter's section [Updating the raw schema](#) [page 49].
- 2 Select the **SchemaCustomization** node.

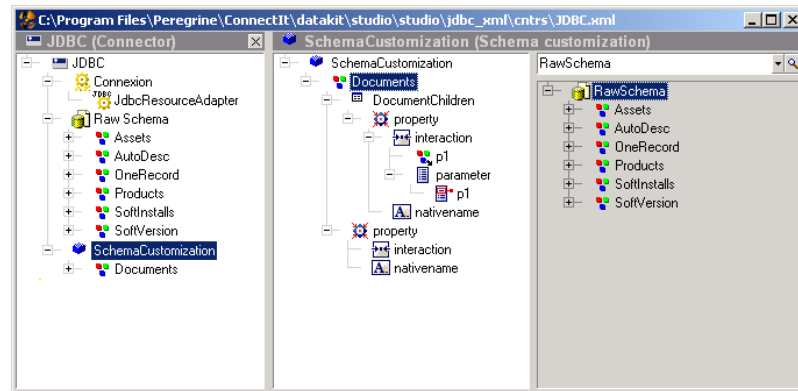
- Click  in the detail of the node (in the pane displaying the **RawSchema** node).



- Drag and drop the **RawSchema** node on the **SchemaCustomization** node until you see the following icon appear: .




- 5 Wait until the **Documents** node is created under the **SchemaCustomization** node.

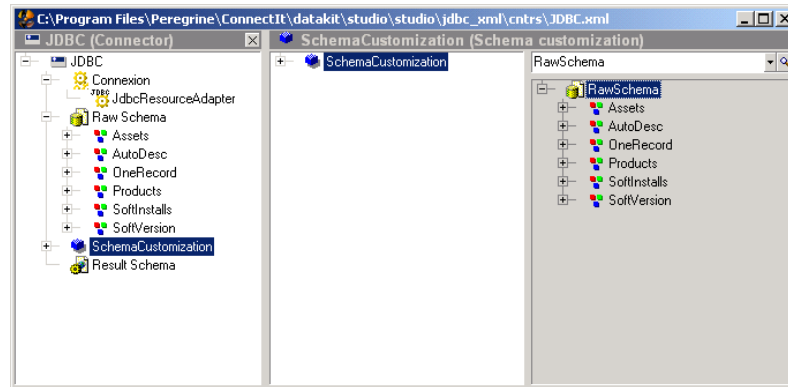




This **Documents** node enables you to obtain all the document types of the raw schema. This structure is identical to the one used to obtain the document types appearing under the **RawSchema** node.

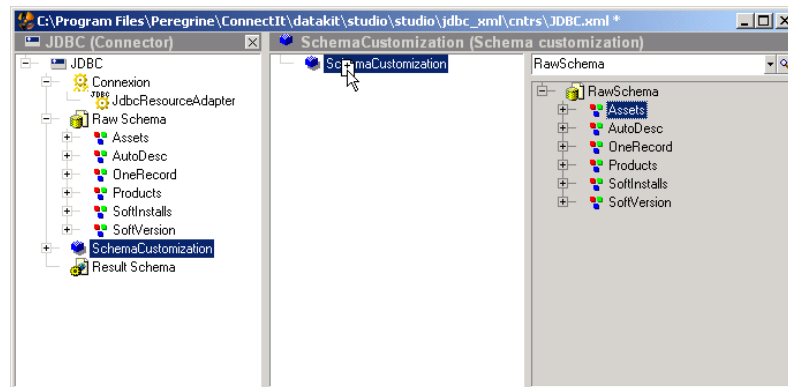
To create a customized document-type by drag-and-drop

- 1 Verify that you have updated the raw schema of the resource connector.
To update the raw schema, refer to this chapter's section [Updating the raw schema](#) [page 49].
- 2 Select the **SchemaCustomization** node.

