The <oXygen/> User Guide

SyncRO Soft Ltd.

Sean Wheller

The <oXygen/> User Guide

SyncRO Soft Ltd.
Sean Wheller

Table of Contents

List of Figures

List of Tables

List of Examples

Chapter 1. Introduction

Welcome to the <oXygen/> XML Editor User Manual. This chapter provides an overview of <oXygen/>'s features and benefits and the organization of this book.

The <oXygen/> XML Editor is a cross-platform application for document development using structured mark-up languages such as XML, XSD, XSL, DTD.

<oXygen/> offers developers and authors a powerful Integrated Development Environment. Based on proven Java technology the <oXygen/> XML Editor's intuitive Graphical User Interface is easy-to-use and provides robust functionality for editing, project management and validation of structured mark-up sources. Coupled with XSLT and FOP transformation technologies, <oXygen/> supports output to multiple target formats, including: PDF, PS, TXT, HTML and XML.

<oXygen/> is the XML Editor of choice for developers, authors and integrators that demand high-quality output with a flexible and robust, single-source, structured mark-up environment.

Key Features

The <oXygen/> XML Editor offers the following key features and benefits.

Multiplatform availability: Windows, Mac OS X, Linux, Solaris.	Multilanguage support: English, German, French, Italian and Japanese.
Can be used as standalone desktop application, run through Java Web Start or as an Eclipse plugin.	Non blocking operations, you can perform validation and transformation operations in background.
Support for XML, XSLT, XML Schema, Relax NG , DTD, NRL schemas, WSDL and XQuery.	Ready to use FOP support to generate PDF or PS documents.
Validate XML Schemas, Relax NG schemas, DTDs, NRL schemas, WSDL, XQuery and CSS.	Validate XML documents with XML Schemas, Relax NG schemas, DTDs or NRL schemas.
Outliner.	Bookmark support.
Support for editing remote files over FTP, HTTP/ WebDAV and HTTPS/WebDAV.	Experimental XInclude support.
Easy error tracking - locate the error source by clicking on it.	Spell checking supporting English, German and French including locals.
Generate HTML documentation from XML Schemas.	Support for document frameworks: Docbook and TEI.
Conversions from DTD, Relax NG schema or a set of documents to XML Schema, DTD or Relax NG schema.	Context sensitive content assistant driven by XML Schema, DTD or by the edited document structure.
XML Catalog support.	Unicode support.
New XML document wizards to easily create documents specifying a schema or a DTD.	Syntax coloring for XML, DTD, Relax NG compact syntax, Java, C++, C, PHP, Perl, etc.
Pretty-printing of XML files.	Easy configuration for external FOPs.
Apply XSLT and FOP transformations.	XPath search and evaluation support.
Preview transformation results as XHTML or XML or in your browser.	Support for document templates to easily create and share documents.
Drag&drop support.	XML project manager.
Tree view/edit support for XML documents.	Batch validate selected files in project.
Configurable external tools.	Configurable actions key bindings.
Find and replace support allows regular expressions, is	All the usual editor capabilities (cut, copy, paste, find,

XML aware, handle multiple files.	replace, windows management).
Associate extensions with <oXygen/> on Windows.	Plugin support.
Mac OS X ready.	Print documents.
Import HTML documents.	Multidocument environment.
Model View.	Text transparency levels adjuster.
WSDL Support.	XQuery 1.0 support.
SVG Editor and Viewer.	XPath 2.0 support.
Debugger Backmapping support.	XSLT 2.0 full support.

About the <oXygen/> Handbook

This User Manual gives a complete overview of the <oXygen/> XML Editor and describes the basic process of authoring, management, validation of structured mark-up documents and their transformation to multiple target outputs. Throughout this manual it is assumed that you are proficient in the use of your operating system and the concepts related to structured mark-up.

The <oXygen/> XML Editor User Manual is comprised of the following parts:

- Chapter 1, *Introduction* : Introduction - you are reading it.
- Chapter 2, *Installation* : Installation - defines the platform and environment requirements of <oXygen/> and instructions for application installation, license installation, starting <oXygen/>, upgrade and uninstalling.
- Chapter 3, *Getting Started* : Getting Started with the <oXygen/> Interface - provides general orientation, explains concepts and defines functionality of the components that comprise the <oXygen/> Graphic User Interface (GUI).
- Chapter 4, *Editing Documents* : Editing - explains how to obtain maximum benefit from the editor, project and error validation features.
- Chapter 5, *Transforming Documents* : Transforming - explains the considerations for transformation of structured sources to multiple target format and how to obtain maximum benefit.
- Chapter 6, *XSLT Debugger* : XSLT Debugger - This chapter explains the Debugger modes functionality, which provides a rich set of features for development, testing and solving of XSL problems.
- Chapter 7, *WSDL Support* : WSDL Support - This chapter explains the facilities offered by <oXygen/> for WSDL support.
- Chapter 8, *XQuery Support* : XQuery Support - This chapter explains the support offered by <oXygen/> for editing, validating and running XQuery expressions.
- Chapter 9, *SVG Editor* : SVG Editor - explains how <oXygen/> can render the result of an XSL transformation that generates SVG documents..
- Appendix A, *Appendix* : Appendix - a collection of documents covering topics such as credits, licensing, errors and known problems.

Feedback and input to the <oXygen/> Handbook is welcomed.

Chapter 2. Installation

This section explains platform requirements and installation procedures. It also provides instructions on how to obtain and apply an <oXygen/> license, how to perform upgrades and uninstall <oXygen/> if required.

If you need help at any point during these procedures please send email to <support@oxygenxml.com>.

Caution

If you want to execute <oXygen/> with Java WebStart directly from <oXygen/> Java WebStart page [http://www.oxygenxml.com/javawebstart/] or your intranet server please configure your Java WebStart not to ask for desktop integration (File -> Preferences, Shortcuts), otherwise it will show up a dialog in the same time with the <oXygen/> license registration dialog leading to a blocking situation.

Installation Requirements

Platform Requirements

Minimum run-time requirements are listed below.

- Pentium Class Platform
- 128 MB of RAM
- 80 MB free disk space

Operating System, Tools and Environment Requirements

Operating System

Windows	All versions
Mac OS	minimum Mac OS X 10.0
UNIX/Linux	All versions/flavors

Tools

Installation packages are supplied in compressed archives. Ensure you have installed a suitable archive extraction utility with which to extract the archive.

Environment Prerequisites

Prior to installation ensure that your Operating System environment complies with the following:

- JRE 1.4 or higher
- The PATH environment variable is set to the most current Java VM installation.
- References to older Java VM installations are removed from the PATH.

Installation Instructions

Prior to proceeding with the following instructions, please ensure that your system complies with the prerequisites detailed in the installation requirements.

Note

The following instructions assume that JRE is installed. If you have downloaded an installation package that contains the JRE, please note that the package will automatically install Java prior to execution of the application.

Procedure 2.1. Windows Installation

Warning

Do not install the application in the same folder where you have downloaded the oxygen.exe installer as this conflicts with the application name.

1. Download the oxygen.exe installation kit and run it.
2. Follow the instructions presented in the installation program.

Procedure 2.2. Mac OS X Installation

1. Create a folder called oxygen on your local disk.
2. Within the oxygen folder, create child folder named in accordance with the <oXygen/> version number. The directory structure looks as follows: /../oxygen/5.0/
3. Download the Mac OS X Installation package (`oxygen.tar.gz`) to this folder.
4. Extract the archive to the same folder.
5. Execute the file named `oxygen`

Procedure 2.3. All Platforms Installation

1. Create a folder called oxygen on your local disk.

2. Within the oxygen folder, create child folder named in accordance with the <oXygen/> version number. The directory structure looks as follows: /../oxygen/5.0/

3. Download the All Platforms Installation package (`oxygen.tar.gz`) to this folder.

4. Extract the archive to the same folder.

5. On Windows execute `oxygen.bat`, for Mac execute `oxygenMac.sh`, and for Unix execute `oxygen.sh`.

Procedure 2.4. Windows NT Terminal Server

1. Install the editor on the server, making its shortcuts available to all users.

2. Edit the oxygen.lax file located in the install folder, changing the parameter "lax.nl.java.option.additional" to **lax.nl.java.option.additional = -Xmx256m -Dcom.oxygenxml.MultipleInstances=true** The "Xmx" value represents the maximum memory for each editor instance. Please make sure you tune them in a way that the multiple editor instances won't use all the available physical memory.

Procedure 2.5. Unix Server

1. Install the editor on the server, making sure the oxygen.sh script is executable and the installation directory is in the PATH of the users that need to use the editor.

2. Edit the oxygen.lax file located in the install folder, changing the parameter "lax.nl.java.option.additional" to **lax.nl.java.option.additional = -Xmx256m -Dcom.oxygenxml.MultipleInstances=true** The "Xmx" value represents the maximum memory for each editor instance. Please make sure you tune it in a way that the multiple editor instances won't use all the available physical memory.

3. Make sure the X server processes located on the workstations allow connections from the server host. For this use the xhost command.

4. Telnet (or ssh) on the server host.

5. Start an xterm process, with display on the workstation. Ex: **xterm -display workstationip:0.0**

6. Start the editor by typing **oxygen.sh**

Starting <oXygen/>

As a Java based application, <oXygen/> can run on all Operating Systems that support the Java Runtime Environment (JRE version 1.4 or later). The following instructions assume that JRE and the appropriate <oXygen/> distribution package for your Operating System are installed.

To start <oXygen/> follow the instruction for the installed package:

Procedure 2.6. Windows

- From the Windows Explorer double-click `oxygen.exe`.

Procedure 2.7. Linux

- At the prompt type: **sh oxygen.sh**.

Procedure 2.8. Mac OS X

- Double-click `oxygen`.

Procedure 2.9. All Platforms

- On Windows run `oxygen.bat`. On Mac OS X run oxygenMac.sh. On Linux/Unix run `oxygen.sh`

Obtaining and Installing an <oXygen/> License

<oXygen/> is not free software and requires a license in order to enable the application.

For demonstration and evaluation purposes a time limited license is available upon request from the <oXygen/> Web Site [http://www.oxygenxml.com]. This license is supplied at no cost for a period of 30 days from date of issue. During this period <oXygen/> is fully functional enabling you to test all aspects of the application. Thereafter, the application is disabled and a permanent license must be purchased in order to use the application. For special circumstances, if a trial period of greater than 30 days is required, please contact `<support@oxygenxml.com>`. All licenses are obtained from <oXygen/> Web Site [http://www.oxygenxml.com].

Once you have obtained a license the installation procedure is as follows:

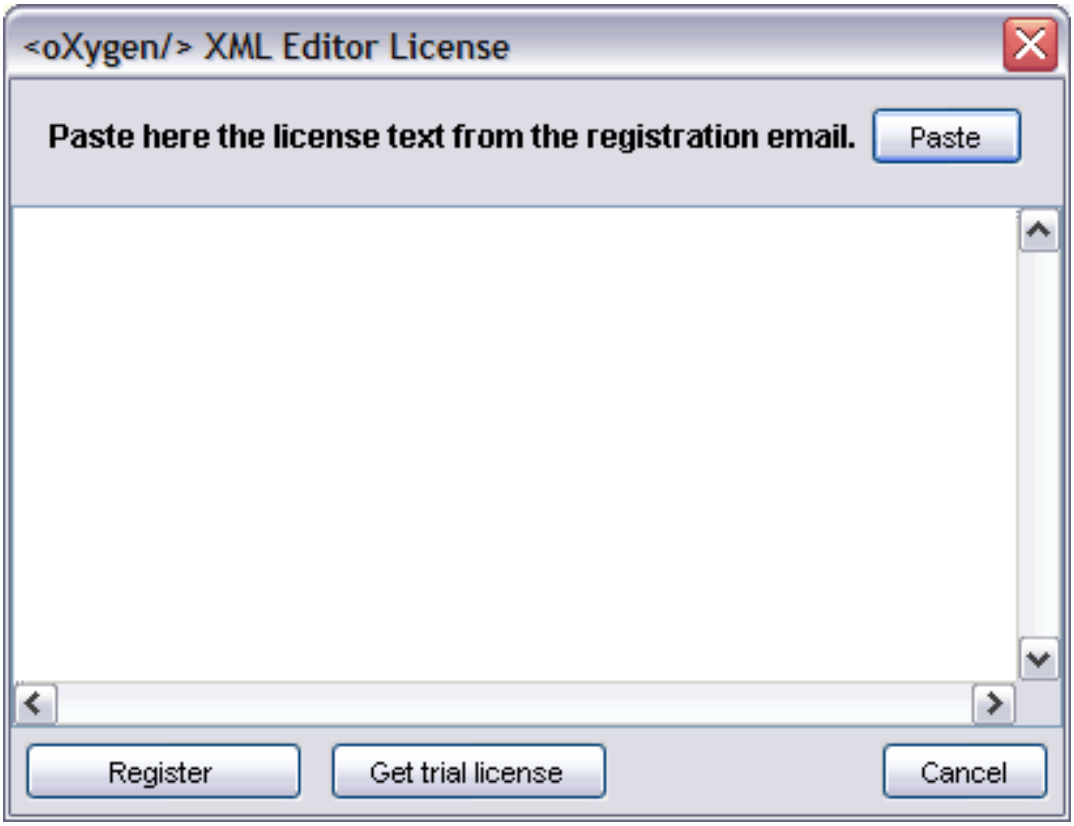
Procedure 2.10. License Installation

1. Save a backup copy of the message containing the new license file.
2. Start the <oXygen/> application.

3. Copy to the clipboard the license text as explained in the message.

4. If there is a new install of the editor then it will display automatically the registration dialog when it is started. In the case you already used the editor and obtained a new license, use the menu option Help/Register to make the registration dialog appear.

Figure 2.1. Registration Dialog



5. Paste the license text in the registration dialog, and press ok.

Upgrading <oXygen/>

From time to time, upgrade and patch versions of <oXygen/> are released to provide enhancements that rectify problems, improve functionality and the general efficiency of the application.

This section explains the procedure for upgrading <oXygen/> while preserving any personal configuration settings and customizations.

Unless otherwise stated by instructions supplied with a patch or upgrade kit, the following procedure is recommended:

Procedure 2.11. Upgrade Procedure

1. Create a new folder under /../oxygen e.g. /../oxygen/5.0
2. Download and extract the upgrade to the new folder.
3. If you have defined <oXygen/> in the system PATH, modify it to the new installation folder.
4. Start <oXygen/> to ensure that the application can start and that your license is recognized by the upgrade installation.
5. If you are upgrading to a major version, for example from 4.2 to 5.0, then you will need to enter the new license text into the registration dialog that is shown when the <oXygen/> plugin is activated.
6. Select Help->About to determine the version number. If the previous version was 4.2, the About dialog should now show <oXygen/> XML Editor v5.0.

Uninstalling <oXygen/>

Caution

The following procedure will remove <oXygen/> from your system. It will not remove the JRE. If you wish to uninstall JRE please see the instructions provided with the Java product. *Please ensure that all valuable data is saved to another location prior to performing this procedure.*

Procedure 2.12. Uninstall Procedure

1. Backup all valuable data from the <oXygen/> installation folder.
2. On Windows use the appropriate uninstall provided with your OS. Make sure that Options/Integrate into Explorer shell feature is not active.

   On Mac OS X and Unix manually delete the installation folder and all its contents.
3. If you wish to completely remove the application directory and any work saved in it, you will have to delete this directory manually. To remove the application configuration and any personal customizations delete the `.com.oxygenxml` directory from the user home directory.

Memory Management

When starting <oXygen/> it is possible to specify the amount of memory that will be available to the application by adjusting parameter variables contained in the startup script.

While there are minor changes in syntax to cater for differences in operating systems, the general structure of the command is as follows: **[execute] [memory variables] [application path] [call main class] [pass script arguments to main class]**.

For memory management we are only interested in the [memory variables] portion of the command. The installation default for the memory value is set to a value from 140 to 256 MB, as shown in the following example, where `-Xmx256m` establishes the maximum memory to be used to 256 MB. Adjust this value as required.

Another way to modify the available memory, if you are using the installer, is editing the oxygen.lax file located in the install folder and changing the parameter "lax.nl.java.option.additional" to **lax.nl.java.option.additional = -Xmx256m -Dcom.oxygenxml.MultipleInstances=true** The "Xmx" value represents the maximum memory for each editor instance.

When installed on a multi-user environment such as Windows Terminal Server or Unix/Linux, each instance of <oXygen/> will be allocated the amount stipulated in the memory value. To avoid depreciating the general performance of the host system, please ensure that the amount of memory available is optimally apportioned for each of the expected instances.

Example 2.1. Example Startup Script (Windows)

```
java -Xmx256m -cp ".: ./lib/oxygen.jar" ro.sync.exml.Oxygen $1 $2 $3
$4 $5 $6 $7 $8 $9
```

Modifying the value from 256 to 100 sets the memory available from 256 to 100.

Chapter 3. Getting Started

This section provides an overview of the <oXygen/> Graphic User Interface (GUI). It provides you with an explanation for each of the interface components and a short description of its purpose or usage. The <oXygen/> work area is split in three panes:

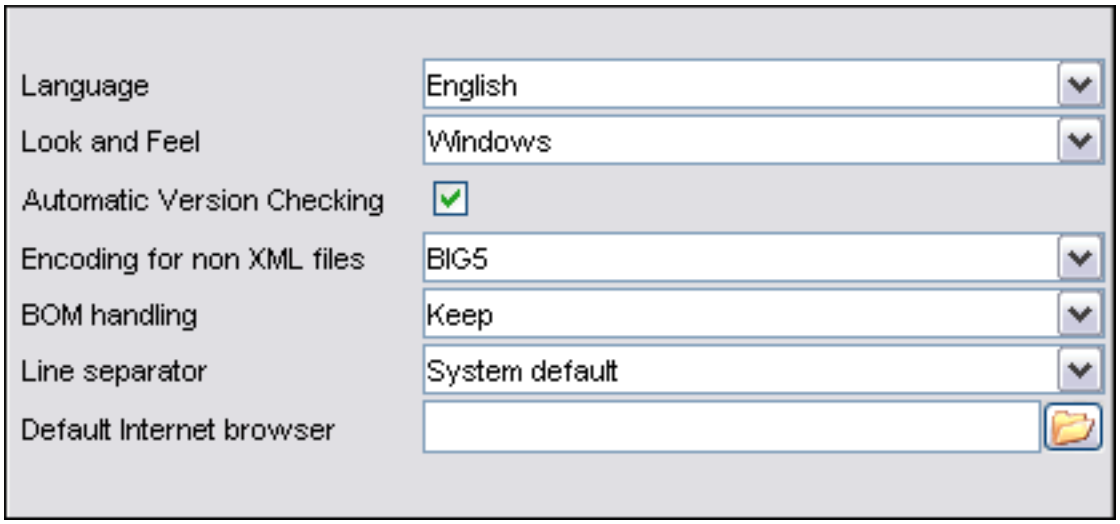
- The <oXygen/> preferences
- The main window
- The Tree View window

Preferences

Once <oXygen/> is installed you may want to use the following preferences to customize <oXygen/> for your requirements and network environment.

Global

Figure 3.1. The Global preferences



Change Interface Language	<oXygen/> supports a number of languages for localization of the GUI. Select Options->Preferences->Global+Language droplist to display the language choices.
Changing Look and Feel	Use this option to change graphic style (look and feel) of the GUI.
Automatic Version Checking	When enabled, checks the availability of new <oXygen/> versions at http://www.oxygenxml.com.
Encoding for non XML files	This option defines the default encoding to be used when opening non XML documents.

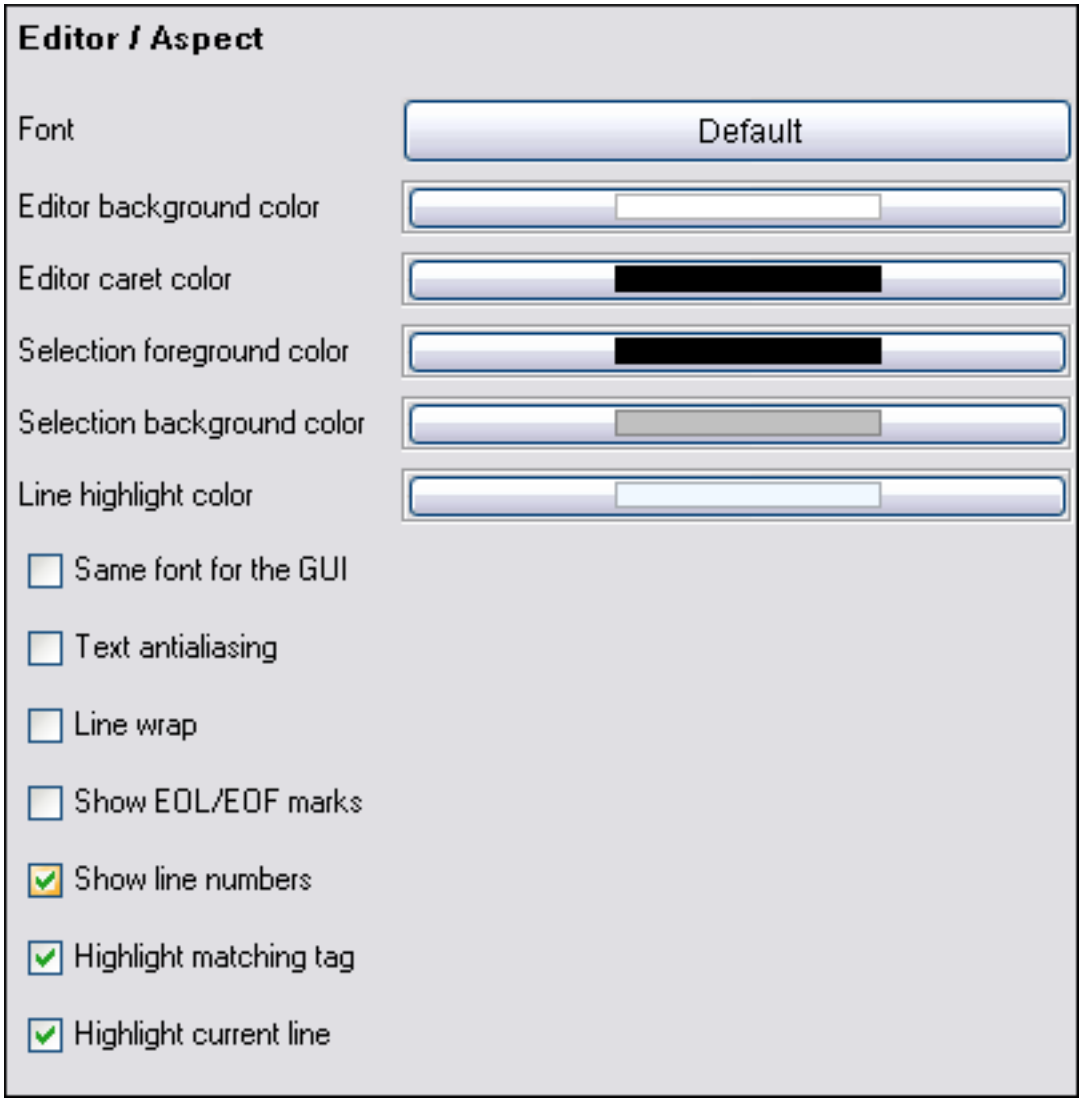
BOM handling	This option defines how to handle the BOM (Byte Order Mark) on document save. The available options are: • Don't Write - Don't write the BOM bytes, the loaded BOM bytes are ignored; • Write - Write the BOM bytes accordingly with chosen encoding; • Keep - If the loaded document has BOM then write them accordingly with chosen encoding. This is the default option.
Line separator	This option defines line separator to be used.System Default choice sets the line separator from the platform.
Default Internet browser	The path to a web browser executable. The browser is used to open XSLT or PDF transformation results, to open the <oXygen/> homepage or to point to specific paragraphs in the W3C recommendation of XML Schema grammars on the W3C website in case of validation errors.