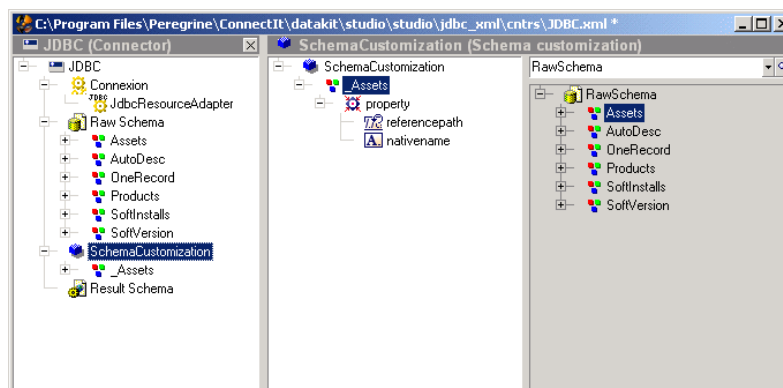
- 3 Click  in the detail of the node (in the pane displaying the **RawSchema** node).



- 4 Drag and drop the node of a raw-schema document-type on the SchemaCustomization node until the following icon appears: . You can drag and drop several document types at once. In this case, the following icon will appear: .



- 5 Wait until the document type is created under the **SchemaCustomization** node.




The following rules must be respected when creating a customized document-type by drag-and-drop:


- The customized document-type that is created must have the name of the raw-schema document-type suffixed by an underscore (_).
Example: The customized document-type based on the **Assets** document-type is thus named **_Assets**.
If you create several customized document-types from the same common document type, they will be numbered: **_Assets**, **_Assets2**, **_Assets3**, etc.
- The '**refnodename**' attribute of the native name (A) takes the value **SchemaCustomization.[Name of the document type in the raw schema].name**. Example: **SchemaCustomization.Assets.name**.
- The '**refnode**' attribute of the reference path (R) takes the value **SchemaCustomization.[Name of the document type in the raw schema].name**. Example: **SchemaCustomization.Assets**.

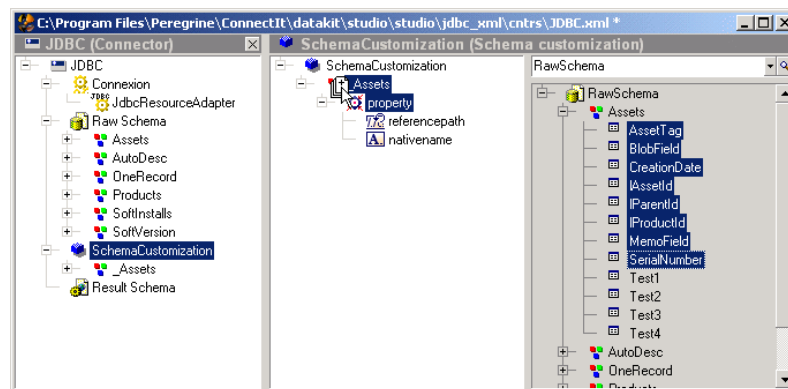
To drag and drop attributes in a customized document-type

Dragging and dropping one or more attributes into a customized document-type enables you to modify their type in the customization rules.

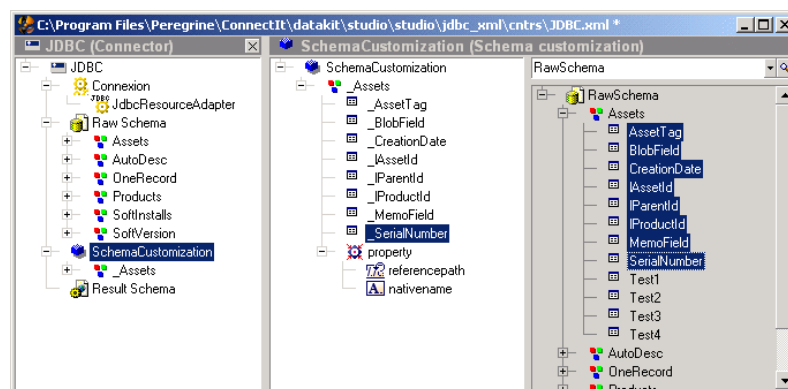
For more information about extending attribute types, refer to this chapter's section [Editing an element](#) [page 61].

- 1 Select one or more document-type fields for which you just created a customized document-type.
- 2 Drag and drop this attribute under the customized document-type until you see the following icon appear: .

You can drag and drop several attributes at once. In this case, the following icon will appear: .



- 3 Wait until the attributes appear under the customized document-type.



The following rules must be respected when creating a attributes by drag-and-drop:

- The attribute that is created must have the name of the raw-schema attribute suffixed by an underscore (_).
Example: The attribute based on the **AssetTag** attribute is thus named **_AssetTag**.
If you create several attributes from the same common attribute, they will be numbered: **_AssetTag**, **_AssetTag2**, **_AssetsTag3**, etc.
- For each added attribute, the type of attribute defined in the raw schema is conserved.


Editing an element

In the customization rules, Studio enables you to edit all the elements of a document type:

- Attribute
Attributes are those elements containing the values of the document types produced or consumed by resource connectors.
Example: For a JDBC connector, the attributes correspond to the columns of a database table.
- Structure
Structures are those elements corresponding to 1-1 links of the document types and other data containers.
- Collection
Collections are those elements corresponding to 1-n links of the document types and other data containers.
Example: A **Computer** document type with a **Software** collection means that n software(s) are installed on one computer (1:n link).

An element is characterized by:

- Its name.
The '**name**' attribute.
- The data type it contains (only for attributes).
The '**type**' attribute.
The type of an attribute can be a **character string**, **integer**, **long integer**, **floating**, **boolean**, etc.
- Properties.

Symbolized by the icon  in the tree structure of the customized document-type.

To rename an element

- 1 Select the element.
- 2 Select the **General** tab in the detail of this element.
- 3 Change its name.

You can also use the **F2** keyboard shortcut.

To expand the type of an attribute

- 1 Select the attribute.
- 2 Select the **Extended type** tab in the detail of this attribute.
- 3 Select a value in the **Extended type** field.

This field enables you to specify the new type that you want to assign to the attribute. Example: a **Short integer** type attribute that you want to convert to a **Long integer** type attribute in the customization of your resource connector.

- 4 If necessary, enter a value in the **Format** or **Memo description file** fields.

To select the Build a sub-document option

- 1 Select the element.
- 2 Select the **Build a sub-document** option in the element's detail window.

This option enables you to specify that the element is created by an interaction subordinate to the auto-description interaction of the resource connector.

For more information about the auto-description of resource connectors, refer to the Connect-It User's guide, chapter **Connect-It Java development kit**, section **Creating a Java connector**, section **Connector auto-description**.

In the **demo.adw** project, the **SchemaCustomization** node of the **demo/RelationalModel** node contains a **link1-1** structure and a **link1-n** collection whose vales are obtained using a **selectFromFile** interaction. The **Build a sub-document** option is selected for these two documents.

To select the Group the collections option

- 1 Select a collection.
- 2 Select the **General** tab in the detail of this collection.

- 3 Select the **Group the collections** option in the detail window of the collection.

This option enables you to group several collections into one single collection. This facilitates the mapping operations in the Scenario builder.

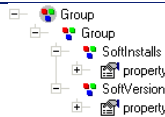
In the **demo.adw** project, the **SchemaCustomization** node of the **SchemaConfigurations/GroupCollection** node contains an **Assets** structure having a **Group** collection. This collection itself contains a sub-structure called **Group**, which has the nodes corresponding to the collections that you want to group.

The following table shows the different collections in the document types published by the resource connector in the Scenario builder.

Without collection grouping



With collection grouping



To define the properties of an attribute

- 1 Select the attribute.
- 2 Select the **Properties** tab.
- 3 Enter the property values in the **Properties** tab of the attribute detail.

To enter these values, consult the this section's sub-section: [Properties of an element](#) [page 63].

Properties of an element

The elements (attribute, structure, collection) of a document type have properties that you can modify in Studio.