Editor

Use these options to configure the visual aspect, formatting parameters, and behaviour of the content assistant.

Aspect

Figure 3.2. The Aspect pane

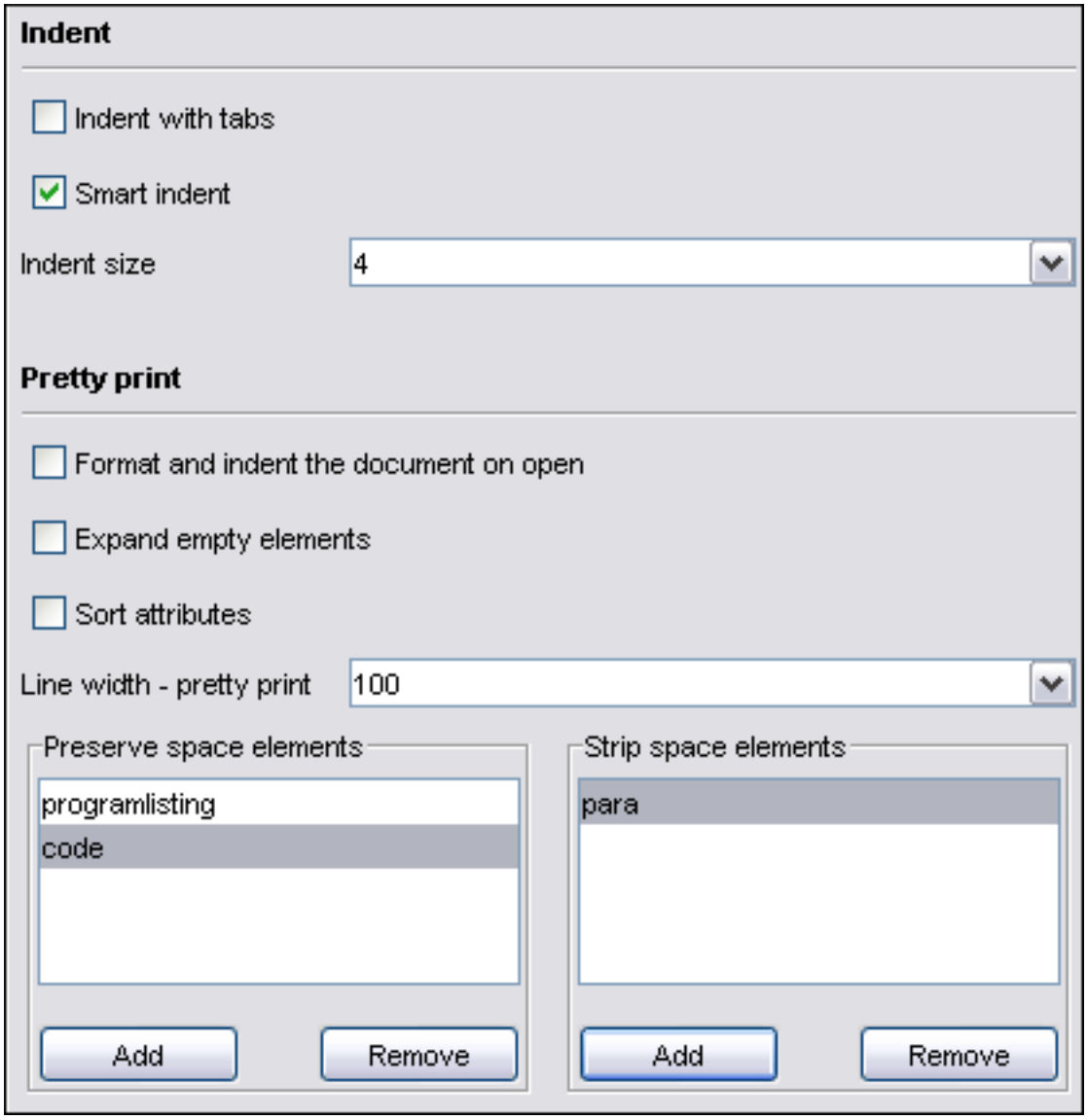


Font	Use this option to select the font family and size used to display text in the editor.
Editor background color	Use this option to set the background color of the editor.
Editor caret color	Use this option to set the background color of the editor.
Selection foreground color	Use this option to set the text color of selected text.
Selection background color	Use this option to set the background color of selected text.
Line highlight color	Use this option to set the highlight color for the line on which the caret is situated.
Same font for the GUI	When checked the editor will use the font from the GUI.
Text antialiasing	This option indicates whether text strings should be drawn with antialiased rendering.

Line Wrap	This option will automatically wrap lines in edited documents.
Show EOL/EOF marks	Marks the EOL/EOF using small icons, for a better visualisation of the document.
Show line numbers	This option enables the line numbers column located in the left part of the editing space. When unchecked, line numbers option is disabled.
Highlight matching tag	This options enables highlight for the tag matching the one on which the caret is situated.
Highlight current line	Enables highlight for the current line.

Format

Figure 3.3. The Format pane

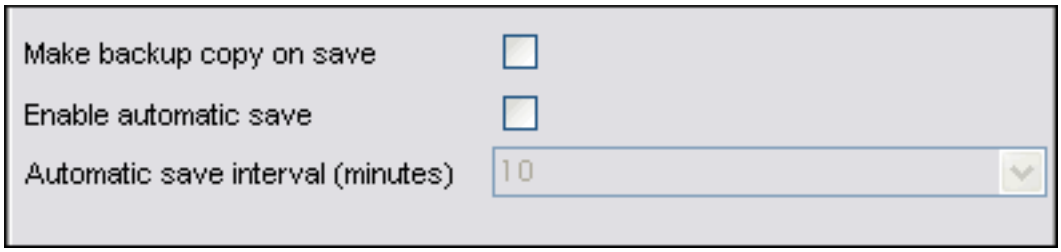


Indent with tabs	When checked enables 'Indent with tabs' to sets the indent to a tab unit. When unchecked, 'Indent with tabs' is disabled and the indent will measure as many spaces as defined by the 'Indent size' option.
Smart indent	Smart Indent attempts to indent mark-up in accordance to the tag/ text and its position/ context within the body of a document. This option either enables or disables the use of smart indenting.
Indent size	Sets the number of spaces or the tab size that will equal a single indent. The Indent can be spaces or a tab, select the preference using the Indent With Tabs option. If set to 4 one tab will equal 4 white spaces or 1 tab with size of 4 characters depending on which option was set in the Indent With Tabs option.
Format and indent the document on open	When checked, the *Format and indent the document on open* operation will format and indent the document before open

Expand empty elements	When checked the *Format and Indent* operation will output empty elements with a separate closing tag, ex. <a atr1="v1">. When not checked the same operation will represent an empty element in a more compact form: <a atr1="v1"/>
Sort attributes	When checked the *Format and Indent* operation will sort the attributes of an element alphabetically. When not checked the same operation will leave them in the same order as before applying the operation.
Line width - pretty print	Defines the point at which the "Format and Indent" (Pretty-Print) function will perform line wrapping. So if set to 100 Pretty-Print will wrap lines at the 100th space inclusive of white spaces, tags and elements.
Preserve space elements	This list contains the names of the elements for which the contained white spaces like blanks, tabs and newlines are preserved by the *Format and Indent* operation exactly as before applying the operation.
Strip space elements	This list contains the names of the elements for which contiguous white spaces like blanks, tabs and newlines are merged by the *Format and Indent* operation into one blank.

Save

Figure 3.4. The Save pane



Make backup copy on save	If checked, a backup copy is made when saving the edited document.
Enable automatic save	Automatic save is a useful feature that ensures your work is being saved in the background. You can specify the time intervals between automatic saves. If checked it enables Automatic Save. When unchecked, Automatic Save is disabled.
Automatic save interval (minutes)	Select the period in minutes for Auto Save intervals.

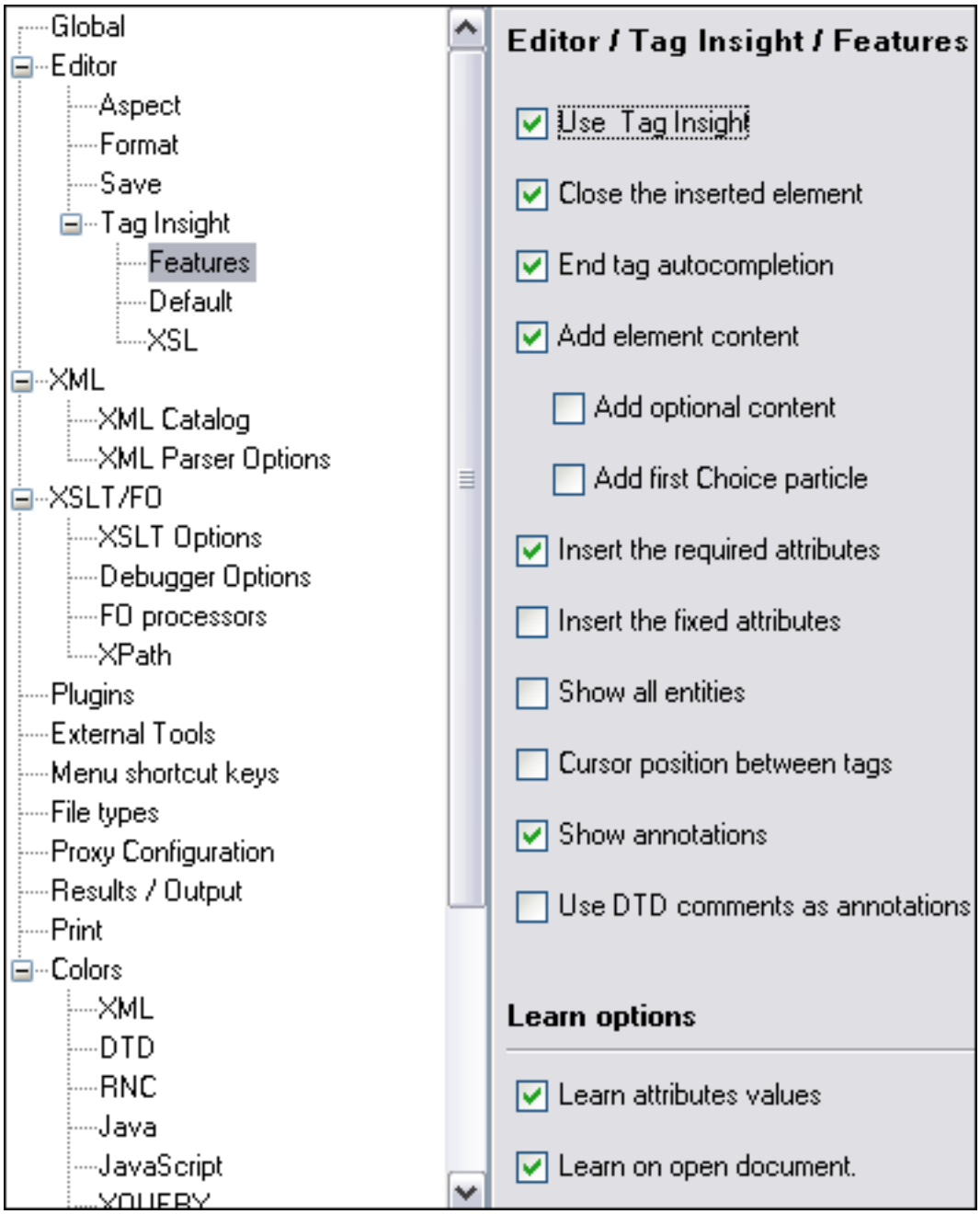
Tag-Insight

The Tag-Insight feature enables inline syntax lookup and Auto Completion of mark-up elements and attributes to streamline mark-up and reduce errors while editing.

Features

These settings define the operating mode of the content assistant.

Figure 3.5. The Tag Insight Features pane



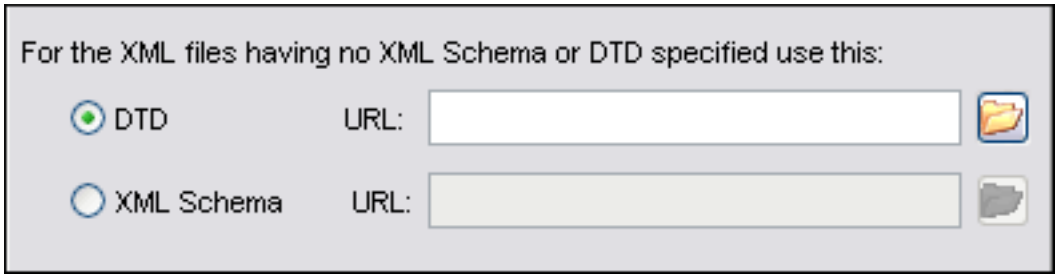
Use Tag-Insight — This option enables Tag-Insight feature. When unchecked, all Tag-Insight features are disabled.

Close the inserted element	When inserting elements from the Tag-Insight assistant, both start and end tags are inserted.
End tag autocompletion	Having manually typed the start tag of an element, <oXygen/> will automatically insert the end tag when </ is typed.
Add element content	When checked, <oXygen/> will insert automatically the required elements from the DTD or XML Schema.
Add optional content	When checked, <oXygen/> will insert automatically the optional elements from the DTD or XML Schema.
Add first Choice particle	When checked, <oXygen/> will insert automatically the first Choice particle from the DTD or XML Schema.
Insert the required attributes	When checked, <oXygen/> will insert automatically the required attributes from the DTD or XML Schema for an element inserted with the help of the Tag-Insight assistant.
Insert the fixed attributes	When checked, <oXygen/> will insert automatically any *FIXED* attributes from the DTD or XML Schema for an element inserted with the help of the Tag-Insight assistant.
Show all entities	When checked, <oXygen/> will display a list with all the internal and external entities declared in the current document when the user types the start character of an entity reference (i.e. &).
Cursor position between tags	When checked, <oXygen/>, will set the cursor automatically beetween tags. Even if the auto-inserted elements have attributes that are not required, the position of cursor can be forced between tags.
Show annotation	When checked, <oXygen/>, will display the annotations that are present in the used schema for the current element, attribute or attribute value.
Use DTD comments as annotation	When checked, <oXygen/> will use all DTD comments as annotation.
Learn attributes values	When checked, <oXygen/> will display a list with all attributes values learned from the current document.
Learn on open document	When checked, <oXygen/> will automaticaly learn the document structure when the document is opened.

Default

The URL of the default DTD / XML Schema to be used when no grammar is specified in the edited document.

Figure 3.6. The Tag Insight Default pane



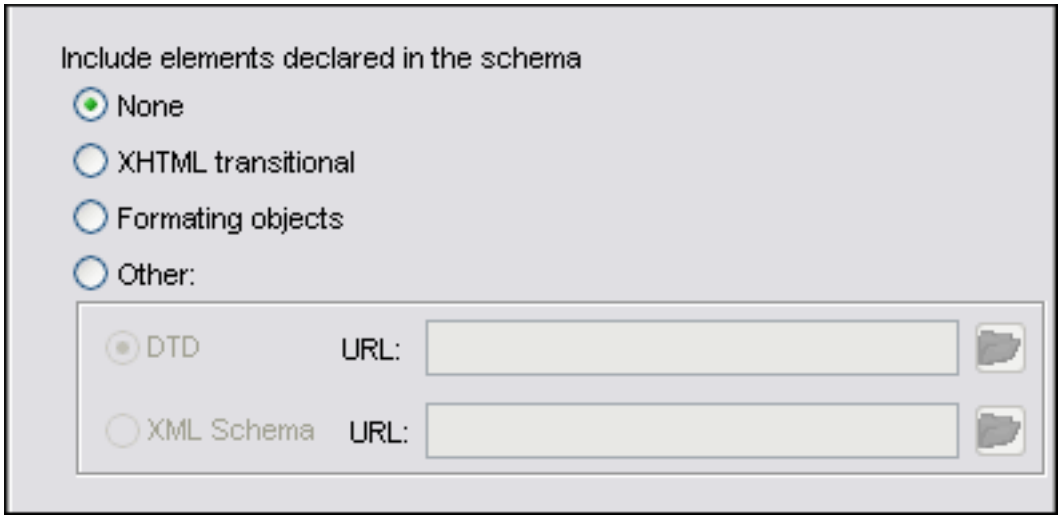
DTD — Used to specify the full path location of the DTD file that will be used to initialize the Tag-Insight assistant when a document does not define a DTD, XML Schema, Relax NG or NRL schema.

XML Schema — Used to specify the full path location of the XML Schema file that will be used to initialize the Tag-Insight assistant when a document does not define a DTD, XML Schema, Relax NG or NRL schema.

XSL

These settings define what elements are suggested by the content assistant in addition to the XSL ones.

Figure 3.7. The Tag Insight XSL pane



None — The Tag-Insight will offer only the XSL information.

XHTML transitional — Includes XHTML Transitional elements as substitutes for xsl:element.

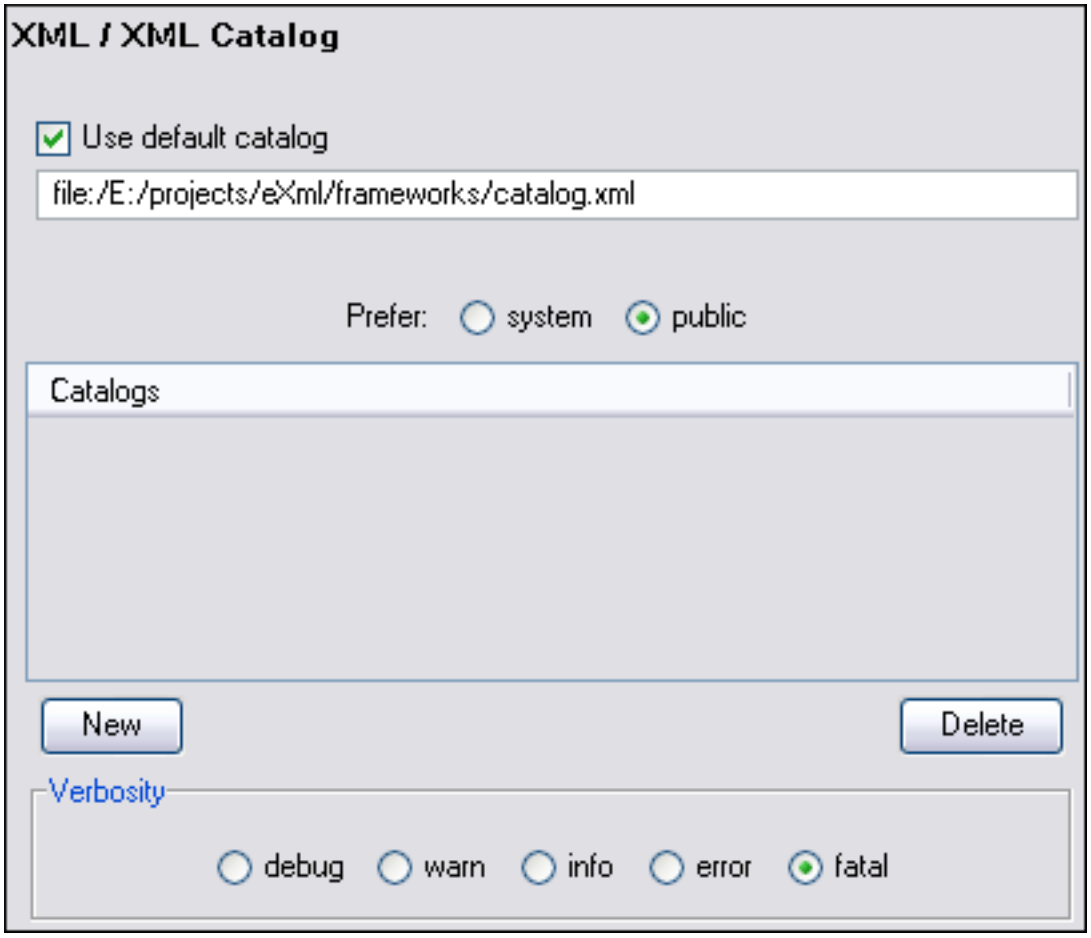
Formating objects — Includes Formating Objects elements as substitutes for xsl:element.

Other — Includes elements from a DTD file or a XML Schema file specified from a URL as substitutes for xsl:element.

XML Catalog

An XML catalog is a set of mappings between remote DTD and/or XML Schema and/or Relax NG files and local copies of these files. When Internet access is not available or the connection is slow, one or more XML catalogs can be added to the list in the dialog below and the local copies of the DTD and/or XML Schema and/or Relax NG files will be used during validation. When you add/delete an XML catalog to/from the list of XML catalogs in the Options -> Preferences -> XML Catalog pane you must restart the application so that the changes take effect.

Figure 3.8. The XML Catalog pane



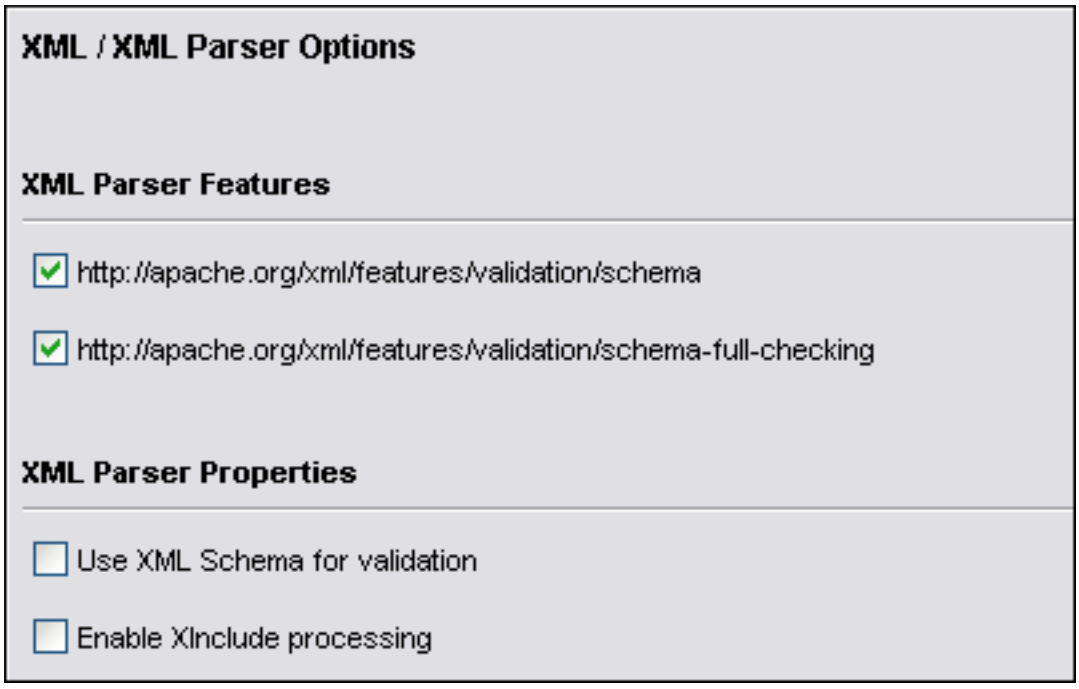
If "Use default catalog" option is checked <oXygen/> will use the built-in catalogs for DocBook, TEI and XHTML documents located in the *frameworks* subdirectory of the installation directory. Otherwise <oXygen/> will use the catalogs specified in the list.

The Prefer option is used to specify whether <oXygen/> will try to resolve first the PUBLIC or SYSTEM reference using the specified XML catalogs. If a PUBLIC reference is not mapped in any of the catalogs then a SYSTEM reference is looked up.

The verbosity level specifies the types of output messages displayed to standard output and can have one of the values: debug, warn, info, error and fatal.

XML Parser Options

Figure 3.9. The XML Parser Options pane



http://apache.org/xml/features/validation/schema - This option sets the 'schema' feature to true.

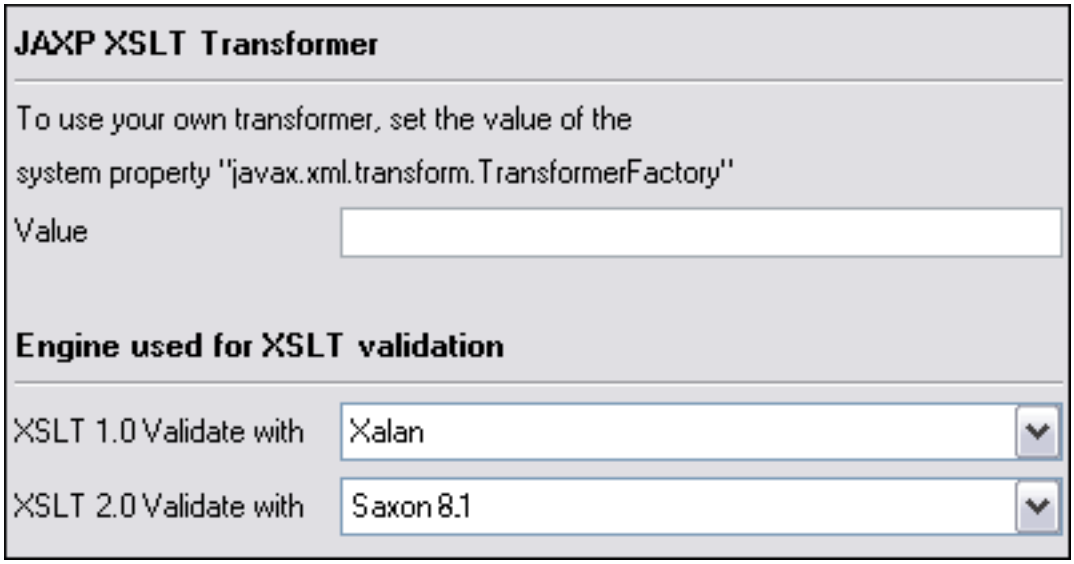
http://apache.org/xml/features/validation/schema-full-checking - This option sets the 'schema-full-checking' feature to true.

Use XML Schema For Validation - This option forces validation against a referred XML Schema even if the document includes a DTD declaration.

Enable XInclude processing - if checked the XInclude support in <oXygen/> is turned on.

XSLT Options

Figure 3.10. The JAXP XSLT Transformer option



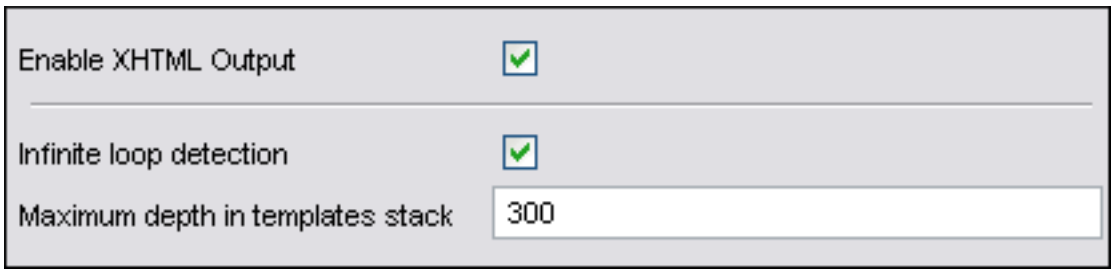
If you want to use an XSLT transformer different than the ones that ship with <oXygen/> namely Apache Xalan and Saxon all you have to do is to specify the name of the transformer's factory class which <oXygen/> will set as the value of the Java property "javax.xml.transform.TransformerFactory". To perform an XSLT transformation with Saxon 7 for instance you have to place the Saxon 7 jar file in the <oXygen/> libraries directory (the *lib* subdirectory of the installation directory), set "net.sf.saxon.TransformerFactoryImpl" as the property value and select JAXP as the XSLT processor in the transformation scenario associated to the transformed XML document.

Value	Allows the user to enter the name of the transformer factory Java class.
XSLT 1.0 Validate with	Allows the user to set the XSLT Engine used for validation of XSL 1.0 documents.
XSLT 2.0 Validate with	Allows the user to set the XSLT Engine used for validation of XSL 2.0 documents.

Debugger Settings

This section explains the settings available for Debugger mode. To display settings select Options->Preferences+Debugger Options (see Figure 3.11, “ Debugger Settings ”).

Figure 3.11. Debugger Settings



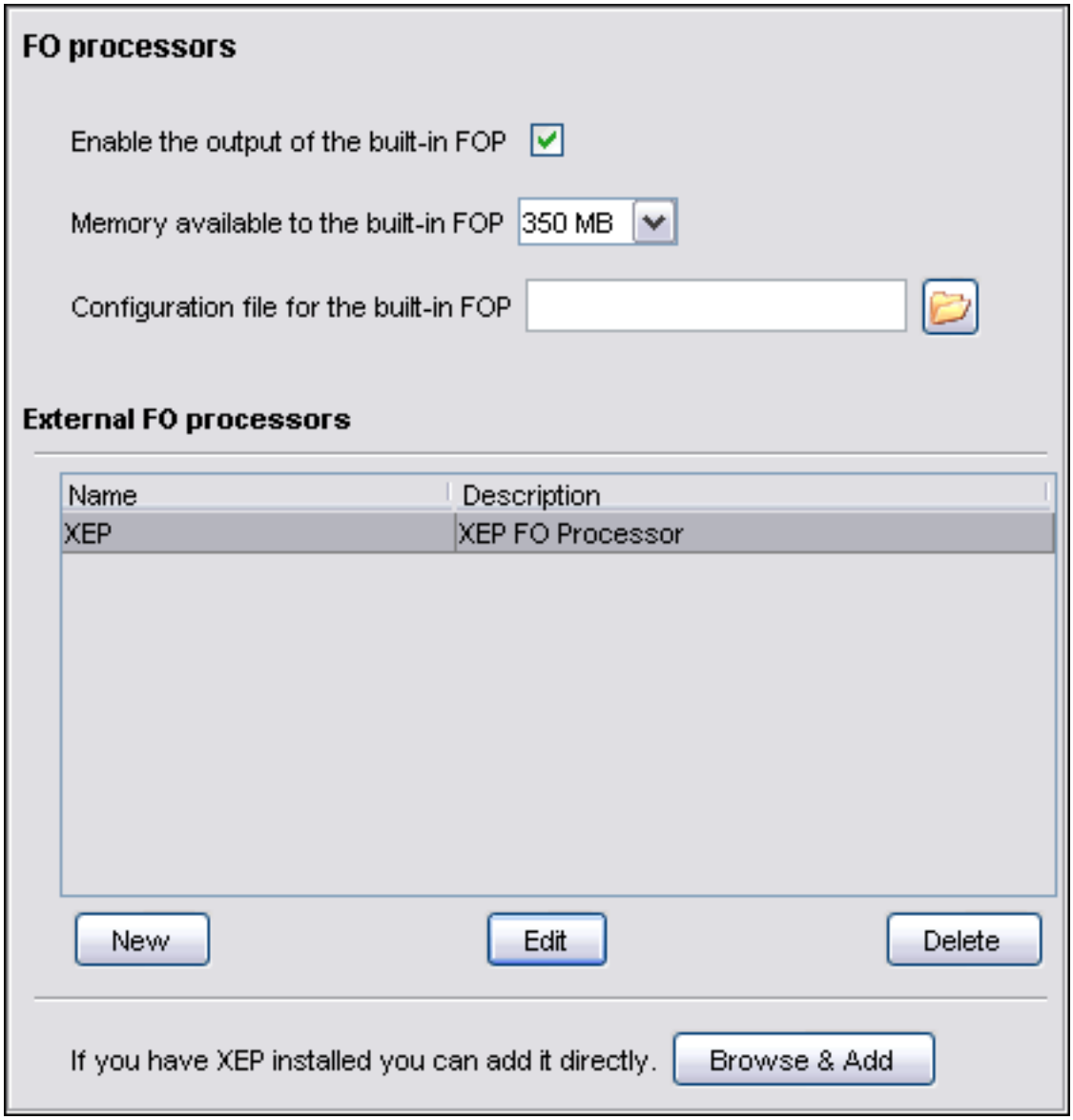
The following settings are available:

Enable XHTML output	Enable or disable rendering of output to the XHTML Output document View during the transformation process. For performance issues, it is advisable to disable XHTML output for large jobs. Also, the XHTML area is only able to render XHTML documents. In order to view the output result of other formats, such as HTML, save the Text output area to a file and use the required external browser for viewing.
Infinite loop detection	Set this option to receive notifications when an infinite loop occurs during transformation.
Maximum depth in templates stack	How many templates (`<xsl:templates>`) instructions can appear on the current stack. This setting is used by the infinite loop detection.

FO processors

Besides the built-in formatting objects processor (Apache FOP) the user can use other external processors. <oXygen/> has implemented an easy way to add XEP as external FO processor if the user has the XEP installed.

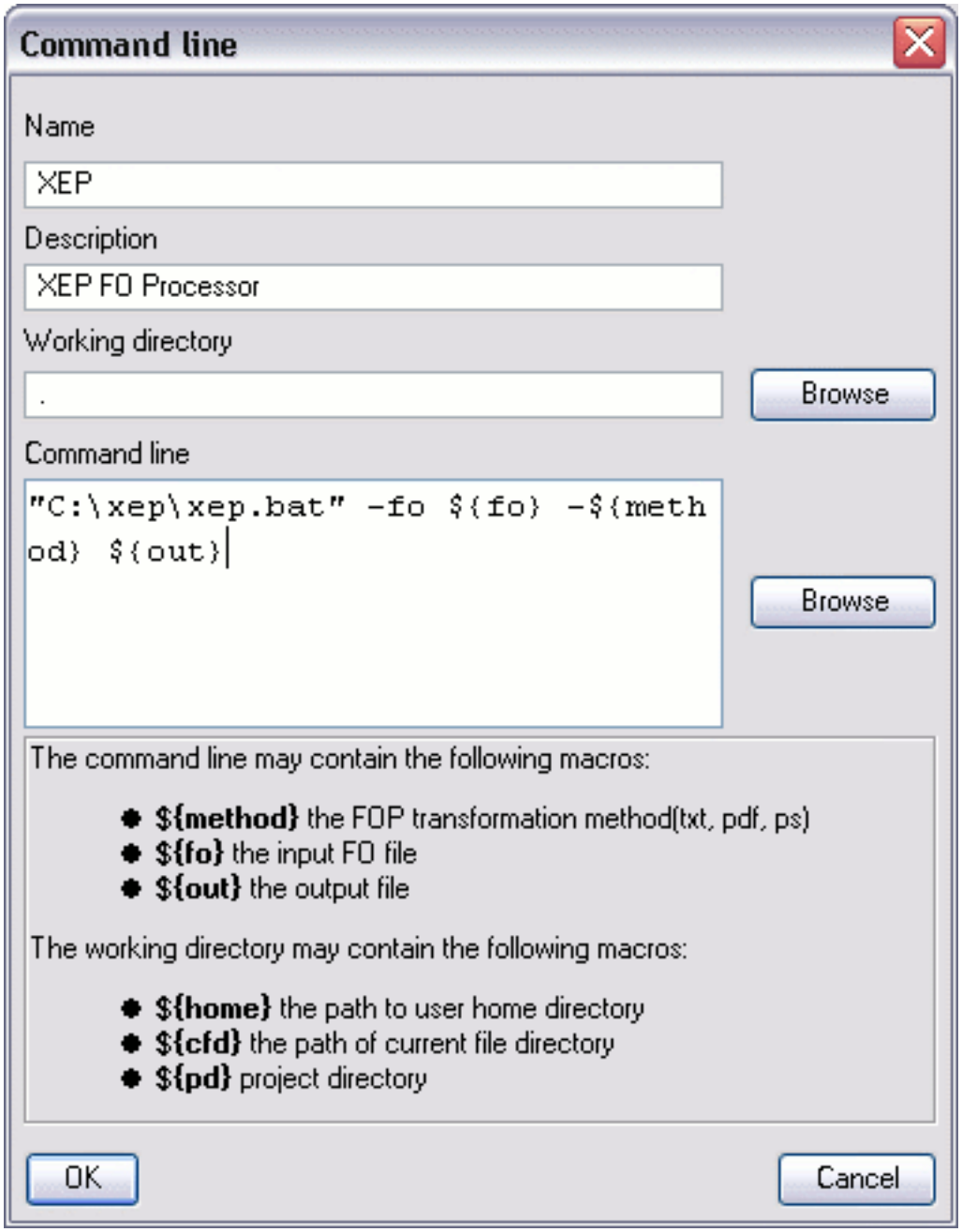
Figure 3.12. The FO processors pane



Enable the output of the built-in FOP	When checked all FOP output will be displayed in a results pane at the bottom of the editor window including warning messages about FO instructions not supported by FOP.
Memory available to the built-in FOP	If your FOP transformations fail with an "Out of Memory" error select from this combo box a larger value for the amount of memory reserved for FOP transformations.
Configuration file for the built-in FOP	You should specify here the path to a FOP configuration file, necessary for example to render to PDF using a special true type font a document containing Unicode content.

The users can configure the external processors for use with <oXygen/> in the following dialog.

Figure 3.13. Configure the external processors



Name	The name that will be displayed in the list of available FOP processors on the FOP tab of the Transforming Configuration dialog.
Description	The description of the FO processor displayed in the Preferences->FO Processors option.
Working directory	The directory in which the intermediate and final results of the processing

	will be stored.
Command line	The command line that will start the FO processor, specific to each processor.

XPath

Figure 3.14. The XPath pane

Perform XPath over XSLT source	If checked <oXygen/> will evaluate XPath expresion over XSLT documents, otherwise the XPath expresion is applied on the XML document selected in the eventual transformation scenario associated with the XSLT document.

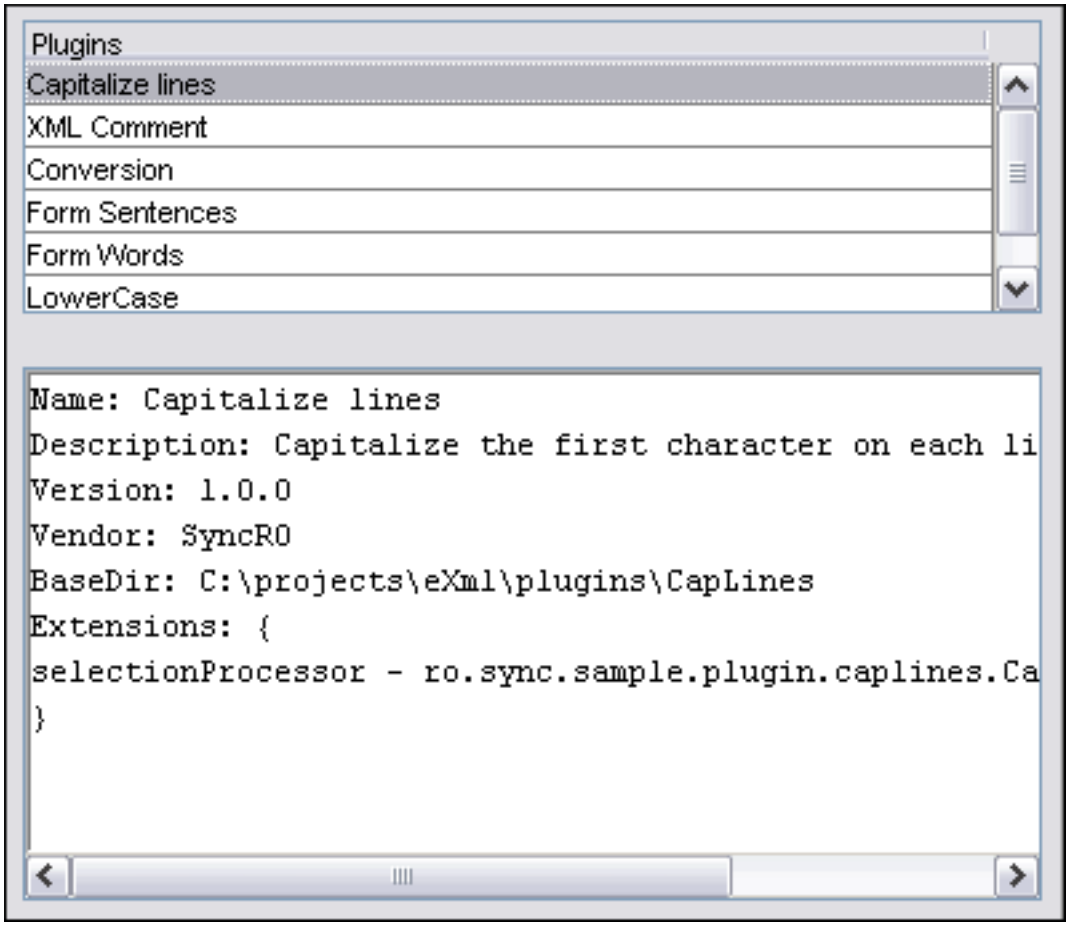
Plugins

<oXygen/> provides the ability to add plugins that extend the functionality of the application. The plugins are shipped as separate packages; check for new plugins on <oXygen/> site: http://www.oxygenxml.com.

One plugin consists of a separate sub-folder in the Plugins folder in the <oXygen/> installation folder. This sub-folder must contain a valid plugin.xml in accordance with the plugin.dtd file from the Plugins folder.

<oXygen/> automatically detects and loads plugins correctly installed in the Plugins folder and displays them in the Plugin option from the Preferences dialog.

Figure 3.15. The Plugins pane

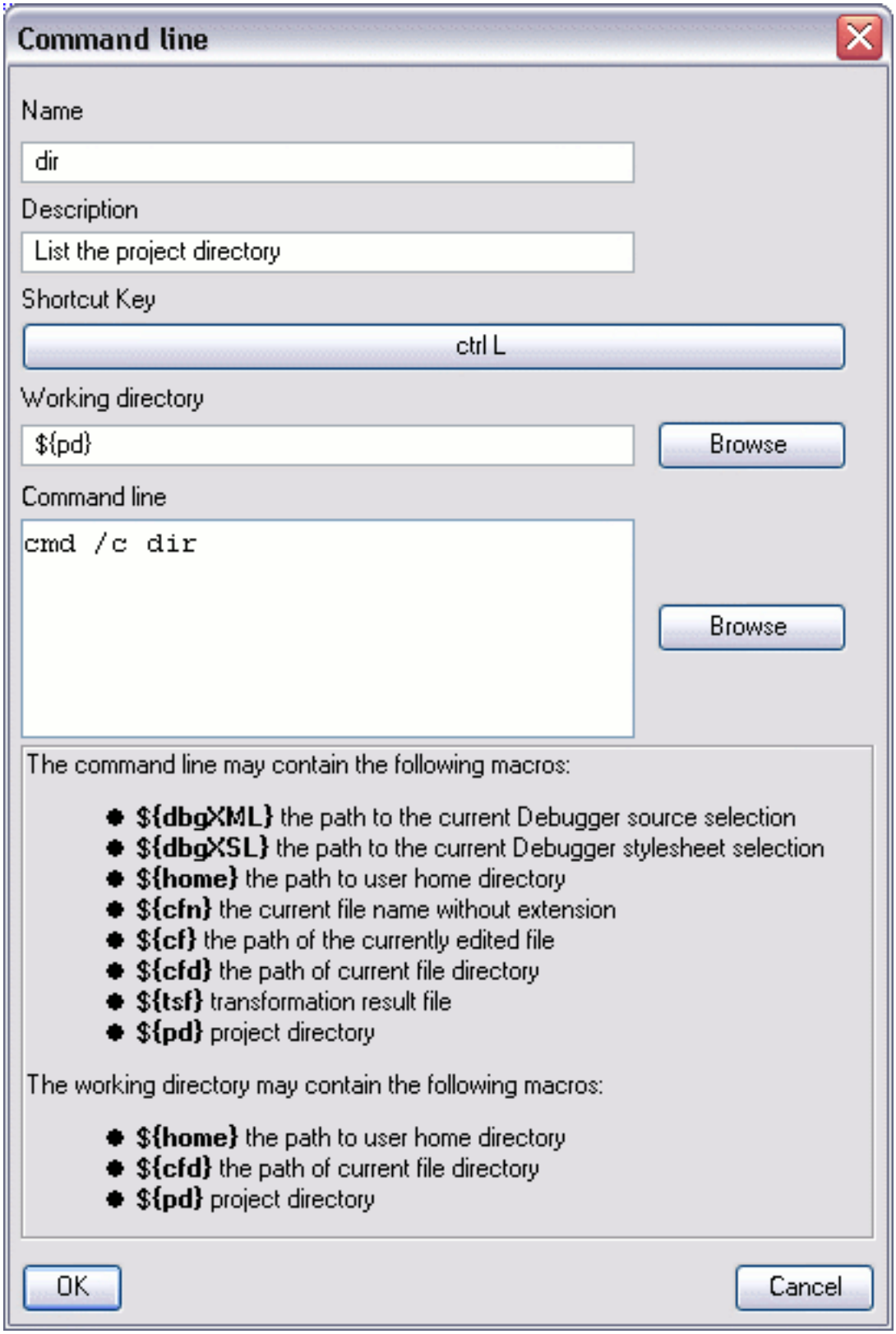


A short description of the plugin can be obtained with a click on the plugin name.