To modify the properties of an element:

- 1 Select this element.
- 2 Right-click.
- 3 Select **New/Property** from the shortcut menu.

4 Modify the properties of the element in the three tabs displayed in the element detail:

- General
- Filter
- Interaction

Example: The **Display mode** property enables you to indicate if the element is exposed in the document types for production or for consumption.

General properties

Display mode

This field enables you to specify whether or not the selected node is present in the produced or consumed document-types published by the resource connector.

The following values are available:

- **Both**
- **Consumption**
- **Production**
- **Neither**

Reference path

This property enables you to specify the path (complete or relative) of a document-type element under the **SchemaCustomization** node. Studio uses this path to automatically obtain the sub-elements of the referenced node and to add them to the selected node.

Example: You create a structure corresponding to an **ITAssets** document-type for which you indicate the path of the **Assets** document.

This property uses the two following sub-properties:

- **Reference node**

This property enables you to specify the reference element of a document type already present under the **SchemaCustomization** node.

The path of an element respects the following syntax:

SchemaCustomization.[Name of a document type].[path of the element inside the document type]. [name of the element]. All parts of the path must be separated by a period (.).

Example: The reference of the **name** attribute in an **Assets** document-type is: **SchemaCustomization.Assets.Name**.

You can use the asterisk (*) to reference an attribute in the parent document-type of the selected structure or collection.

Example: If, for an **ITAssets** document-type structure or collection, you attribute the value ****Name** to the **refnode** attribute, then you will indicate the **Name** attribute of the parent document type of the selected parent element. The identical result would have been obtained with **ITAssets.ITAssets.Name**.

- **Reference node for interactions**

This property enables you to indicate the path of an element to which interactions apply.

This path defines the reference path of **@queryrefnode@**. When building the project, all the instances of **@queryrefnode@** in the code are replaced by the value of the **Reference node for interactions** field.

Native name

This property enables you to associate the name of one of the elements of a customized document-type to one of the elements of one of the raw schema's document types.

Filtering properties

This section details the parameters of the **Filter** tab of an element's properties.

Filters

Creating a filter for a node enables you to filter the data contained in this node in the customized document-type.

Creating a filter involves creating an operation between a right operand (**rightpart**) and a left operand (**leftpart**).

The following operations are available:

- =

Equal to

Example: For a filter formula, 'equal' indicates that the value of the **leftpart** must be equal to the value of the **rightpart** in order for the data to be produced by the resource connector.

- !=

Exception

Example: For a filter formula, the exception symbol means that all data corresponding to the value of the **leftpart** must be produced except for the data equal to the value of the **rightpart**.

- And

Addition (logical operation **and**)

- Or

Alternative (logical operation **or**)

- >

Greater than

- >=

Greater than or equal to

- <

Less than

- <=

Less than or equal to

Creating a filter - wizard mode

In Studio, a wizard enables you to create a filter.

To launch the **Add or modify a filter** wizard:





- 1 Select an attribute, a structure or a collection under the **SchemaCustomization** node.
- 2 Right-click.
- 3 Select the **Wizard/Add or modify a filter** menu.

The following table presents all the pages in this wizard.

Table 3.4. Create a filter - wizard pages

 _____ Select a type of operation [page 67]

 _____ Select a type of operand [page 67]

 Select a document-type node [page 67]	 Enter a constant value [page 67]	 Select a type of operation [page 67]
...
 Filter summary [page 67]		

Select a type of operation

This page enables you to select the filter's operation type.

Select a type of operand

This page enables you to select your operation's first operand type.

- Document-type node

In the case of a filter corresponding to a WHERE clause, the document-type node enables you to indicate a field from the document type on which the wizard was launched. Example: the **AssetTag** field of the **Assets** document-type.

- Constant value

An operand can be a constant value whose value and data type you indicate on the following page.

- Operation

If you select this option, a new [Select a type of operation \[page 67\]](#) page will open that enables you to indicate a new operation.

Select a document-type node

This page enables you to select a node in the document type on which you have launched the filter-creating wizard.

Enter a constant value

This page enables you to enter a constant value and its type.