External Tools

The user can run within <oXygen/> other tools as if from the command line of the operating system shell. The configuration of such a tool is done in the following dialog.

Figure 3.16. Configure External Tools

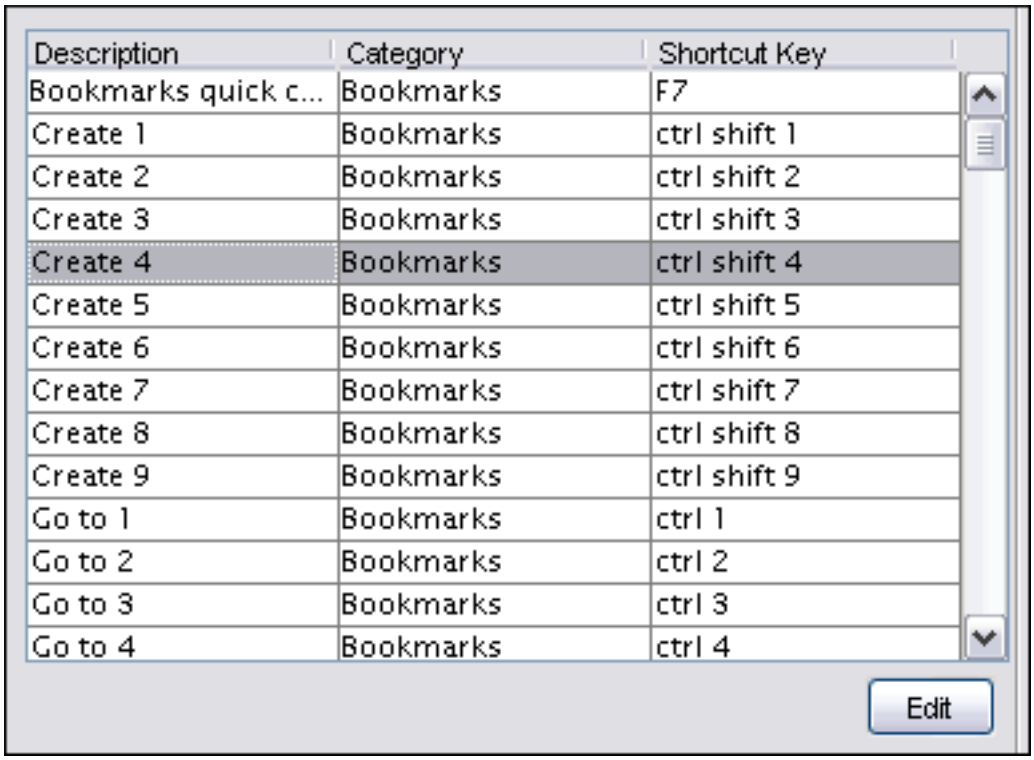

Command line
Name
dir
Description
List the project directory
Shortcut Key
ctrl L
Working directory
${pd}
Browse
Command line
cmd /c dir
Browse
The command line may contain the following macros:
${dbgXML} the path to the current Debugger source selection
${dbgXSL} the path to the current Debugger stylesheet selection
${home} the path to user home directory
${cfn} the current file name without extension
${cf} the path of the currently edited file
${cfd} the path of current file directory
${tsf} transformation result file
${pd} project directory
The working directory may contain the following macros:
${home} the path to user home directory
${cfd} the path of current file directory
${pd} project directory
OK
Cancel

Name	The name of the menu entry corresponding to this tool that will be displayed in the External Tools menu and in the external tools combo box on the toolbar.
Description	The description of the tool displayed in the Preferences->External Tools option.
Shortcut key	The keyboard shortcut that launches the external tool.
Working directory	The directory the external tool will use to store intermediate and final results.
Command line	The command line that will start the external tool.

Menu shortcut keys

The user can configure in one place all the keyboard shortcuts of the menu items available in <oXygen/>. The current shortcuts assigned to menu items are displayed in the following table.

Figure 3.17. The Menu shortcut keys pane



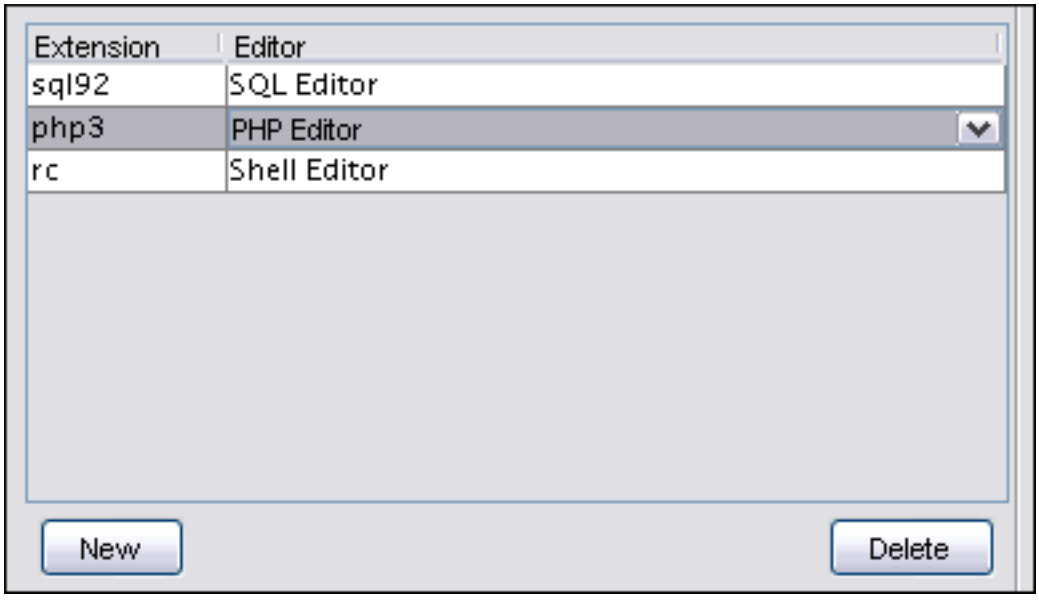
Description	A short description of the menu item operation.
Category	The shortcuts are classified in categories for easier management. For example the "Cut" operation for the source view is distinguished from the tree view one by assigning it to a separate category.

Shortcut key — The keyboard shortcut that launches the operation. Double-clicking on a table row or pressing the "Edit" button allows the user to register a new shortcut for the operation displayed on that row.

File Types

<oXygen/> offers support for a wide variety of file types, but users are free to add new file types specified by extension and associate them with the editor type which fits better.

Figure 3.18. The File Types



Extension — The new file types.

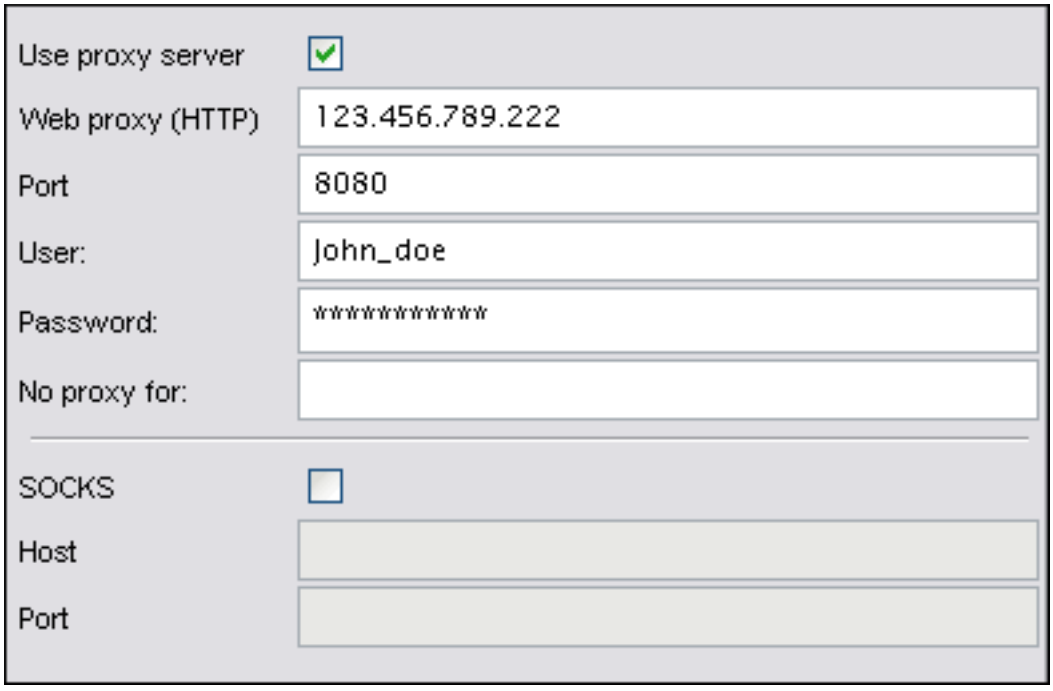
Editor — The type of editor which the extensions will be associated with. Some editors provide easy access to frequent operations via toolbars (e.g. XML editor, XSL editor, DTD editor) while other provide just a syntax highlight scheme (e.g. Java editor, SQL editor, Shell editor, etc.)

Proxy Configuration

Some networks use Proxy servers to provide Internet Services to LAN Clients. Clients behind the Proxy may therefore, only connect to the Internet via the Proxy Service. The Proxy Configuration dialog enables this configuration. If you are not sure whether your computer is required to use a Proxy server to connect to the Internet or the values required by the Proxy Configuration dialog, please consult your Network Administrator.

Open the Proxy Configuration dialog by selecting Options->Preferences->Proxy Configuration.

Figure 3.19. The Proxy Configuration Dialog



Complete the dialog as follows:

Use proxy server	When checked enables <oXygen/> to use the specified Proxy Server. When unchecked, Proxy Server is disabled.
Web Proxy (HTTP)	The IP address or Fully Qualified Domain Name (FQDN) of the Proxy Server.
Port	The TCP Port Number, normally set to 80 or 8080.
User	The Name of the user if required. Can be empty.
Password	The Password for authentication. Can be empty.
No proxy for	Specify domains for which no proxy should be used.
SOCKS	When checked enables SOCKS using the specified host and port for the server. When unchecked, SOCKS is disabled.
Host	The SOCKS host you wish to connect to.
Port	The SOCKS port you wish to connect to.

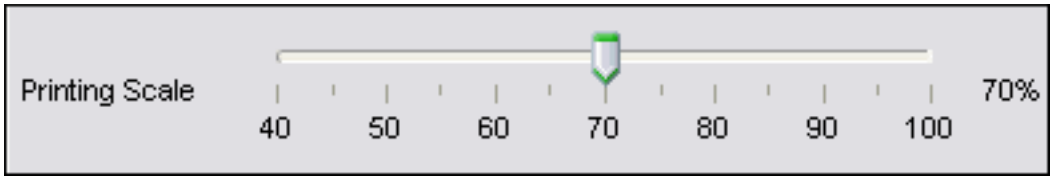
Results / Output

Figure 3.20. The Results/Output option

Maximum number of lines — This option sets the maximum number of lines of the output console where the external tools place their output.

Print

Figure 3.21. The Printing Scale option



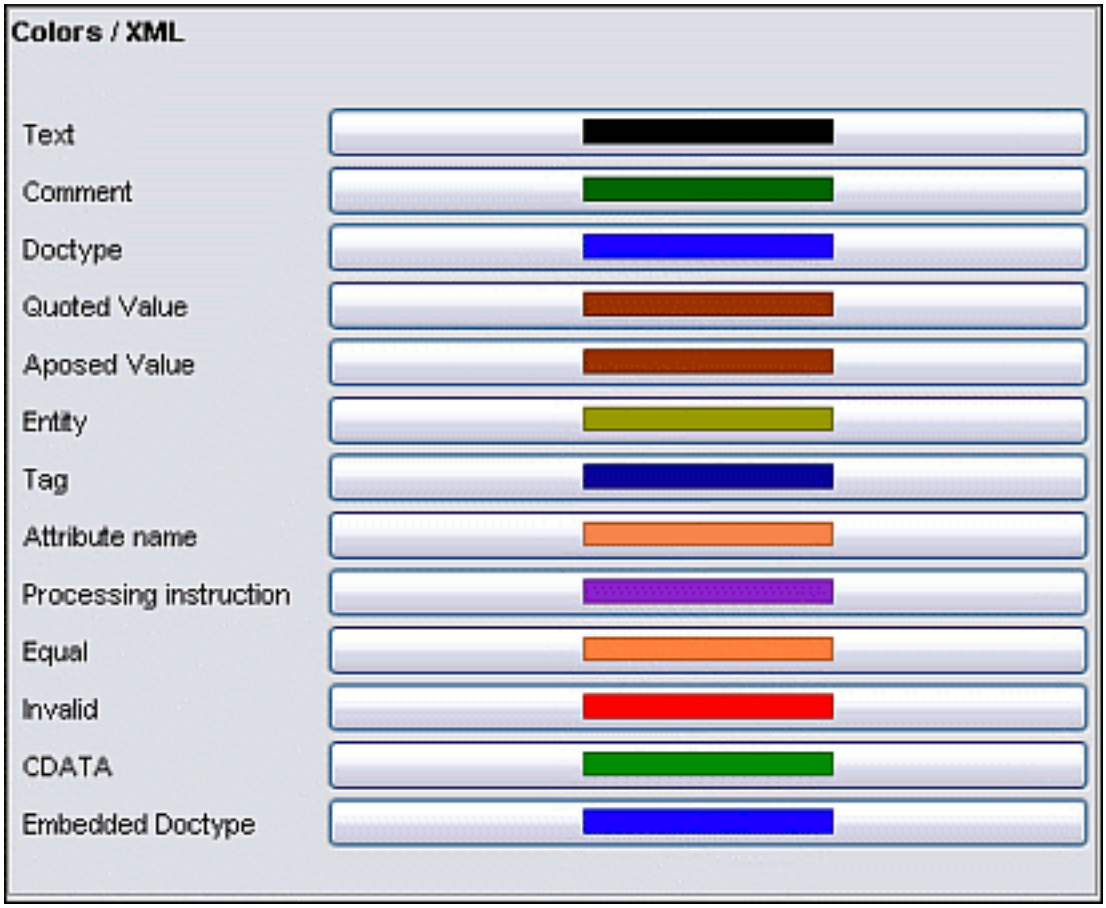
It is sometimes useful to print out the contents of a document on paper. <oXygen/> allows you to adjust the scale of the print output to make it easier to read on a page.

Printing Scale — Displays a slide allowing the user to adjust the printing scale between 40% and 100%.

Colors

<oXygen/> supports Syntax Highlight for XML, DTD, Relax NG (XML and Compact Syntax), Java, JavaScript, XQuery, C++, C, PHP,CSS, Perl, Properties, SQL, Shell and Batch documents. While <oXygen/> provides a default color configuration for highlighting the tokens, you may choose to customize, as required, using the Colors dialog.

Figure 3.22. The Colors pane



Open the Colors dialog by selecting Options->Preferences->Colors and choose one of the supported Document Types. Each document type contains a set of Tokens. When the Document Type is selected the associated tokens are listed. Selecting a token displays the current color properties and enables you to modify them.

Use Swatch, HSB or RGB models from the Color Dialog to define the color properties.

Modifications are saved when the OK button is clicked. Cancel discards changes. Reset button changes the color to the default value.

Swatches	Displays a color pallete containing a variety of colors from across the color spectrum and shades thereof. Select a color.
HSB	Hue, Saturation and Brightness (HSB) enables you to specify a color by describing it using hue, saturation and brightness.
RGB	Red, Green and Blue (RGB) enables you to specify a color using triplets of red, green and blue numbers.
Preview	Displays the color properties of the current token and results of customization.

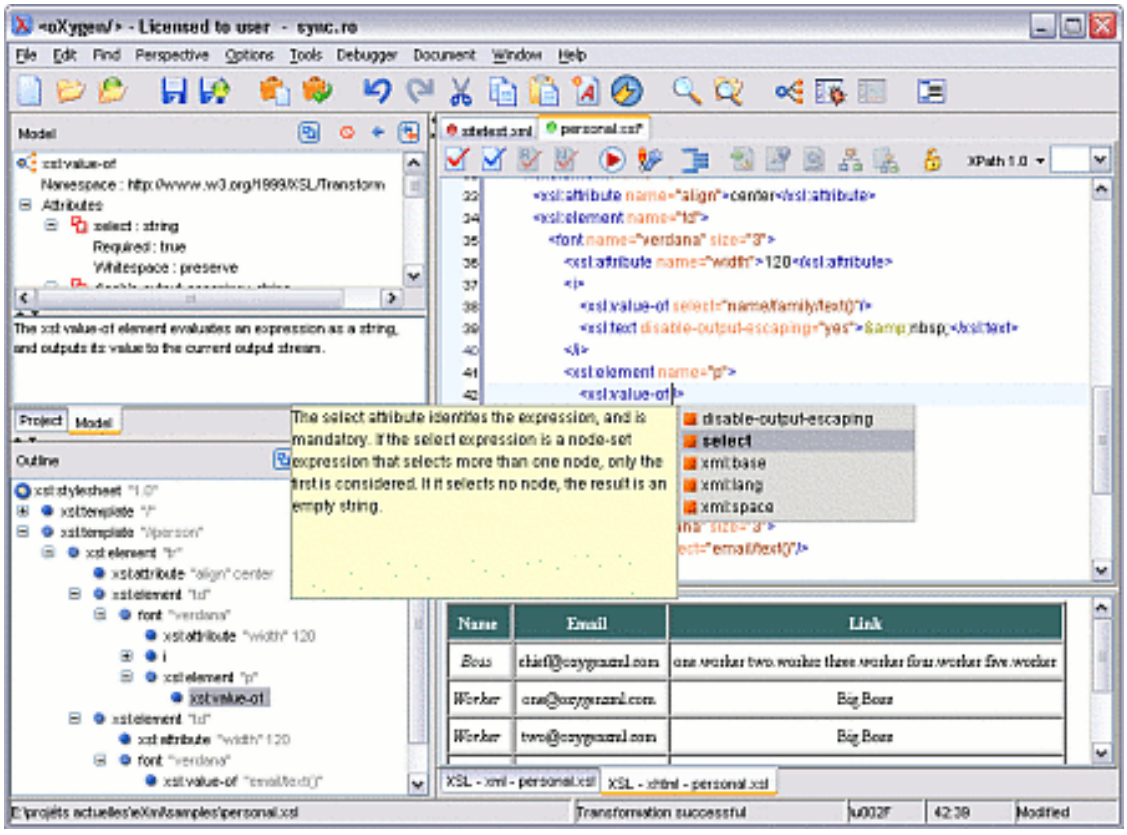
Main Window

The <oXygen/> interface uses standard interface conventions and components to provide a familiar and intuitive editing environment across all operating systems.

The main window is central to the work process. In addition to the title bar, which shows the <oXygen/> program icon, name, license information, the main window is comprised of the following main components:

- The Main Menu
- The Main Toolbar
- The Project Panel
- The Model View Panel
- The Editor Panel
- The Message Panel
- The Outliner Panel

Figure 3.23. The Main Interface



When two or more main window panels are displayed, <oXygen/> provides divider bars that allow arrangement of screen real-estate. By selecting a divider bar, it can be dragged to a new position, therefore

increasing the space occupied by one panel while decreasing it for the other.

As majority of the work process centers around the Editor panel, other panels can be hidden from view using the expand and collapse controls located on the divider bars.

Main Menu

The main menu, located below the program title bar, provides menu driven access to all the features and functions available within <oXygen/>.

- File Menu
- Edit Menu
- Find Menu
- Perspective Menu
- Options Menu
- Plugins Menu
- Tools Menu
- Debugger Menu
- Document Menu
- Window Menu
- Help Menu

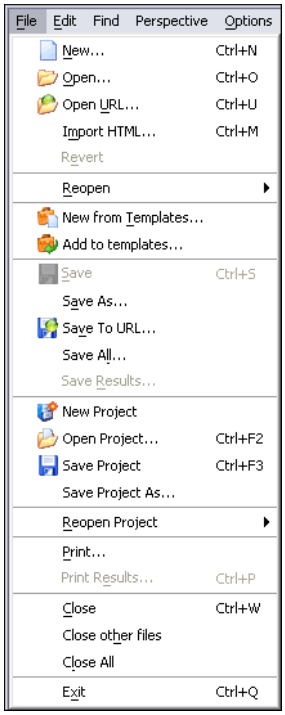
File Menu

Note

Macintosh users should use the command key instead of the control key for all keyboard short-cuts.

Table 3.1. File Menu Options

- File-> New (**Ctrl+N**) : Displays the New dialog from which to select the document file type.
- File-> Open (**Ctrl+O**) : Displays the Open dialog used to discover, select and open one or more files.
- File-> Open URL (**Ctrl+U**) : Displays the Open URL dialog used to discover, select and open one or more files using FTP/WebDAV.
- File-> Import HTML (**Ctrl+M**) : Import HTML files to XHTML 1.0 Transitional or Strict. It results an XHTML file which contains a DOCTYPE declaration referring to the XHTML DTD definition on the Web and the parsed content of the imported file



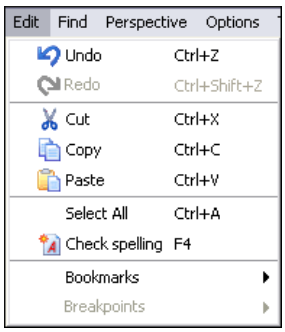
as XHTML Transitional or Strict depending on what radio button the user chose when performing the import operation.

- File->Revert: Loads the last saved file content. All unsaved modifications are lost.
- File->Reopen: Displays a list of recently opened document files. Select a file to open.
- File->New from Templates: Displays the Templates dialog used to discover, select and open a new document based on an existing template document. Template documents act as starting points that have predefined properties such as file type, prolog, root element, containers and even existing content.
- File->Add to Templates: Displays the Add Templates dialog used to define the name by which the template will be recognized in the "New from templates" option.
- File-> Save (**Ctrl+S**) : Saves the current document. If the document does not have a file, displays the "Save As" dialog.
- File->Save As: Displays the Save As dialog, used to name and save an open document to a file; or save an existing file with a new name.
- File->Save To URL: Displays the Save to URL dialog, used to name and save an open document to a file; or saves an existing file with a new name, using FTP/WebDAV.
- File->Save All: Saves all open documents. If any document does not have a file, displays the "Save As" dialog.
- File->Save Results (**Ctrl+R**) : Displays the Save Results dialog, used to save the result-list of the, currently in focus, message tab.
- File->New Project: Create a new project in the Project pane.
- File->Open Project (**Ctrl+F2**) : Displays the Open Project dialog used to discover, select and open a project file.
- File->Save Project (**Ctrl+F3 (Cmd+G on Mac)**) : Saves the current project. If the project does not have a file, displays the "Save Project As" dialog.
- File->Save Project As: Displays the Save Project As dialog, used to name and save an open project to a file; or save an existing project file with a new name.
- File->Print (**Ctrl+P**) : Displays the Page Setup dialog used to define the page size and orientation properties for printing.
- File->Print Results: Displays the Page Setup dialog used to define the page size and orientation properties for printing the result-list of the current message tab.
- File-> Close (**Ctrl+W**) : Closes only the selected tab. All other tab instances remain.
- File->Close All: Closes all opened documents. If a document is modified or has no file, a prompt to save, not to save, or cancel the save operation is displayed.
- File->Reopen Project: Displays a list of recently opened project files. Select a file to open.
- File->Exit (**Ctrl+Q**) : Terminates the <oXygen/> XML Editor. Session information such as the current Project, open Documents and Option settings is made persistent.

When the <oXygen/> editor is re-opened, the persistence information returns to the last saved state.

Edit Menu

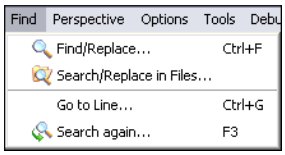
Table 3.2. Edit Menu Options



- Edit->Undo (**Ctrl+Z**) : Reverses, a maximum of 100, editing actions to return to the preceding state.
- Edit->Redo (**Ctrl+Shift+Z**) : Recreates, a maximum of 100, editing actions that were undone by the "Undo" function.
- Edit->Cut (**Ctrl+X**) : Removes the current selected node from the document and places it in the clipboard.
- Edit->Copy (**Ctrl+C**) : Places a copy of the current selection in the clipboard.
- Edit->Paste (**Ctrl+V**) : Places the current clipboard content into the document at the cursor position.
- Edit->Select All (**Ctrl+A**) : Selects the entire body of the current document, including whitespace preceding the first and following the last character.
- Edit->Check Spelling (**F4**) : Checks the spelling in your document.
- Edit->Bookmarks : Placing bookmarks and quickly accessing bookmarked positions in edited documents.

Find Menu

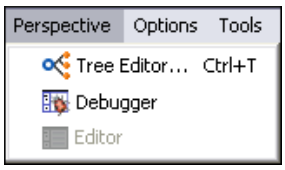
Table 3.3. Find Menu Options



- Find->Find/Replace... (**Ctrl+F**) : Displays the Find/Replace dialog, used to define "search for" or "search for and replace" operations on the current document. The replace operation can bind Perl 5-like regexp group variables ($1, $2, etc.) from the find match.
- Find->Search/Replace in Files... (**Ctrl+F**) : Displays the Search/Replace in Files dialog, used to define "search for" or "search for and replace" operations across a number of files. The replace operation can bind Perl 5-like regexp group variables ($1, $2, etc.) from the search match.
- Find->Go to Line (**Ctrl+G (Cmd+L on Mac)**) : Displays the Go to Line dialog used to move the cursor directly to the line number specified.
- Find->Search again (**F3**) : Performs another search using the last search configuration.

Perspective Menu

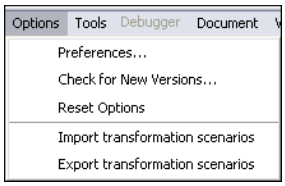
Table 3.4. Perspective Menu Options



- Tree Editor... (**Ctrl+T**) : This option opens the Tree View window.
- Debugger : This option opens the Debugger perspective.
- Editor : This option opens the Editor perspective.

Options Menu

Table 3.5. Options Menu Options



- Preferences: Includes all configuration options necessary to customize <oXygen/> for your requirements and network environment.
- Check for New Versions: Uses the version checker to query the <oXygen/> Web Site for new versions.
- Reset Options: Reverts all custom user settings within <oXygen/> to the installation defaults.
- Import transformation scenario: Load a properties file with scenarios.
- Export tansformation scenario: Store all the scenarios in a separate file , a properties file.

Plugins Menu

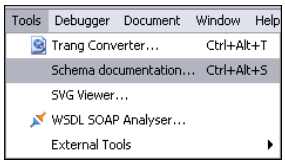
Table 3.6. Plugins Menu Options

Contains the list of the general type plugins detected in the Plugins sub-directory of the installation folder. If there are no general plugins configured the menu will not be displayed.

Tools Menu

Table 3.7. Tools Menu Options

- Trang Converter... (**Ctrl+Alt+T**) : Converts the current document to a supported

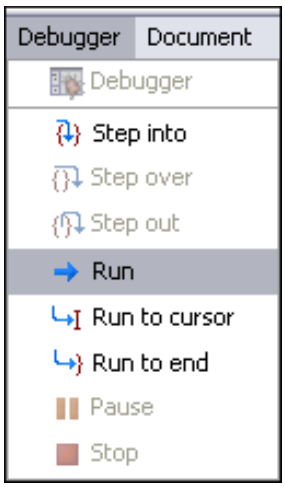


grammar language using the integrated Trang converter.

- Schema documentation... (**Ctrl+Alt+S**) : A tool used to generate HTML documentation for an XML Schema document.
- SVG Viewer : Contains a viewer for Scalable Vector Graphics file types.
- WSDL SOAP Analyser : Contains a SOAP analyser and sender for Web Services Description Language file types.
- External Tools : Contains the Preferences for the External Tools and the list of the external tools configured in the Preferences->External Tools option.

Debugger Menu

Table 3.8. Debugger Menu Options

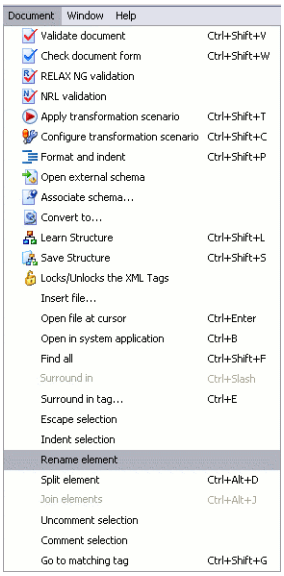


- Debugger : This option opens the Debugger perspective.
- Debugger actions : The descriptions of the debugger actions can be found in Control Toolbar section.

Document Menu

Table 3.9. Document Menu Options

- Document->Validate document (**Ctrl+Shift+V**): Executes the Validation operation on the current document using a validating parser. Returns an error result-list in the Message panel. Mark-up of current document is checked to conform with the specified DTD, XML Schema or Relax NG schema rules. For validation against Relax NG schemas the XML document must include a processing instructions of the form: [<?oxygen RNGSchema="URL-of-schema" type="xml (or compact)">]
- Document->Check document form (**Ctrl+Shift+W**): Executes the XML Form check operation on the current document using a non-validating parser. Returns an error result-list in the Message panel.
- Document->Relax NG Validation: Displays the RELAX NG Validation dialog, used



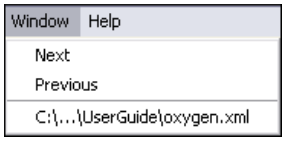
to select the Relax NG schema and to execute the Validation operation on the current document using the selected Relax NG Schema. Returns an error result-list in the Message panel. Mark-up of current document is checked to conform with the specified RNG Schema rules.

- Document->Apply transformation scenario (**Ctrl+Shift+T**): Executes the transformation process using the configuration properties defined in the Configure Transformation dialog.
- Document->Configure transformation scenario (**Ctrl+Shift+C**): Displays the Configure Transformation dialog, used to define properties for conversion of documents to multiple output targets. Also enables saving of scenarios. Each scenario, can store a unique configuration ready to be used in the future.
- Document->Format and Indent (**Ctrl+Shift+P**): Also referred to as "Pretty Print", "Format and Indent" performs layout functions to make mark-up easier to read on screen and in print output.
- Document->Learn Structure (**Ctrl+Shift+L**): Reads the mark-up structure of the current document so that it can be saved as a template using the Save Structure option.
- Document->Save Structure (**Ctrl+Shift+S**): Displays the Save Structure dialog, used to name and create DTD documents learnt by the "Learn Structure" function.
- Document->Lock/Unlock XML Tags: When in "Lock" mode tags cannot be edited. Unlock to enable tag editing.
- Document->Insert file...: Inserts in a file under the current position of the caret in the current document.
- Document->Open file at cursor: Opens in a new panel the file with the name under the current position of the caret in the current document.
- Document->Surround in <tag> (**Ctrl+/**): Selected Text in the editor is marked with start and end tags of the last 'Surround in' action.
- Document->Surround in tag... (**Ctrl+E**): Selected Text in the editor is marked with the specified start and end tags.
- Document->Escape Selection: Replaces the <, >, &," ' symbols with <, >, &, ", '.
- Document->Indent selection: Performs layout functions to the current selection to make mark-up easier to read on screen and in print output.
- Document->Rename element: The element from the caret position and the elements that have the same name as the current element can be renamed according with the options from the Rename dialog.
- Document->Split element (**Ctrl+Alt+D**): Split the element from the caret position in two identical elements. The caret must be inside the element
- Document->Join elements (**Ctrl+Alt+J**): Joins the left and the right elements relative to the current caret position. The elements must have the same name, attributes and attributes values.
- Document->Uncomment selection: Removes <!-- --> comment brackets from currently selected comment.

- Document->Comment selection: Places <!-- --> comment brackets around current selection or cursor position.
- Document->Go to the matching tag (**Ctrl+Shift+G**): Moves the cursor to the end tag that matches the start tag, or vice versa.

Window Menu

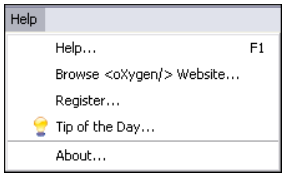
Table 3.10. Window Menu Options



- Window-> Next: Traverses the Editor focus across the list of open documents from left to right.
- Window-> Previous: Traverses Editor focus across the list of open documents from right to left.
- Open Files List: Displays a list of documents currently open in the Editor panel.

Help Menu

Table 3.11. Help Menu Options



- Help->Help (**F1**): Opens the <oXygen/> XML Editor Help Window.
- Help->Browse <oXygen/> web site :Displays the <oXygen/> home page using the default Web Browser. The <oXygen/> e-Store provides information about licensing costs and provides facilities for secure online transactions.
- Help->Register :Allows the user to enter the license text from the registration mail.
- Help->Tip of the day: Randomly selects an <oXygen/> productivity tip and displays it in the Tip of the Day dialog.
- Help->About : Displays the About dialog, which provides information about the <oXygen/> version number and license status.

Main Toolbar

The Main toolbar, located below the Main menu, provides easy access to common and frequently used functions. Each icon is a button that acts as a shortcut to a related function. Hold the pointer/cursor over an icon to display a context label that will give you a hint about its functionality. Click an icon to use its function.

Figure 3.24. The Main Toolbar Buttons

Table 3.12. Description of Main Toolbar Buttons

	File-> New (**Ctrl+N**): Displays the New dialog from which to select the document file type.
	File-> Open (**Ctrl+O**): Displays the Open dialog used to discover, select and open one or more files.
	File-> Open URL (**Ctrl+U**): Displays the Open URL dialog used to discover, select and open one or more files using FTP/WebDAV.
	File-> Save (**Ctrl+S**): Saves the current document. If the document does not have a file, displays the "Save As" dialog.
	File->Save To URL : Displays the Save URL dialog, used to name and save an open document to a file; or save an existing file with a new name, using FTP/WebDAV.
	File->New from Templates :Displays the Templates dialog used to discover, select and open a new document based on an existing template document. Template documents act as starting points that have predefined properties such as file type, prolog, root element, containers and even existing content.
	File->Add to Templates : Displays the Add Templates dialog used to define the name by which the template will be recognized in the "New from templates" option.
	Edit->Undo (**Ctrl+Z**): Reverses, a maximum of 100, editing actions to return to the preceding state.
	Edit->Redo (**Ctrl+Shift+Z**): Recreates, a maximum of 100, editing actions that were undone by the "Undo" function.
	Edit->Cut (**Ctrl+X**): Removes the current selection from the document and places it in the clipboard.
	Edit->Copy (**Ctrl+C**): Places a copy of the current selection in the clipboard.
	Edit->Paste (**Ctrl+V**): Places the current clipboard content into the document at the cursor position.
	Edit->Check Spelling (**F4**): Checks the spelling of the current document.
	Edit->Bookmarks (**F7**) -> Quick creation: Places a bookmark at current caret position.
	Find->Find/Replace (**Ctrl+F**): Displays the Find/Replace dialog, used to define "search for" or "search for and replace" operations on the current document.
	Find-> Search Replace in Files : Displays the Search/Replace in Files dialog, used to define "search for" or "search for and replace" operations across a number of files.
	Perspective->Tree Editor... (**Ctrl+T**): Displays the Tree View window.
	Perspective->Debugger : Opens the debugger perspective.

	Perspective->Editor : Opens the editor perspective.
	Transparency Contrast Adjuster->Transparency : Adjusts the contrast.

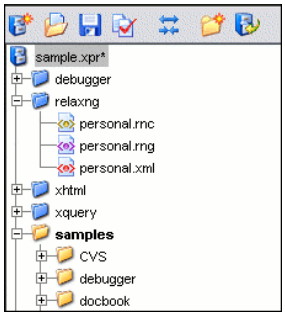
Project Panel

The Project panel, located on the left-hand side of the Main window, enables the definition of projects and logical management of the documents it contains.

The tree of documents displayed in this panel supports batch validation: one or more files in the current project can be validated at once by selecting the "Validate selection" option on the tree's context menu, and all project files can be validated with a single click on the "Validate all project files" button on the toolbar.

If a project folder contains many documents a certain document can be quickly located in the project tree if the user selects with the mouse the folder containing the desired document (or some arbitrary document in this folder) and types the first characters of the document name. The desired document will be automatically selected as soon as the typed characters uniquely identify its name in the folder. Once selected the document can be opened by pressing the ENTER key or by double-clicking on it or it can be deleted by pressing the DELETE key or by choosing "Remove File" from the context menu.

Table 3.13. Description of Project Panel



The Project Panel is comprised of:

- The Project Toolbar
- The Project View

Project Toolbar

The Project toolbar, located on the top of the Project panel, provides easy access to <oXygen/> project functions. Each icon is a button that acts as a shortcut to a related function. Hold the pointer over an icon to display a context label that will give you a hint about its functionality. Click an icon to use its function.

Figure 3.25. The Project Toolbar

Table 3.14. Description of Project Toolbar Buttons


File->New Project :Create a new project in the Project pane.


File->Open Project (**Ctrl+F2**): Displays the Open Project dialog used to discover, select and open a project file.


File->Save Project (**Ctrl+F3**): Saves the current project. If the project does not have a file, displays the "Save Project As" dialog.


Validate all: Validate all project files.


Show/Hide Path Option: Toggles the file path ON and OFF for all files displayed in the Project View.

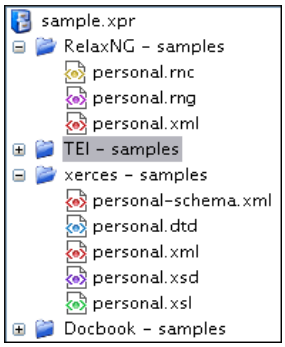

New Folder: Displays the New Folder dialog, used to specify the name of a folder about to be created under the existing and selected Project or Folder in the Project View.


Add File : Includes the current document into the selected folder in the Project View.

Project View

The Project view provides a method for logical management and visualization of files associated with a project. Once a project is opened, saved files can be added to or removed from the project. Either operations can be performed while the file is opened in the Editor Panel directly from a local or remote file system.

Table 3.15. Description of Project View



To help organize files, virtual folders can be created. Virtual folders only exist within the project file, providing a logical method for visual organization. In addition to internal creation of Virtual Folders, you can also import system folders and their contents from local or remote file systems. The result is a tree-view that allows navigation and easy file management within the project, regardless of the files' physical location.

All folders, whether created internally or imported, have no physical connection with a local or remote file system. i.e. If you add, delete or rename a Project folder those operations are only performed in the <oXygen/> project file. The same principle is applicable to files, with the exception that the representation of a file in the Project View does maintain a backward link to the physical file. i.e. Actions such as "Remove" only removes the file from the current project file, it does not delete the file from the file system. Double-clicking a file is equivalent to File-> Open (**Ctrl+O**) .

The full path to file is hidden by default. Click the Show/Hide Path button to toggle the file path on or off.

The default target when adding files to a project is the project root. Selecting a folder changes the target to the selected folder. Files may have multiple instances, within the folder system but cannot appear twice within the same folder.

Right-clicking any object in the tree-view displays the Project menu with functions that can be performed on, or from the selected object. Options available from the Project menu are specific to the object type selected in the tree-view.

The project menu contains a list of actions that is dependent on the selected item. The description of all available actions follows.

Table 3.16. Description of Project Menu

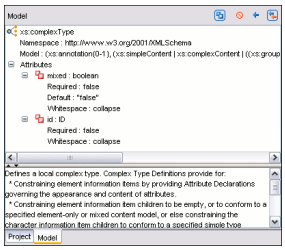
- New Folder: Displays the New Folder dialog, used to specify the name of a folder about to be created under the existing and selected Project or Folder in the Project View.
- Rename Folder: Enables modification to the selected project folder name.
- Delete Folder: Removes the selected folder and its contents from the Project View.
- Add Files : Displays the Add File dialog, used to discover and select files to include in the Project View.
- Add File : Includes the current document into the selected folder in the Project View.
- Import Folder: Displays the Import Folder dialog, used to add folders and their contents to the Project View.
- Import Remote Folders : The user may choose to import a folder that exists on a WebDAV or FTP server. The project will have a logical structure similar to the content of the remote folder. The files from all the remote subfolders are added into the correspondent logical folders.
- Link to external folder : The user may create link folders in the project tree. A link folder has a correspondent in the filesystem. The project will present the structure of that folder as it exists on the disk.
- Refresh : Use this option to refresh the content of the folders that are children of a link to an external folder.
- Open all Files: Opens all files associated with the current project. If a file appears twice, only the first instance is opened.
- Open: Opens the selected file.
- Remove File: Removes the selected file from the Project View.
- Validate Selection: Validates the selected file or all the files from the selected folder.
- Validate selection with...: Validates the selected file or all the files from the selected folder with a specified schema.
- Search/Replace in Files: Searches for a string in the project selected files or in the entire project.

Model View Panel

The Model View panel, located on the left-hand side of the Main window, eases the editing effort by presenting the structure of the currently edited tag and additional tag documentation.

The Model View, combined with the powerful Outliner is an useful tool in providing the user with spacial and insight information on the edited document.

Table 3.17. Description of Model View Panel



The Model View Panel is comprised of:

- The Element Structure View.
- The Annotation View.

The Element Structure View.

The Element Structure View shows the structure of the current edited or selected tag in a Tree format.

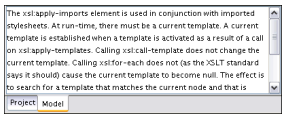
Table 3.18. Description of the Element Structure View

The information includes the name, model and attributes the currently edited tag may have. The allowed attributes are shown along with any restrictions they might possess.

The Annotation View.

The Annotation View shows the annotations that are present in the used schema for the currently edited or selected tag.

Table 3.19. Description of the Annotation View



This information can be very useful to persons learning XML because it has small available definitions for each used tag.

Editor Panel

The Editor panel is the place where you spend most of your time, reading, editing, applying markup and checking the validity and form of your documents.

The Editor panel is comprised of:

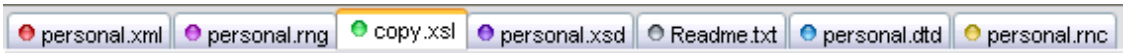
- Editor Document Tabs
- Editor Toolbar
- Editing Space

- Editor Status Bar

Editor Document Tabs

The <oXygen/> editor uses the tabbed page mode to display open documents. The document tab is located above the Editor Toolbar and shows the file type, file name and its edit status.

Figure 3.26. The Editor Document Tabs



Modified files are denoted with an "*"; saving the document removes this status until the next edit operation.

Each open document has an associated tab displaying this information. To distinguish which document is currently under edit, the active document is highlighted and brought to the foreground. Inactive document tabs are shadowed in the background. Selecting a tab changes the document focus to display its contents for editing. Hovering the pointer over a tab displays the file's full path in a tool tip.

Right-click on any tab to display the Document Tab menu.

Document Tab Menu

- Close: Closes only the selected tab. All other tab instances remain.
- Close other files: Closes the other files except the selected tab.
- Close All: Closes all open documents. If a document is modified or has no associated file, a prompt to save, not to save, or cancel the save operation is displayed.
- Add to project: Includes the current document into the selected folder in the Project View.
- Add all to project: Includes all open documents into the selected folder in the Project View.

Document File Types

The file type is denoted by an icon preceding the filename. While this information can be obtained from the file extension, visual identification of style types is improved using a color legend that correlates to the Project View icons:

RED	XML Documents
LIGHT BLUE	DTD Documents
DARK PURPLE	XSD Documents
PURPLE	Relax NG Schema - XML
YELLOW	Relax NG Schema - Compact
YELLOW	NRL Schema

GREEN	XSL Documents
MAROON	WSDL Documents
RED	XQuery Documents
WHITE	TXT Documents

Editor Toolbar

The Editor toolbar, located below the Document Tabs, provides easy access to <oXygen/> editing functions. Each icon is a button that acts as a shortcut to a related function. Hold your the pointer/cursor over an icon to display a context label that will give you a hint as to its function. Click an icon to use its function.