Filter summary

This page proposes a summary of the created filter. The **Filter detail** pane shows the complete tree structure of the filter as it appears under the selected element after clicking **Finish**.

Interaction properties

The customization rules use the set of interactions supported by the resource connector. Example: A customization rule of the JDBC connector uses the **GetDocumentNames** interaction. This interaction enables it to obtain the name of the node that you added from the native name specified in the node detail.

For each **Interaction** node, the following tabs are available in the interaction detail:

- **General**
- **Document type**
- **Layer**
- **Parameters**
- **Template**

The resource connectors developed by Peregrine Systems have interactions specific to Connect-It that are presented in the following sections.

General

This tab enables you to indicate the type of interaction that you want to use among all those supported by your resource connector.

An interaction is an XML document that, when specifically related to Connect-It, uses different parameters - parameters, document types, layers - corresponding to document types that are manipulated by the connector. Example: The **select** type interaction uses a document type describing the structure of records to be found in the database to which the JDBC connector is connected.

For the interactions that can be used by the JDBC connector in Studio, consult this section's sub-section: [Interactions of the JDBC resource connector](#) [page 70].

In this tab, you must enter the values of the following fields:

- **Interaction name**
Enter the name of the interaction that you want to use for this customized document-type.
Example: The **Select** interaction enables the JDBC connector to perform queries to the database to which it is connected.
- **Exception**

This field enables you to limit the range of the interaction. Example: For the **GetDocumentNames** interaction, which enables you to obtain the set of document types and their attributes, you can enter the names of the document types that you don't want to obtain into this field. To separate names, use a semi-colon (;). Example: **Assets; Products**

To obtain only one document type, you can indicate its name in the exception field, preceding it with an exclamation point (!). Example: The value **!Assets** enables you to receive only the Assets document-type from the raw schema.

- **File path**

Specify here the file path of an XML file built automatically. This file is only used with a **selectFromFile** interaction.

This path must always be prefixed with the **%INTERACTION_VALUES_DIR%** environment variable. This variable references the folder containing the XML file, which has the values of the parameters used by the interaction.

Document type

This tab enables you to create a document type (structure) that describes the document type used by the interactions. Example: For the JDBC connector, the **select** and **selectFromFile** interactions use this document type to obtain records in the source database or in the file specified in the **File path** field.

Layer

This tab enables you to create a document type that details the document types already defined. Each layer contains additional parameters: schedule pointers, WHERE clause, etc.

Parameters

This tab enables you to create parameters used by the interaction. **Example:** the **_P1** parameter used by several **select** interactions of the project found in **[Connect-It installation folder]/datakit/studiojdbc_xml/jdbc_xml.wsp**.

To create a parameter

- 1 Select the interaction node under a customized document-type (🔍).
- 2 Select the **Parameters** node (📄) in the detail of the interaction.

- 3 Right-click.
- 4 Select the **New/Parameter** menu.
- 5 In the detail of the parameter (🔍) populate the following fields:
 - Name
 - Type
This field enables you to indicate the data type supported by the parameter.
 - Reference node
The reference node enables you to indicate the path of the 'name' attribute, which lets a parameter be linked to a structure, a collection or a document-type attribute.

Template

This tab corresponds to a document-type template that is reproduced for each record obtained by the interaction you use.

Interactions of the JDBC resource connector

getDocumentNames

This interaction enables you to obtain the document types that the resource connector can produce or consume.

getDocumentChildrenName

This interaction uses the reference node to recover the children of the nodes using this interaction.

select

This interaction enables the connector to perform the **resultSet** interaction operation. This corresponds to how the connector functions in production mode.

selectFromFile

This interaction is identical to the **select** interaction. In this case, the data is not obtained after reading the database. Instead it is obtained from a file specified in the **File path** field.

Building and updating customized document-types

After you create your customization rules, you can build customized document-types as they will be published by the connector in the Scenario builder.

 **Note:**

The update of customized document types is only available for the JDBC connector.