Figure 3.27. The Editor Toolbar Buttons

Table 3.20. Description of Editor Toolbar Buttons

	Document->Validate document (**Ctrl+Shift+V**): Executes the Validation operation on the current document using a validating parser. Returns an error result-list in the Message panel. Mark-up of current document is checked to conform with the specified DTD, XML Schema or Relax NG schema rules.
	Document->Check document form (**Ctrl+Shift+W**): Executes the XML Form check operation on the current document using a non-validating parser. Returns an error result-list in the Message panel.
	Document->Relax NG Validation: Displays the Relax NG Validation dialog, used to select the Relax NG schema and to execute the Validation operation on the current document. The schema file can be either in Relax NG XML syntax or in Relax NG compact syntax. In case of errors the validation returns an error result-list in the Message panel.
	Document->NRL Validation: Displays the NRL Validation dialog, used to select the NRL (Namespace Routing Language) schema and to execute the Validation operation on the current document. In case of errors the validation returns an error result-list in the Message panel.
	Document->Convert to ...: Converts the current document to a supported grammar language using the integrated Trang converter.
	Document->Associate Schema ...: Associates a schema with the current document.
	Document->Apply transformation scenario (**Ctrl+Shift+T**): Executes the transformation process using the configuration properties defined in the Configure Transformation dialog.
	Document->Configure transformation scenario (**Ctrl+Shift+C**): Displays the Configure Transformation dialog, used to define properties for conversion of documents to multiple output targets. Also enables saving of scenarios. Each scenario can store an unique configuration ready to be used in the future.
	Document->Format and Indent (**Ctrl+Shift+P**): Also referred to as "Pretty Print", "Format and Indent"

performs layout functions to make mark-up easier to read on screen and in printed output.

Document->Learn Structure (**Ctrl+Shift+L**): Reads the mark-up structure of the current document so that it can be saved as a dtd file using the Save Structure option.

Document->Save Structure (**Ctrl+Shift+S**): Save the learned document structure to an external dtd file.

Lock/Unlock XML Tags: When in "Lock" mode tags cannot be edited. Unlock to enable tag editing.

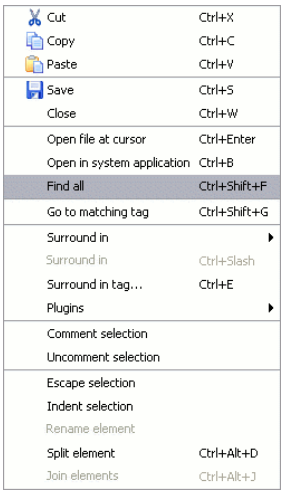
XPath: Executes an XPath expression, used to select specified elements within the current document. <oXygen/> can execute expressions according to XPath version 1.0 or 2.0.

Editing Space

The <oXygen/> editor provides a useful set of features that promote easy editing of structure mark-up documents. Features such as automatic syntax highlighting and checking, intelligent Tag-Insight, end tag auto completion help improve productivity and reduce errors. Text attributes such as font, formatting, coupled with customizable coloring options for syntax highlighting provide a flexible and powerful structured mark-up editor.

Right-click anywhere in the editor to display a menu with options related to editing and mark-up operations.

Table 3.21. Description of Editor Space Menu



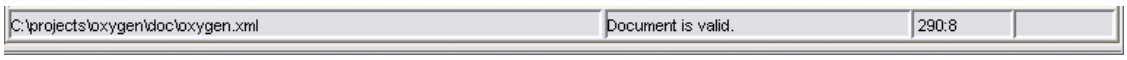
- Edit->Cut (**Ctrl+X**): Removes the current selection from the document and places it in the clipboard.
- Edit->Copy (**Ctrl+C**): Places a copy of the current selection in the clipboard.
- Edit->Paste (**Ctrl+V**): Places the current clipboard content into the document at the cursor position.
- File-> Save (**Ctrl+S**): Saves the current document. If the document does not have a file, displays the "Save As" dialog.
- File-> Close (**Ctrl+W**): Closes only the selected tab. All other tab instances remain.
- Document->Open file at cursor: Opens in a new panel the file with the name under the current position of the caret in the current document.
- Document->Find All (**Ctrl+Shift+F**): Finds all occurencies of selected word in current file.
- Document->Go to the matching tag (**Ctrl+Shift+G**): Moves the cursor to the end tag that matches the start tag, or vice versa.
- Surround in: Displays a sub-menu with mark-up elements arranged in alphabetical order. Selected Text in the editor is marked with start and end tags of the selected element.
- Document->Surround in <tag> (**Ctrl+/**): Selected Text in the editor is marked with start and end tags of the last 'Surround in' action.

- Document->Surround in tag... (**Ctrl+E**): Selected Text in the editor is marked with the specified start and end tags.
- Plug-ins: Displays a sub-menu with the list of the selection type plugins that can be launched in the context of the document currently edited.
- Document->Comment selection: Places <!-- --> comment brackets around current selection or cursor position.
- Document->Uncomment selection: Removes <!-- --> comment brackets from currently selected comment.
- Document->Escape Selection : Replaces the <, >, &, ", ' symbols with <, >, &, ", '.
- Document->Indent selection: Performs layout functions to the current selection to make mark-up easier to read on screen and in printed output.
- Document->Rename element: The element from the caret position and the elements that have the same name as the current element can be renamed according with the options from the Rename dialog.
- Document->Split element (**Ctrl+Alt+D**): Split the element from the caret position in two identical elements. The caret must be inside the element
- Document->Join elements (**Ctrl+Alt+J**): Joins the left and the right elements relative to the current caret position. The elements must have the same name, attributes and attributes values.

Editor Status Bar

The Editor status bar, located below the Editing space, provides information about the current document:

Figure 3.28. The Editor Status Bar



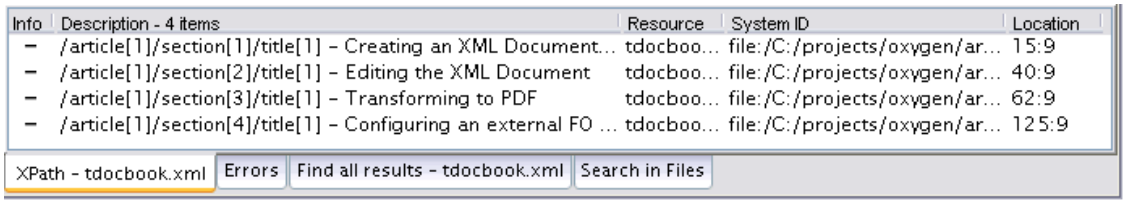
The Document Path Pane	Displays the full path of the current document.
The Message Pane	Displays the status of Validate XML, Check XML Form, Apply Transformation, XPath expression evaluation and Format and Indent (Pretty Print) operations and the result (success or not) and number of errors found in the last operation.
The Cursor Coordinates Pane	Displays the current position of the cursor in terms of Line Number: Column Number.
The Edit Status Pane	Indicates whether or not the document has been modified since the last save.

Message Panel

The Message panel, located across the bottom half of the Main Window, displays messages returned from the following operations:

- Validate XML
- Check XML Form
- Transformation
- XPath Expressions Evaluation
- Find/Replace
- Search/Replace in Files

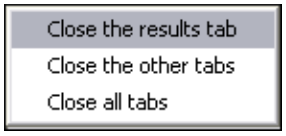
Figure 3.29. The Message Panel



To distinguish between result types, each functions result-list is displayed in a separate tab within the panel.

To change focus between the result-lists, just select the tab required. To close a tab, or all tabs, right-click the tab to display a menu and select an option.

Table 3.22. Message Panel Tab Menu

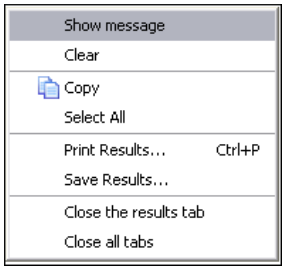


- Close the Results Tab: Closes the selected Results Tab.
- Close the Other Tabs: Closes the other tabs except the selected tab.
- Close All Tabs: Closes all open tabs within the panel.

In all instances, each record within a result-list is linked to the document location where the indicated error or character string will be found. Selecting a result list record moves the Editors cursor to, and highlights the object in question. If the target document is closed, it will be opened in the Editor panel.

In addition to this time saving feature, the Message panel menu enables several operations to be performed on single or multiple selected records. The Message panel menu is displayed by selecting a record, then pressing right-click.

Table 3.23. The Message Panel Menu



- Edit->Copy (**Ctrl+C**): Places a copy of the current selection in the clipboard.
- Edit->Select All (**Ctrl+A**): Selects the entire body of the current document, including whitespace preceding the first and following the last character.
- File->Print (**Ctrl+P**): Displays the Page Setup dialog used to define the page size and orientation properties for printing.
- File->Save Results (**Ctrl+R**): Displays the Save Results dialog, used to save the result-list of the currently selected message tab.
- Close Results Tab: Closes the selected Results Tab.
- Close the other Tabs : Closes all the other opened Results Tabs.
- Close all Tabs: Closes all open tabs within the panel.

Validate and Check XML Parser Errors

The "Validate XML " and "Check XML Form" functions use the same tab instance named "errors" to display their results. As both functions use the same tab instance, "Validate XML " and "Check XML Form" result-lists overwrite each another. Each time one of these functions is executed the result-list is populated with new results.

XPath Messages

When working with XPath expressions, <oXygen/> outputs errors and node results to the message panel.

Transformation Output Messages

During transformation processes, XML, XHTML or FOP, the message panel is used to display parser errors, warning messages and output of XML or XHTML. Each is displayed in a separate tab. Tab instances for XML/XHTML are created on a per document basis, error and warning tabs are simply repopulated.

Find/Replace Results

The "Find/Replace" function opens a new tab instance for each document on which the function is run. Each tab instance is named "Find all results - <filename.ext>". Each time that the function is run the result-list in the documents associated tab instance is populated with new results.

Search Replace in files Messages

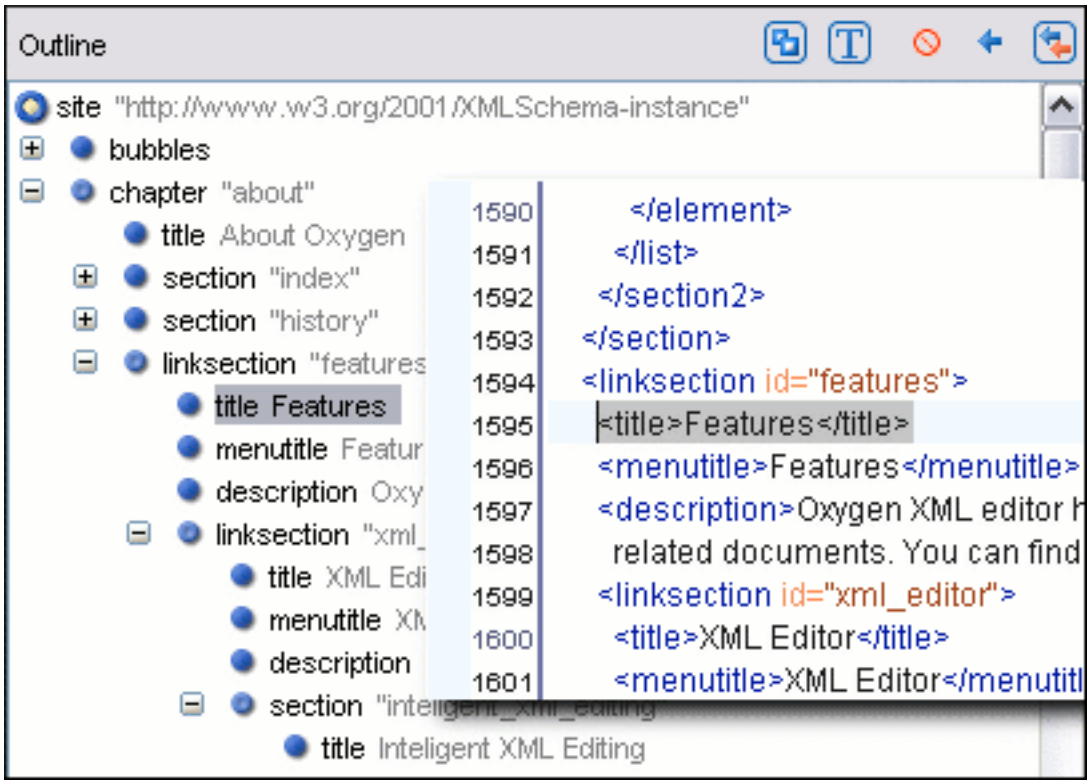
The "Search/Replace in Files" function uses a single tab instance named "Search in Files" to display results. It does not share the tabs with any other function, repopulating the same tab instance each time the function is executed.

Outliner Panel

The Outliner panel, located in the bottom left part of the Main Window has the following available functions:

- XML Document Overview
- Modification Follow-up
- Document Tag Selection

Figure 3.30. The Outliner Panel



XML Document Overview

The Outliner displays a general tag overview of the current edited XML Document. It also shows the correct hierarchical dependencies between the tag elements, making it easier for the user to be aware of the document's structure and the way tags are nested.

Modification Follow-up

When editing, the Outliner dynamically follows the modifications introduced by the user, showing in the middle of the panel the node which is currently being modified. This gives the user better insight on to where in the document he/she is positioned and how the structure of the document is affected by his/hers modifications.

Document Tag Selection

The Outliner can also be used to search for a specific tag's location and contents in the edited document. Intuitively, by selecting with the left mouse button the desired tag in the Outliner Panel, the document is scrolled to the position of the selected tag. Moreover, the tag's contents are selected in the document, making it easy to notice the part of the document contained by that specific tag and furthermore to easily

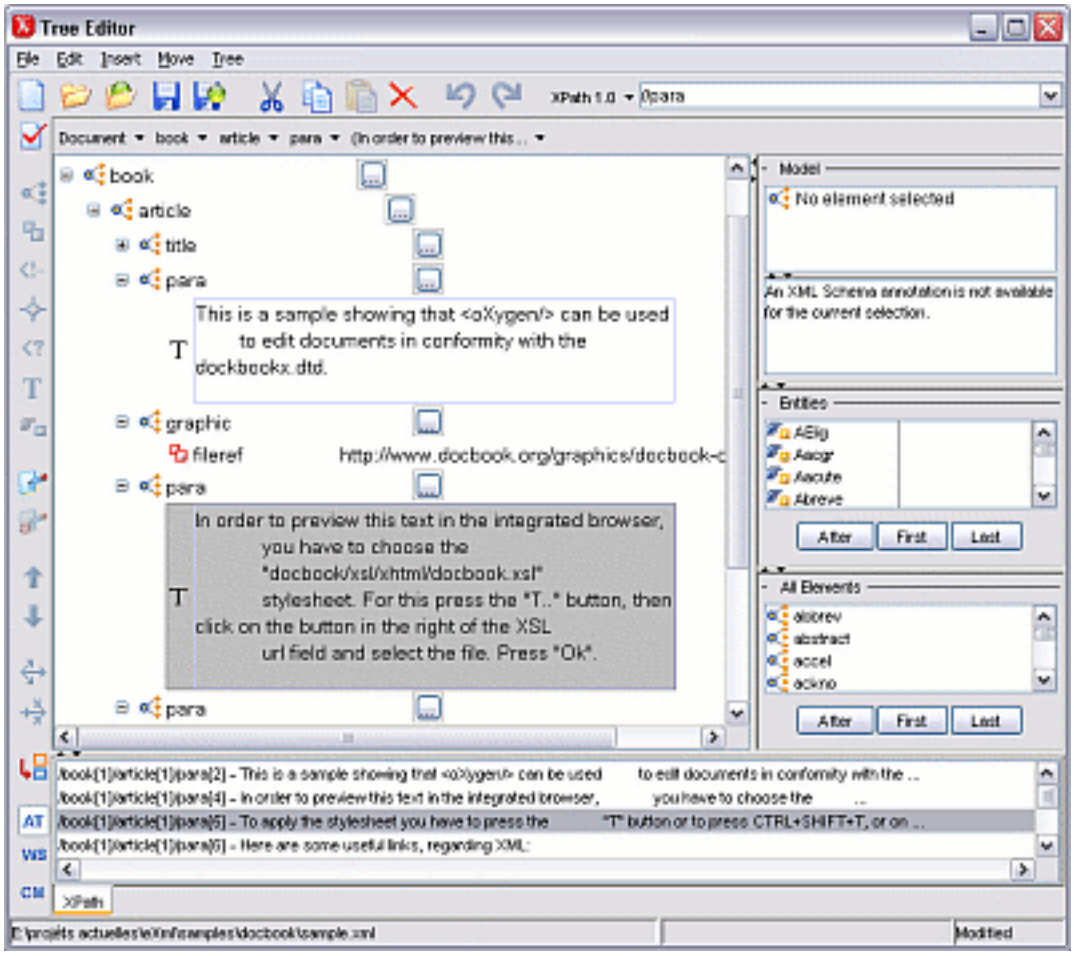
copy and paste the tag's contents in other parts of the document or in other documents.

Tree View Window

The Tree View window is comprised of the following main components:

- The Main Menu
- The Main Toolbar
- The Editor Panel
- The Message Panel
- The Information Panel
- The Entities Panel
- The All Elements Panel

Figure 3.31. The Main Interface



Main Menu

The main menu, located below the program title bar, provides menu driven access to all the features and functions available within <oXygen/> Tree View.

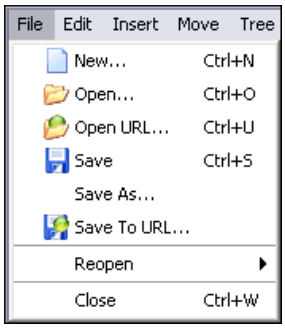
- File Menu
- Edit Menu
- Insert Menu
- Move Menu

File Menu

Note

Macintosh users should use the command key instead of the control key for all keyboard shortcuts.

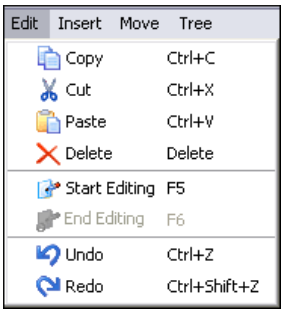
Table 3.24. File Menu Options



- File-> New (**Ctrl+N**) : Creates a new empty document and displays it in the Tree View Editor.
- File-> Open (**Ctrl+O**) : Displays the Open dialog used to discover, select and open a file to be edited.
- File-> Open URL (**Ctrl+U**) : Displays the Open URL dialog used to discover, select and open one or more files using FTP/WebDAV.
- File-> Save (**Ctrl+S**) : Saves the current document. If the document does not have a file, displays the "Save As" dialog.
- File->Save As: Displays the Save As dialog, used to name and save an open document to a file; or save an existing file with a new name.
- File->Save To URL: Displays the Save to URL dialog, used to name and save an open document to a file; or save an existing file with a new name, using FTP/WebDAV.
- File->Reopen: Displays a list of recently opened document files. Select a file to open.
- File-> Close (**Ctrl+W**) : Closes the Tree View Editor.

Edit Menu

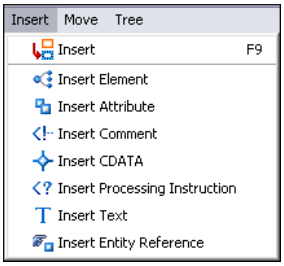
Table 3.25. Edit Menu Options



- Edit->Copy (**Ctrl+C**) : Places a copy of the current selection in the clipboard.
- Edit->Cut (**Ctrl+X**) : Removes the current selected node from the document and places it in the clipboard.
- Edit->Paste (**Ctrl+V**) : Places the current clipboard content into the document at the cursor position.
- Edit->Delete (**Delete**) : Delete the selected node from the document.
- Edit->Start Editing (**F5**) : Starts editing the selected node from the document.
- Edit->End Editing (**F6**) : Ends editing the selected node.
- Edit->Undo (**Ctrl+Z**) : Reverses, a maximum of 100, editing actions to return to the preceding state.
- Edit->Redo (**Ctrl+Shift+Z**) : Recreates, a maximum of 100 editing actions that were undone by the "Undo" function.

Insert Menu

Table 3.26. Insert Menu Options

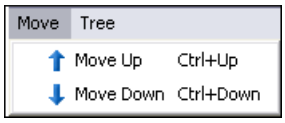


- Insert-> Insert (**F9**) : Insert a new node of the same type like the selected one as its sibling.
- Insert Element: Insert an Element after the selected node as its last child.
- Insert Attribute: Insert an Attribute after the selected node as its last attribute.
- Insert Comment: Insert a Comment after the selected node as its last child.
- Insert CDATA: Insert a CDATA after the selected node as its last child.
- Insert Processing Instruction: Insert a Processing Instruction after the selected node as its last child.
- Insert Text: Insert a Text after the selected node as its last child.
- Insert Entity Reference: Insert a Entity after the selected node as its last child.

Move Menu

Table 3.27. Move Menu Options

- Move->Move Up (**Ctrl+Up**) : Move up the selected node with one position.



- Move->Move Down (**Ctrl+Down**) : Move down the selected node with one position.

Main Toolbar

The Main toolbar, located below the Main menu, provides easy access to common and frequently used functions. Each icon is a button that acts as a shortcut to a related function. Hold the pointer over an icon to display a context label that will give you a hint as to its function. Click an icon to use its function.

Figure 3.32. The Main Toolbar Buttons

Table 3.28. Description of Main Toolbar Buttons

	File-> New (**Ctrl+N**): Displays the New dialog from which to select the document file type.
	File-> Open (**Ctrl+O**): Displays the Open dialog used to discover, select and open a file.
	File-> Open URL (**Ctrl+U**): Displays the Open URL dialog used to discover, select and open one or more files using FTP/WebDAV.
	File-> Save (**Ctrl+S**): Saves the current document. If the document does not have a file, displays the "Save As" dialog.
	File-> Save To URL: Displays the Save URL dialog, used to name and save an open document to a file; or save an existing file with a new name, using FTP/WebDAV.
	Edit->Copy (**Ctrl+C**): Places a copy of the current selection in the clipboard.
	Edit->Cut (**Ctrl+X**): Removes the current selection from the document and places it in the clipboard.
	Edit->Paste (**Ctrl+V**): Places the current clipboard content into the document at the current position.
	Edit->Delete (**Delete**): Delete the selected node from the document.
	Edit->Undo (**Ctrl+Z**): Reverses, a maximum of 100, editing actions to return to the preceding state.
	Edit->Redo (**Ctrl+Shift+Z**): Recreates, a maximum of 100, editing actions that were undone by the "Undo" function.

XPath: Executes an XPath expression, used to select specified elements within the current document. <oXygen/> can execute expressions according to XPath version 1.0 or 2.0.

Tree Editing Toolbar

The tree editing toolbar, located on the left side of the tree editor window, provides easy access to frequently used functions. Each icon is a button that acts as a shortcut to the corresponding function. Hold your mouse pointer over an icon to display a tooltip text that will give you a hint to its function. Click an icon to use its function.

Table 3.29. Description of Tree Editing Toolbar Buttons

	Validate document (Ctrl+Shift+V): Executes the Validation operation on the current document using a validating parser. Returns an error result-list in the Message panel. Mark-up of current document is checked to conform with the specified DTD, XML Schema or Relax NG schema rules.
	Insert->Insert - Element: Insert an element after the selected node as its last child.
	Insert->Insert - Attribute: Insert an attribute after the selected node as its last child.
	Insert->Insert - Comment: Insert a Comment after the selected node as its last child.
	Insert->Insert - CDATA: Insert a CDATA after the selected node as its last child.
	Insert->Insert - Processing Instruction: Insert a processing instruction after the selected node as its last child.
	Insert->Insert - Text: Insert a text after the selected node as its last child.
	Insert->Insert - Entity Reference: Insert an entity reference after the selected node as its last child.
	Start Editing: Starts editing the selected node from the document.
	Stop Editing: Ends editing the selected node.
	Toggle AT: This option shows/hides the attributes.
	Toggle WS: This option shows/hides the whitespaces.
	Move->Move Up (**Ctrl+Up**): This option moves up the selected node with one position.
	Move->Move Down (**Ctrl+Down**): This option moves down the selected node with one position.
	Expand All (Ctrl+P) : This option expands the selected element by one level.
	Collapse All (Ctrl+L): This option collapses the selected element by one level.
	Insert->Insert (**Insert**): Insert a new node of the same type like the selected one as its sibling.

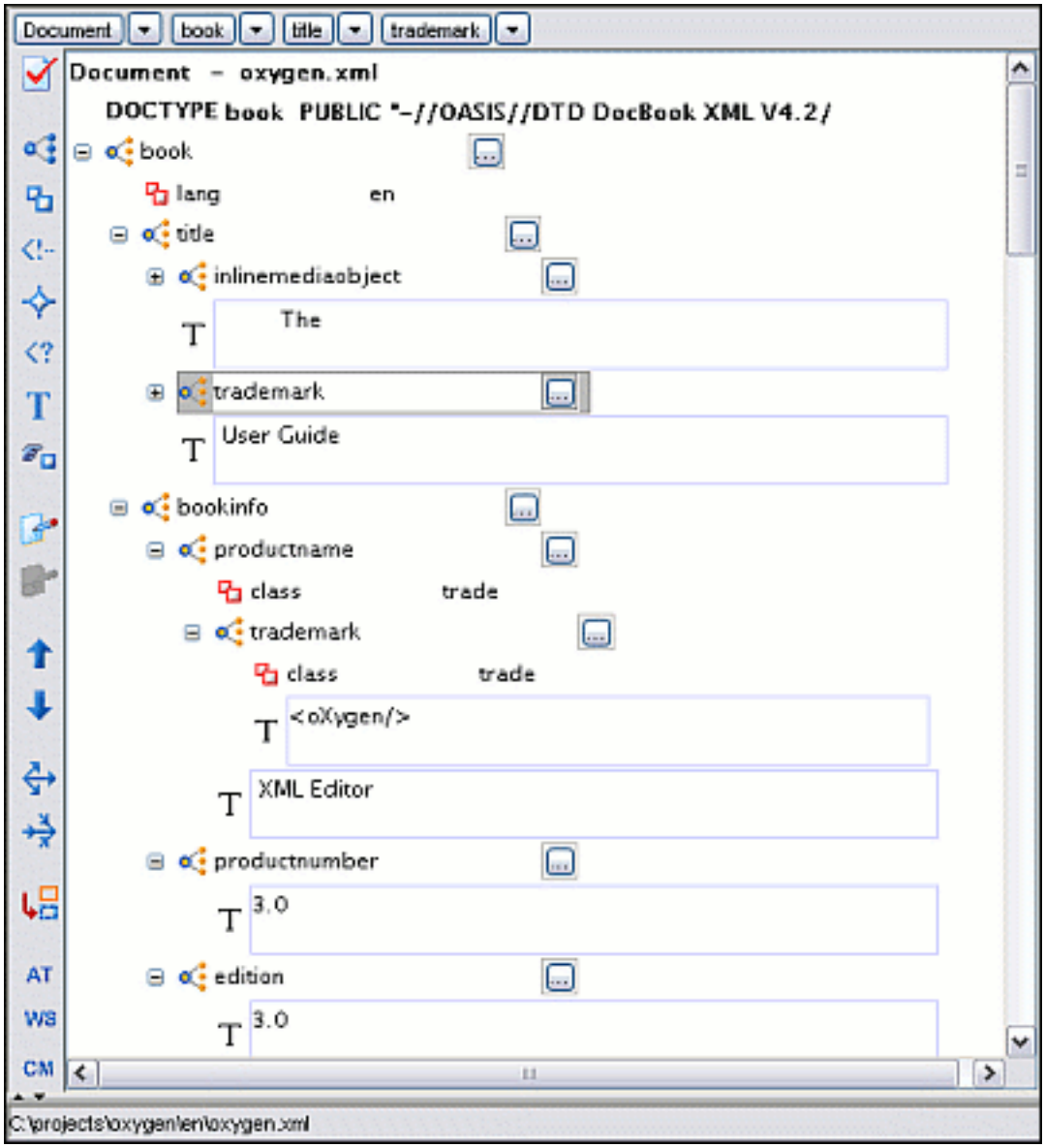
Editor Panel

The Editor panel is comprised of:

- Navigation Panel

- Editing Space
- Editor Status Bar

Figure 3.33. The Editor Panel

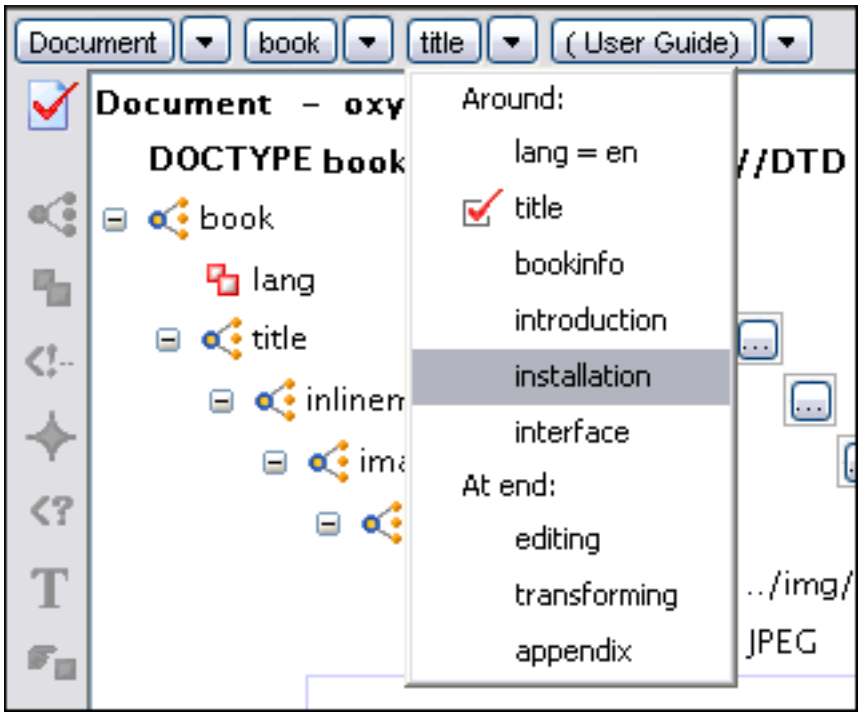


The Navigation Panel

For a quick navigation in the document you can use the navigation panel which displays each level of the document on a different column. You can easily navigate on the same level of the tree by pressing on the down arrow of a column and choosing the desired token.

The navigation panel is located above the editing space.

Figure 3.34. The Navigation Panel

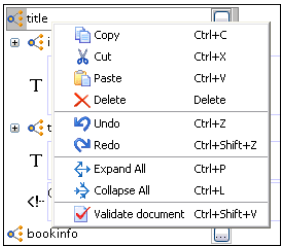


Editing Space

The Tree View Editor provides a useful set of features that promote easy editing of structure mark-up documents. Each token has associated an icon for a easy visual identification of the tokens.

Right-click anywhere in the editor to display a menu with options related to editing and mark-up operations.

Table 3.30. Description of Editor Space Menu



- Edit->Copy (**Ctrl+C**) : Places a copy of the current selection in the clipboard.
- Edit->Cut (**Ctrl+X**) : Removes the current selected node from the document and places it in the clipboard.
- Edit->Paste (**Ctrl+V**) : Places the current clipboard content into the document at the cursor position.
- Edit->Delete (**Delete**) : Delete the selected node from the document.
- Edit->Undo (**Ctrl+Z**) : Reverses, a maximum of 100, editing actions to return to the preceding state.
- Edit->Redo (**Ctrl+Shift+Z**) : Recreates, a maximum of 100, editing actions that were undone by the "Undo" function.

- Edit->Expand All (Ctrl+P) : This option expands the selected element by one level.
- Edit->Collapse All (Ctrl+L) : This option collapses the selected element by one level.
- Edit->Validate document (Ctrl+Shift+V) : Executes the Validation operation on the current document using a validating parser. Returns an error result-list in the Message panel. Mark-up of current document is checked to conform with the specified DTD, XML Schema or Relax NG schema rules.

Editor Status Bar

The Editor status bar, located below the Editing space, provides information about the current document:

Figure 3.35. The Editor Status Bar

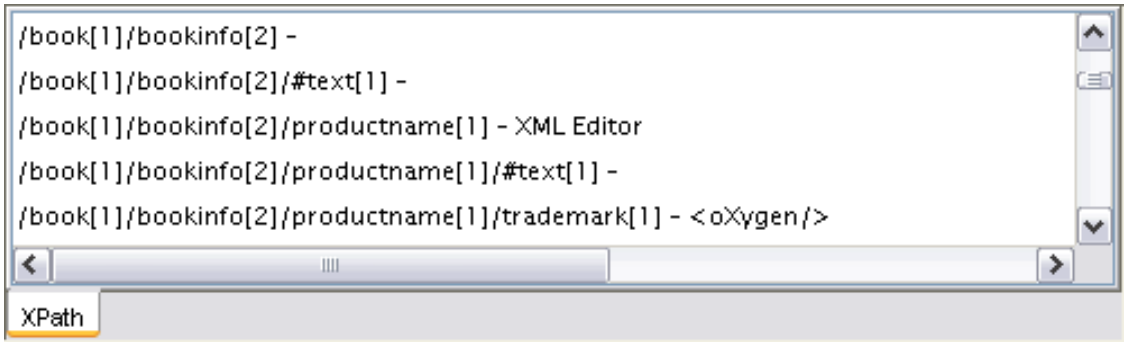
The Document Path Pane	Displays the full path of the current document.
The Message Pane	Displays the status of Validate XML and XPath expression evaluation operations and the result (success or not) and number of errors found in the last operation.
The Edit Status Pane	Indicates whether or not the document has been modified since the last save.

Message Panel

The Message panel, located across the bottom half of the Tree View Window, displays messages returned from the following operations:

- Validate XML
- XPath Expressions

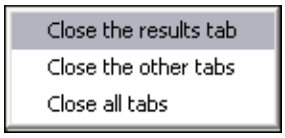
Figure 3.36. The Message Panel



To distinguish between result types, each functions result-list is displayed in a separate tab within the panel.

To change focus between the result-lists, just select the tab required. To close a tab, or all tabs, right-click the tab to display a menu and select an option.

Table 3.31. Message Panel Tab Menu

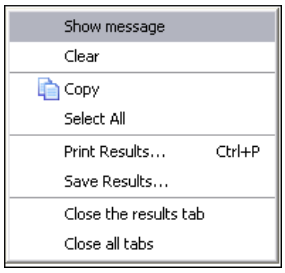


- Close Results Tab: Closes the selected Results Tab.
- Close all Tabs: Closes all open tabs within the panel.

In all instances, each record within a result-list is linked to the document location where the indicated error or character string will be found. Selecting a result list record moves the Editor selection to the object in question.

In addition to this time saving feature, the Message panel menu enables several operations to be performed on single or multiple selected records. The Message panel menu is displayed by selecting a record, then pressing right-click.

Table 3.32. The Message Panel Menu



- Edit->Copy (**Ctrl+C**): Places a copy of the current selection in the clipboard.
- Edit->Select All (**Ctrl+A**): Selects the entire content of the message panel.
- File->Print (**Ctrl+P**): Displays the Page Setup dialog used to define the page size and orientation properties for printing.
- File->Save Results (**Ctrl+R**): Displays the Save Results dialog, used to save the result-list of the, currently in focus, message tab.
- Close Results Tab: Closes the selected Results Tab.
- Close all Tabs: Closes all open tabs within the panel.

Validate XML

The "Validate XML " function use the tab instance named "errors" to display results. Each time this function is executed the result-list is populated with new results.

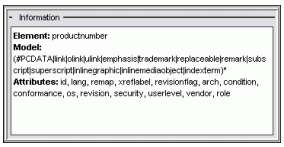
XPath Messages

When working with XPath expressions, <oXygen/> outputs errors and node results to the message panel.

Information Panel

<oXygen/> XML Editor provides a handy information panel that shows the detailed information about the attribute or element that you are working on.

Table 3.33. Elements Information



- Elements: Indicates the name of the element.
- Model: Indicates the list of elements which the selected element may contains, as it is specified in the DTD, XML Schema or Relax NG schema.
- Facets: Indicates all the facets applied to the element.
- Attributes: Indicates the list of attributes which the selected element may contain, as it is specified in the DTD, XML Schema or Relax NG schema.

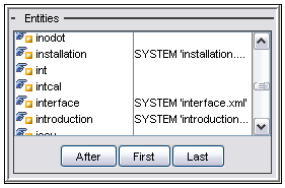
Table 3.34. Attributes Information

- Attributes: Indicates the name of the attribute.
- Model: Indicates the type of the selected attribute, as it is specified in the DTD or XML Schema.
- Facets: Indicates all the facets applied to the element.
- Value: Indicates the list of possible values which the selected attribute may contain, as it is specified in the DTD or XML Schema.
- Default: Indicates the default value of the attribute.
- Required: Indicates if the selected attribute is required.

Entities Panel

The Entities Panel presents a list of defined entities that you can insert within your document.

Table 3.35. Entities Panel

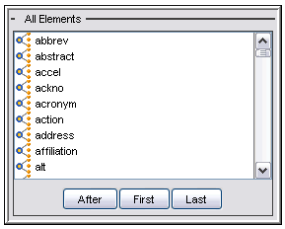


- After: Allows you to insert the selected entity into your document after the current node as next sibling.
- First: Allows you to insert the selected entity into your document as first child of the current element.
- Last: Allows you to insert the selected entity into your document as the last child of the current element.

All Elements Panel

The All Elements Panel presents a list of all defined elements that you can insert within your document.

Table 3.36. All Elements Panel



- After: Allows you to insert the selected element into your document after the current element as next sibling.
- First: Allows you to insert the selected element into your document as first child of the current element.
- Last: Allows you to insert the selected element into your document as the last child of the current element.

Chapter 4. Editing Documents

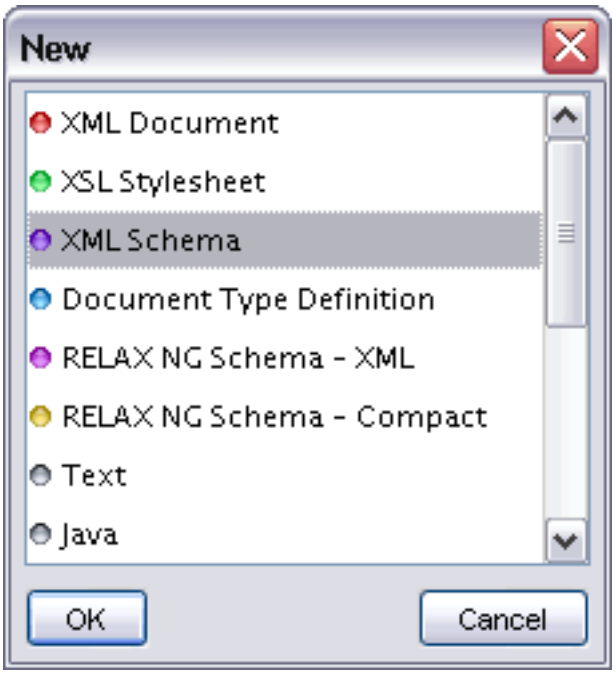
This chapter explains how to create, edit and manage files as part of a project within the <oXygen/> XML Editor.

Creating New Documents

<oXygen/> supports a number of Document Types. Use the following procedure to create documents.

Procedure 4.1. Creating new documents

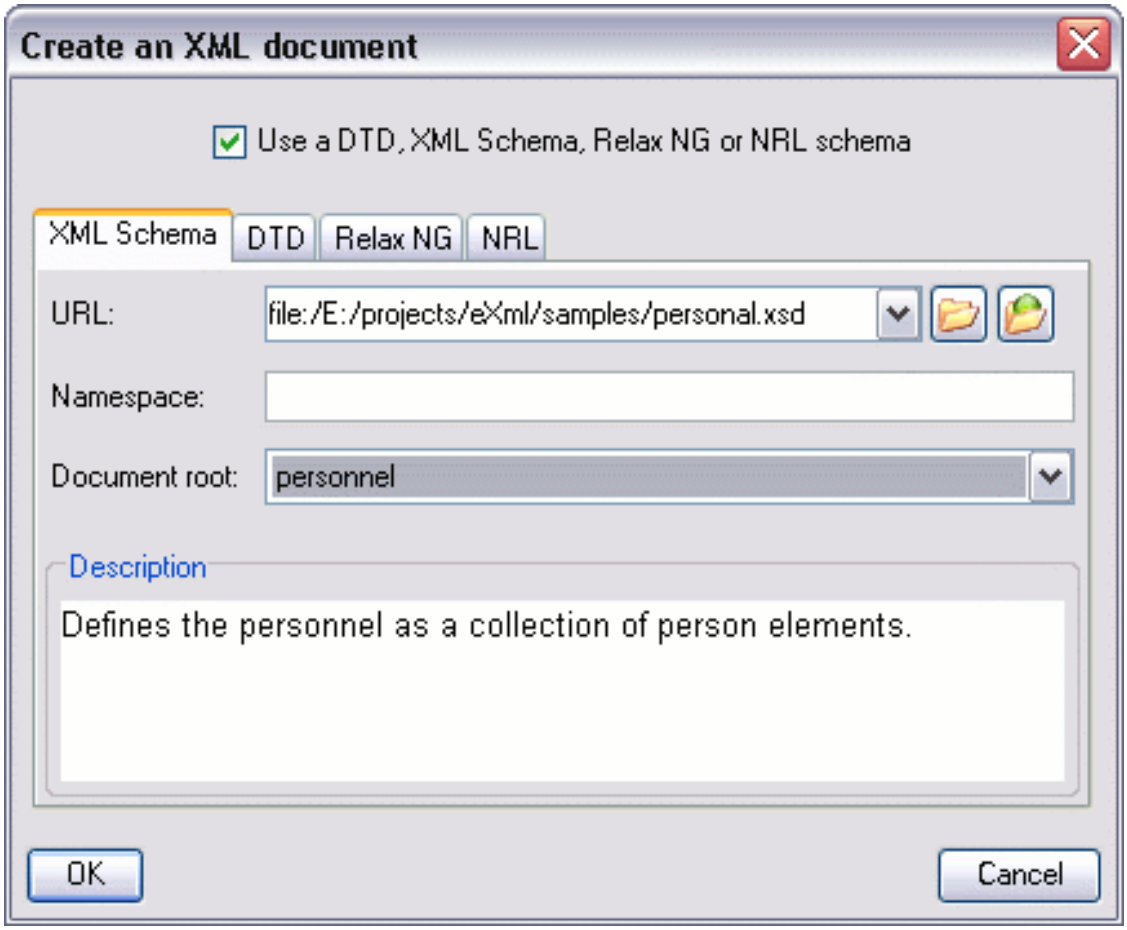
1. Select File-> New (**Ctrl+N**). The New dialog is displayed which contains the supported Document Types: XML, XSL, XML Schema, Document Type Definition, Relax NG Schema, XQuery, Web Services Definition Language, Text File, Java File, JavaScript File, C File, C++ File, Batch File, Shell File, Properties File, SQL File, PHP File, CSS File and PERL File.



2. Select a document type, then click OK. If XML was selected the "Create an XML Document" dialog is displayed otherwise a new document is opened in the Editor Panel.

The Create an XML Document dialog enables definition of a XML Document Prolog using the system identifier of a XML Schema, DTD, Relax NG (full or compact syntax) schema or NRL (Namespace Routing Language) schema. As not all XML documents are required to have a Prolog, you may choose to skip this step by clicking OK. If the prolog is required complete the fields as the following.