To build or update the document types published by a connector:

- 1 Select the node corresponding to your connector.
You can also select the **Result Schema** node.
- 2 Right-click.
- 3 Select **Update customized document-types** from the contextual menu.
Tools\ Contextual commands\ Update customized document-types.

 **Important:**

The update of document types rebuilds the configuration files of the extended connectors (**.ext**) used by the scenario linked to the project. Therefore, you should only update when you are sure of your modifications.


Testing customized document-type

After you build your customized document-types in Studio, you can test these document types in the Scenario builder.


You can test the customized document-types in production without having to execute the integration scenario.

You can test the customized document-types in consumption if you create a mapping between one of these document types and a document type produced by another connector.

To test customized document-types in production



- 1 Build or update the customized document-types in Studio.
- 2 Synchronize your scenario in the Scenario builder using Studio.
- 3 Select your connector.
- 4 Open the connector (F4).
- 5 Select the **Document types** tab.
- 6 Click .
- 7 Create a produced document-type using one of the customized document-types.

The customized document-types have the same name as they do in Studio.

- 8 Click  to obtain the data that will be contained in the documents, which will be produced from this document type.

You can also test these document types by creating a scenario using this document type.

To test customized document-types in production

- 1 Open your integration scenario.
- 2 Create a mapping using your customized document-type and another document type produced by one of your scenario's source connectors.
- 3 Click  to execute the scenario without writing to the destination applications.
- 4 Open your source connector and your destination resource connector (F4).
- 5 Select your source connector.
- 6 Click .
- 7 Consult the Connect-It log to verify the validity of your customized document-types.

Synchronization between Scenario builder and Studio

This section presents the interaction between the Scenario builder and Studio after the reference scenario is associated with a project.

Synchronizing the linked Studio project from Connect-It

The synchronization process enables you to update the Studio project associated with your scenario.

To link a project to an integration scenario, consult this chapter's [Creating a Studio project linked to the reference scenario](#) [page 46].

The synchronization process varies depending on the type of update you are performing:

- The connection parameters of one or several connectors changes.
After synchronization, the **Connection** nodes of your connectors take account of these modifications. Example: You change the data source or the driver of the JDBC connector.
- One or more connectors have been added to the scenario.
In the project, the synchronization creates nodes corresponding to these connector.
- One or more connectors have been deleted from the scenario.

The synchronization does not delete the **.xml** and **.ext** files corresponding to these connectors. On the other hand, they no longer appear in the project.


- One or more connectors have been renamed.

Just like when you add new connectors, the synchronization creates new nodes.

The following are the synchronization rules in the case where you rename any connectors:


- The nodes corresponding to the connectors having the old name (before the names were changed) no longer appear in the project. (The **.xml** and **.ext** files are not deleted.)
- If a connector is renamed with the name of an already existing connector, the synchronization updates this connector's features.

To synchronize the linked Studio project from Connect-It

- 1 Open your scenario in the Scenario builder.
- 2 Delete the changes in the scenario.
- 3 Click .
- 4 Wait for the Studio window to appear that takes into account your changes.

Synchronizing a scenario in the Scenario builder from Studio

The synchronization process enables you to update the integration scenario linked to your Studio project.

This synchronization is only possible when Studio was launched from the Scenario builder (using the icon .

The synchronization of a scenario automatically updates the extended description file (**customizedSchema.xml**) attached to each connector.

This extended description file can be manually entered on the **Connector deployment** page of the connector's configuration wizard (**Extended description file** field).

For more information about connector configuration, refer to the chapter **Connector configuration** in the **Connectors** guide.


The **[name of the connector].ext** file is created for each connector in your Studio project.

The extended description files are created in the same folder as the XML file corresponding to the connector in the project.


The synchronization process respects the following rules:

- A connector is extended in the Studio project.
If you select this connector for the synchronization, the **[name of the connector].ext** file accounts for these new modifications.
- A connector is created in the Studio project.
If you select this connector for synchronization, this connector appears in your integration scenario.

To synchronize a scenario in the Scenario builder from Studio.

- 1 Open your project.
(fichier **.wsp** or **.xwp**)
 - 2 Make the changes in your project.
 - 3 Click .
 - 4 Wait for the Scenario builder window to appear that takes into account your scenario changes.
-

 Note:

If Studio was not launched from Connect-It or if the Scenario builder is closed, the icon  is grayed out in the toolbar.

A Keyboard shortcuts

APPENDIX

The following table lists the keyboard shortcuts available in Studio. Some of these shortcuts can only be used in a specific context. In this case, the Description column informs you of the context.

Shortcut	Description
Ctrl+O	Opens a new project.
Ctrl+S	Saves an on-going project.
Ctrl+Z	Undoes the last operation performed.
Ctrl+Y	Redoes the last canceled operation.
F2	Renames the selected element.
Ctrl+X or Shift+Delete	Cuts the selection (or multi-selection).
Ctrl+C or Ctrl+Inser	Copies the selection (or multi-selection).
Ctrl+V or Shift+Inser	Pastes the selection (or multi-selection).
Delete	Deletes the selection (or multi-selection).
Ctrl+F	Performs a search.
F3	Searches the next element corresponding to the search criteria.
Shift+F3	Searches the previous element corresponding to the search criteria.

Shortcut	Description
Ctrl+R	Performs a replacement. This function is only available when you edit in source mode (accessible by clicking A).
Ctrl+Maj+F2	Adds the current selection as a bookmark in the Bookmarks tab of the General information window.
Ctrl+Alt+F2	Deletes the bookmark associated to the element selected in the project arborescence.
Ctrl+G	In a text (or code) edit window, enables you to directly go to a specific line identified by its line number.
Alt+Left Arrow	Returns to the previous view.
Alt+Right Arrow	Goes to the next view.
Ctrl+F7	Builds the selected element.
F7	Performs a differential build of the project. Those elements that have already been built are not rebuilt.
Alt+F5	Refreshes the screen.
F1	Starts the Studio online Help.
Shift+F1	Extended help on the selection.

B Glossary of terms

APPENDIX

This appendix defines the most common terms used in Studio.

Build

Operation resulting in a set of data that can be reused by the project.

DSD

Acronym of Data Structure Definition. A DSD defines all the element classes used in Studio and determines the relationships between the elements.

Elements

Elements are all objects belonging to a package, and by extension, all objects belonging to a project.

Workspace

File (extension **.wsp**) that saves Status when it is closed. This workspace, associated with a project, lists all the open windows. It can be opened automatically when Studio is opened.

Studio file

File susceptible to being opened by Studio. Each file has a particular type defined by the DSD. These files generally have the **.xml** extension.

Project Explorer

Graphical, tree-view of a Studio project. The explorer has advanced functions available to manipulate elements, such as multiple selection and drag-and-drop.

Link

A link is a property whose value is a reference to another element.

Project

The project contains a set of files that can be opened and edited in Studio.

Property

A property is a feature of an element. It always has a value (at least the default value) and has no hierarchy beneath it. The name of a property cannot be modified.

Index

INDEX

A

Available toolbars, 22

B

Build

Definition, 79

Building customized document-types, 73 , 71

Synchronizing a scenario in the Scenario builder from Studio, 74

Procedure, 75

Synchronizing the Studio project from Connect-It, 73

Procedure, 74

C

Connection element (edit) , 47

Changing parameters, 49

ODBC connections

Creating, 48

Parameters, 48

Connections

JDBC connections, 47

ODBC connections, 48

Connections (project element)

Declaring the connection of a resource connector, 48

JDBC connections

Creating, 48

Parameters, 47

Connector extension project

Connection element (edit) , 47

Contents, 47 , 46

Contents of a node corresponding to a connector, 47

Creating a filter - wizard mode, 66

Wizard pages, 66

Enter a constant value page, 67

Filter summary page, 67

Select a document-type node page, 67

Select a type of operand page, 67 , 67

Customization rules, 50

Adding a node - wizard pages, 51

Adding or modifying a link - wizard mode, 53

Creating a customized document-type with a drag-and-drop operation, 57

Creating customized document-types by drag-and-drop, 55

Displaying all the document types of the raw schema, 55

Dragging and dropping attributes in a customized document-type , 59

Editing an element, 61

Modifying or adding a link to the customization rules, 53

Modifying or adding a node to the customization rules, 51 , 51

D

Display mode of an element, 64

Dockable windows, 18

 Closing, 19

 Moving, 18

 Reopening, 19

 Resizing, 19

DSD

 Definition, 79

E

Editing an element of a customized
document-type, 61

 Attribute properties, 63

 Build a sub-document, 62

 Extending an attribute type, 62

 Group the collections option, 62

 Renaming, 62

Elements

 Definition, 79

Extending the definition of a resource connector,
41

 Building customized document-types, 73 ,
71

 Concepts, 41

 Creating a reference scenario, 46

 Creating a Studio project, 46

 Procedure, 46

 Sample data, 42

 Testing customized document-type, 71

 Updating customized document-types, 71

 Updating the raw schema, 49

F

Files, 28

 Project files, 28

File window

 Navigation, 33

Filtering properties, 65

 Creating a filter - wizard mode, 66

G

Glossary, 79

Grouping the collections, 62

I

Installation

 ADW file association, 14

 Installing for the first time, 14

 Procedure, 15

 Required environment, 13

Interaction properties, 68

 Creating a parameter, 69

 Document type tab, 69

 General tab, 68

 Layer tab, 69

 Parameters tab, 69

 Template tab, 70

Interactions of the JDBC resource connector, 70

 getDocumentChildrenName, 70

 getDocumentNames, 70

 select, 70

 selectFromFile, 71

Interface, 17

 Customization, 18

K

Keyboard shortcuts, 77

L

Layout

 General

 Presentation of the interface, 17

 Toolbar, 20

 Tree structure of a file, 33

 Windows, 18

 Wizards, 39

Link

 Definition, 80

M

Main windows, 19

 Buttons, 19

 Moving, 20

 Reorganizing, 20

 Resizing, 20

N

Native name of an element, 65

O

Options, 22
 Appearance, 23
 Color selection screen, 26
 Colors tab, 25
 General tab, 25
 Keywords tab, 26
 Startup, 24
 Text editor, 24
 Window, 22

P

Project, 28
 Creating, 29
 Definition, 80
 Opening, 30
 Saving, 31
 Project Explorer, 32
 Definition, 80
 Project explorer
 Canceling and reinitializing actions, 38
 Creating a new object, 35
 Drag-and-drop, 36
 Editing objects, 36
 Importing and exporting an element, 34
 Searching, 38
 Selecting objects, 35
 Window, 32
 Wizards and contextual menus, 35
 Properties of an element, 63
 Filtering properties, 65
 General properties, 64
 Display mode, 64
 Native name, 65
 Reference path, 64
 Interaction properties, 68
 Property
 Definition, 80

R

Raw schema
 Displaying all the document types of the raw schema, 55
 Reference node, 64
 Reference node for interactions, 65
 Reference path of an element, 64
 Reference node, 64
 Reference node for interactions, 65
 Resource connector
 Extending the description

S

Sample data, 42
 Database, 43
 Database - detail of the tables, 43
 Example scenarios, 44
 Studio project, 45
 SchemaCustomization (project element), 51
 Adding a node - wizard mode, 51
 Add or modify a node page, 52
 Associate a native name page, 52
 Define the reference string page, 53
 Identify the node page, 52
 Adding or modifying a link - wizard mode, 53
 Define or modify the parameters of the link page, 54
 Select link type page, 54
 Select the source document type page, 54
 Sub-document, 62

T

Testing customized document-type, 71
 Consumed document-types, 72
 Produced document-types, 72
 Toolbar, 20
 Activating or deactivating, 21
 Available toolbars, 22
 Description, 21
 Moving, 21

U

Updating customized document-types, 71

Updating the raw schema, 49

 Procedure, 49

W

Windows, 18

 Dockable windows, 18

 Main windows, 19