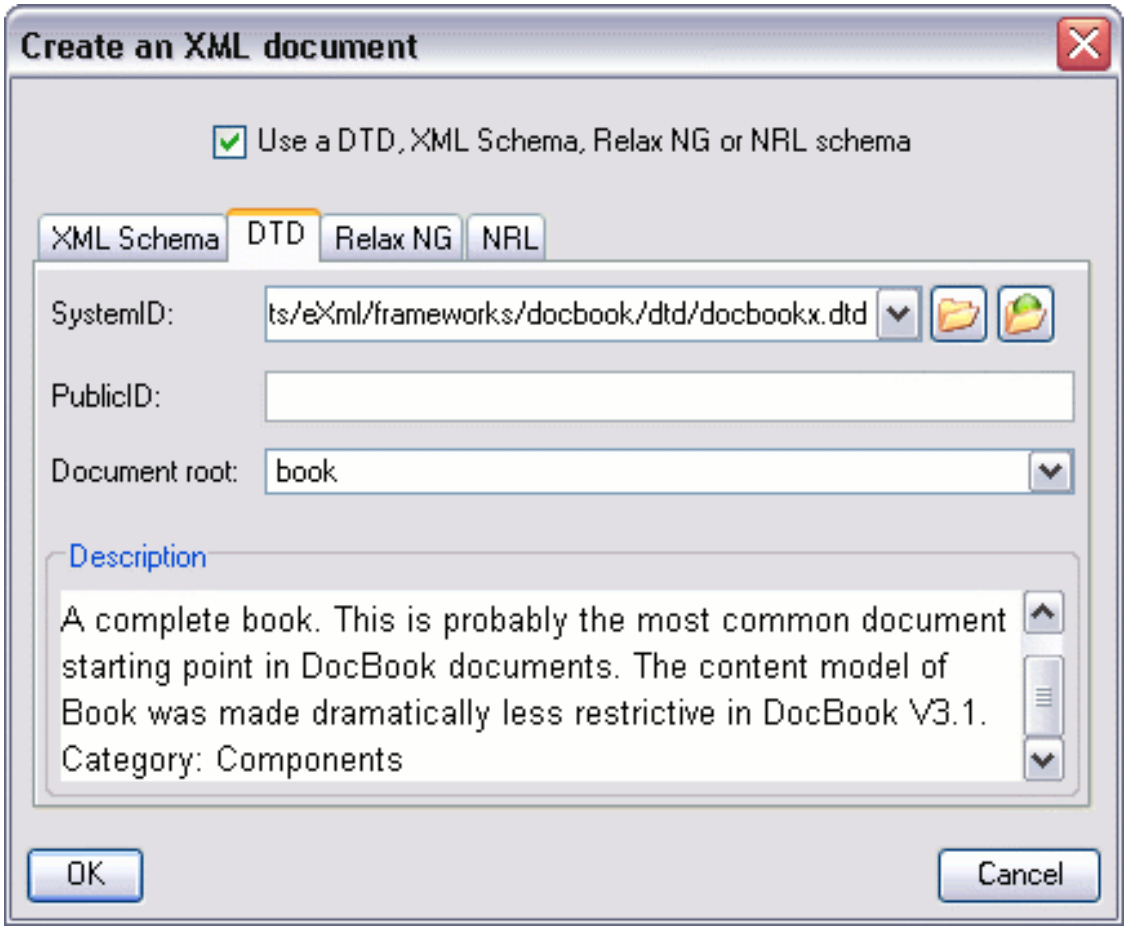
Figure 4.1. The Create an XML Document Dialog - XML Schema Tab



Complete the dialog as follows:

Use a DTD, XML Schema, Relax NG or NRL schema	When checked enables selection between DTD, XML Schema, Relax NG schema or NRL schema.
URL	Specifies the location of an XML Schema Document (XSD).
Namespace	Specifies the document namespace.
Document Root	Populated from the elements defined in the specified XSD, enables selection of the element to be used as document root.
Description	Shows a small definition for the currently selected element.

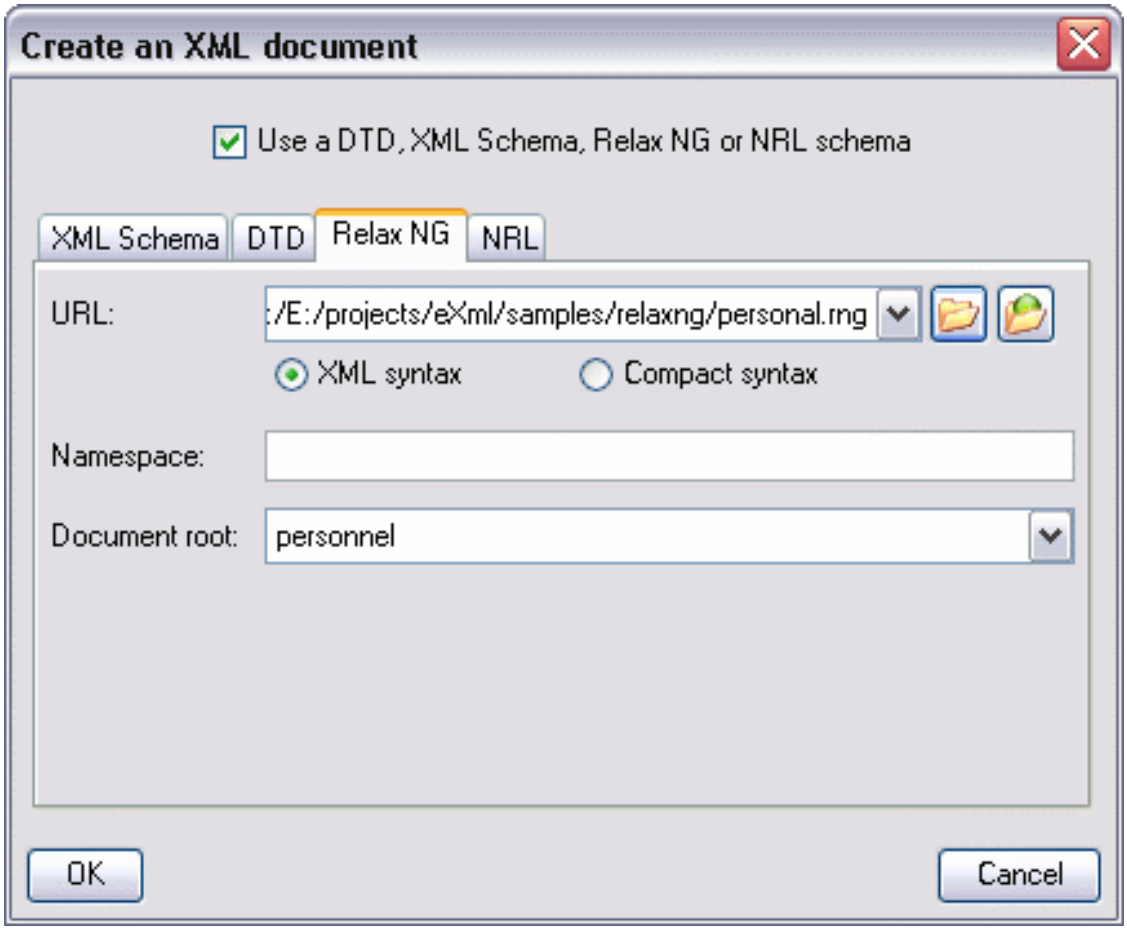
Figure 4.2. The Create an XML Document Dialog - DTD Tab



Complete the dialog as follows:

Use a DTD, XML Schema, Relax NG or NRL schema	When checked enables selection between DTD, XML Schema, Relax NG schema or NRL schema.
System ID	Specifies the location of a Document Type Definition (DTD).
Public ID	Specifies the PUBLIC identifier declared in the Prolog.
Document Root	Populated from the elements defined in the specified DTD, enables selection of the element to be used as document root.
Description	Shows a small definition for the currently selected element.

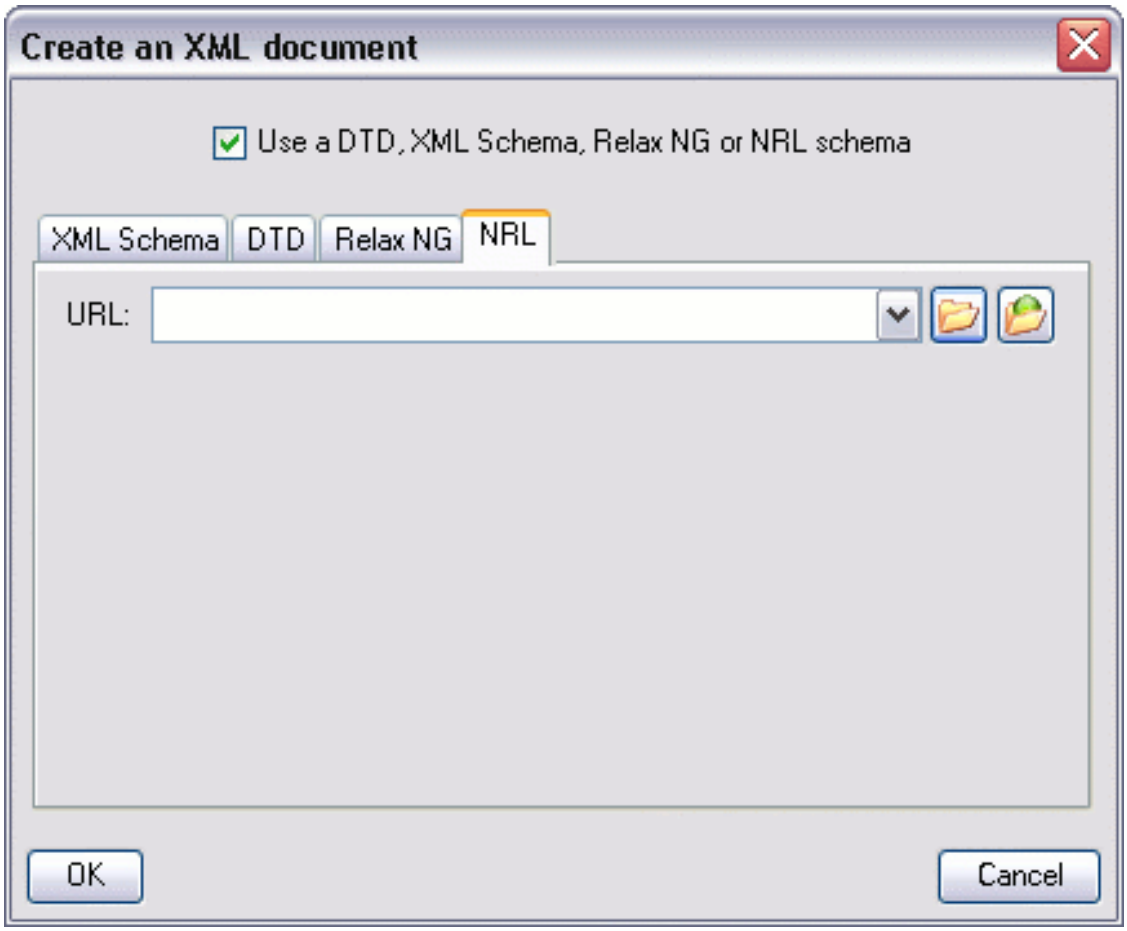
Figure 4.3. The Create an XML Document Dialog - Relax NG Tab



Complete the dialog as follows:

Use a DTD, XML Schema, Relax NG or NRL schema	When checked enables selection between DTD, XML Schema, Relax NG schema or NRL schema.
URL	Specifies the location of a Relax NG schema in XML or compact syntax (RNG/RNC).
XML syntax	When checked the specified URL refers to a Relax NG schema in XML syntax. It will be checked automatically if the user selects a document with the *.rng* extension.
Compact syntax	When checked the specified URL refers to a Relax NG schema in compact syntax. It will be checked automatically if the user selects a document with the *.rnc* extension.
Namespace	Specifies the root element namespace.
Document Root	Populated from the elements defined in the specified RNG or RNC document, enables selection of the element to be used as document root.

Figure 4.4. The Create an XML Document Dialog - NRL Tab



Complete the dialog as follows:

Use a DTD, XML Schema, Relax NG or NRL schema	When checked enables selection between DTD, XML Schema, Relax NG schema or NRL schema.
URL	Specifies the location of a NRL schema (NRL).

Opening, Saving and Closing Documents

As with most editing applications, <oXygen/> lets you open existing documents, save your changes and close them as required.

Opening Documents

Documents can be opened using:

- File-> Open (**Ctrl+O**) : Displays the Open dialog used to discover, select and open one or more

files.

- File->Reopen: Displays a list of recently opened document files. Select a file to open.
- Open: Opens the selected file from the Project View.

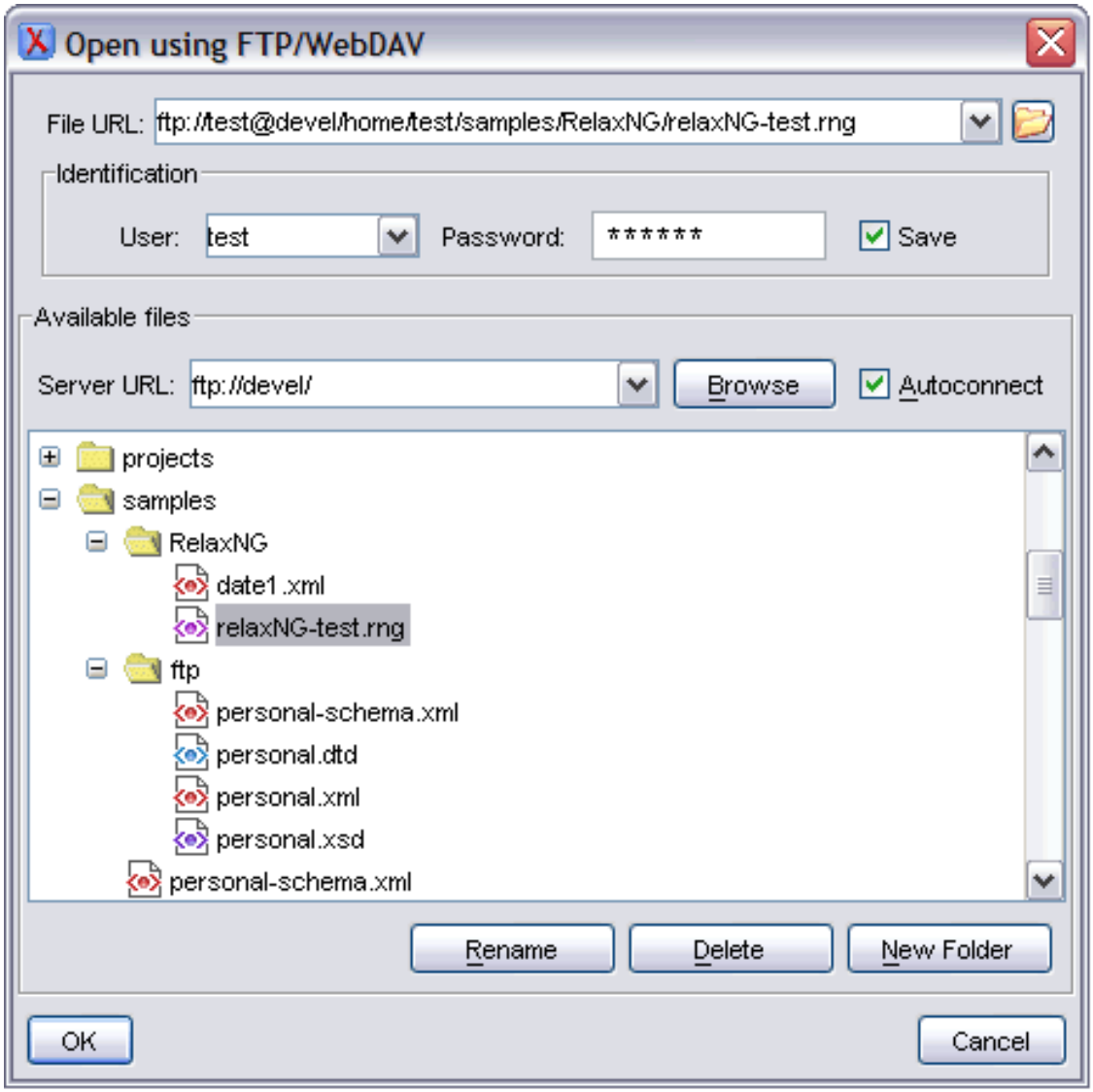
In addition <oXygen/> supports direct opening of files from the command prompt. Use the following command syntax: **sh ./oxygen.sh FileToOpen.xml**

When the Tree View is started the <oXygen/> will display the current document from the Main Window.

Opening and Saving Remote Documents via FTP/ WebDAV

Oxygen supports editing remote files, using the FTP and WebDAV protocols. The remote opened files can be edited exactly as the local ones. They can be added to the project, and can be subject to XSL and FO transformations.

Figure 4.5. Open URL dialog



Note

The WebDAV access is implemented using the Slide package of the Apache Software Foundation. The FTP part is using passive access to the FTP servers. Make sure the server you are trying to connect to is supporting passive connections.

The FTP/WebDAV capabilities have been extensively tested with various servers running on Windows (IIS), Mac OS X and Linux (Apache).

To open the remote files, choose from the main menu "File/Open URL". The displayed dialog is composed of three parts.

Note

If you have set a proxy server to be used by Oxygen, make sure it supports the WebDAV protocol. If it does not, make sure you uncheck the "Use proxy server" from the Options/Preferences/Proxy Configuration pane, otherwise you will not be able to connect to a WebDAV server.

- The first one is an editable combo box, in which it can be specified directly the URL to be opened or saved.

Example 4.1. URLs that can be directly opened

You can type in here an URL like http://some.site/test.xml, in case the file is accessible through normal HTTP protocol, or ftp://anonymous@some.site/home/test.xml if the file is accessible through anonymous FTP.

This combo box is also displaying the current selection when the user changes selection in the browsing tree.

- The second part is controlling the access credentials. If you want to browse for a file on a server, you have to specify the user and password. This information is bound to the selected URL displayed in the "File URL" combo box, and used further in opening/saving the file. If the check box "Save" is selected, then the user and password are saved between editing sessions. The password is kept encrypted into the options file.

Note

Your password is well protected. In the case the options file is moved to another installation, on other machine, the password will become unreadable, since the encryption is user and machine dependent. This is also true if you add URLs having user and password to your project.

- The third part contains the server combo and the "Autoconnect" check box. Into the server combo it may be specified the protocol (HTTP, HTTPS or FTP), the name or IP of the server and, in case of WebDAV, the path to the WebDAV directory.

Example 4.2. Server URLs

When accessing a FTP server, you need to specify only the protocol and the host, like: ftp://server.com, ftp://ftp.apache.org, or if using a nonstandard port: ftp://server.com:7800/ etc.

When accessing a WebDAV server, along with the protocol and the host, it must be specified also the directory of the WebDAV repository.

Important

Make sure that the repository directory ends in a slash "/".
Ex: https://www.some-webdav-server.com:443/webdav-repository/, http://devel:9090/webdav/

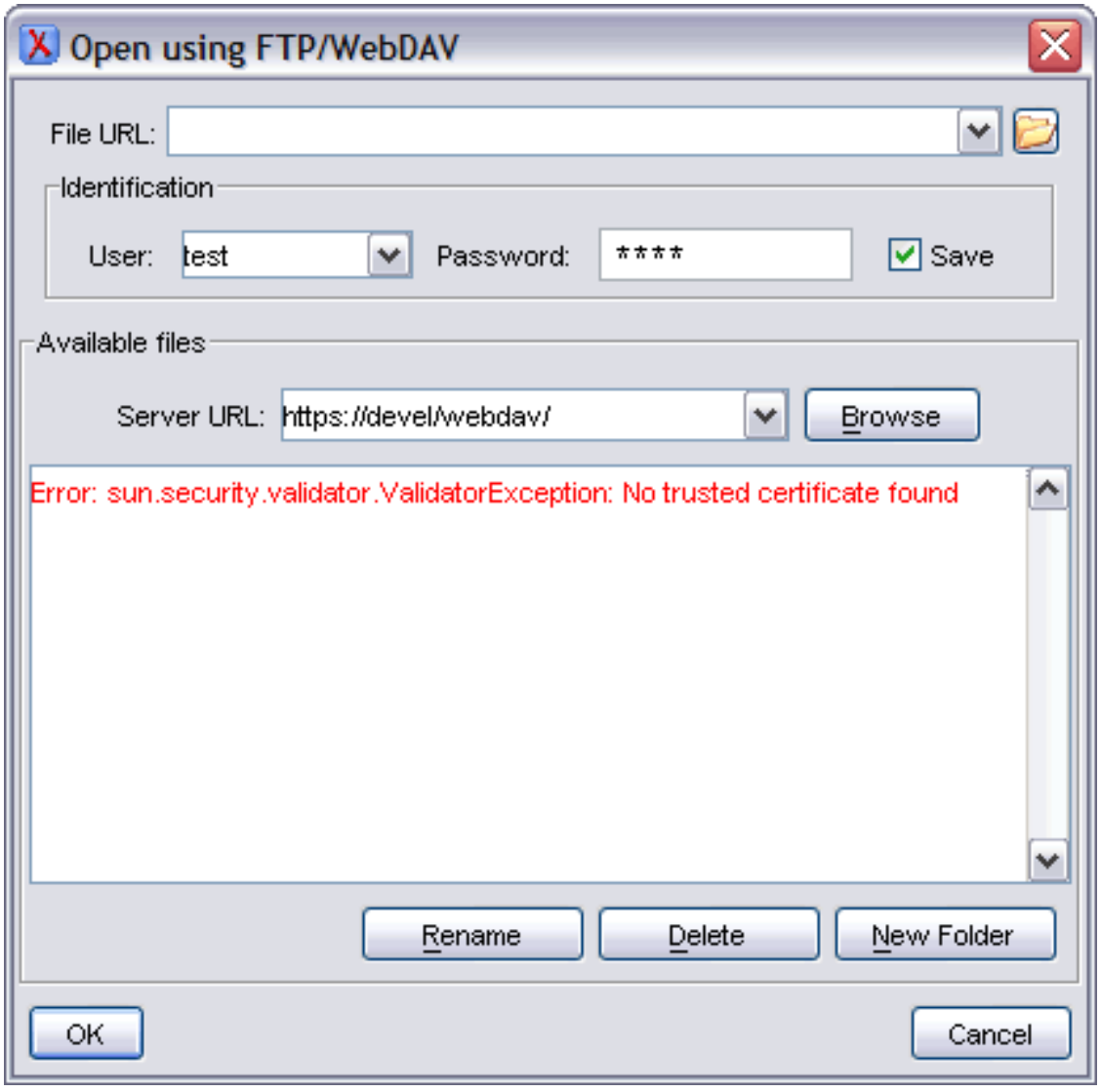
By pressing the "Browse" button the directory listing will be shown in the component bellow. When "Autoconnect" is selected then at every time the dialog is shown, the browse action will be performed.

- The last part consists of a tree view of the documents stored on the server. You can browse the directories, and make multiple selections. Additionally, you may use the "Rename", "Delete", and "New Folder" to manage the file repository.

If you want to access a WebDAV repository across an insecure network <oXygen/> allows you to load and save the documents over the HTTPS protocol (if the server understands this protocol) so that any data exchange with the WebDAV server is encrypted.

When a WebDAV repository is first accessed over HTTPS the server hosting the repository will present a security certificate to <oXygen/> as part of the HTTPS protocol, without any user intervention. <oXygen/> will use this certificate to decrypt any data stream received from the server. For the authentication to succeed you should make sure the security certificate of the server hosting the repository can be read by <oXygen/>. This means that <oXygen/> can find the certificate in the key store of the Java Runtime Environment in which it runs. You know the server certificate is not in the JRE's key store if you get the error "No trusted certificate found" when trying to access the WebDAV repository:

Figure 4.6. The server certificate is not available



You can add a certificate to the key store by exporting it to a local file using any HTTPS-capable Web browser (for example Internet Explorer) and then importing this file into the JRE using the keytool executable bundled with the JRE. The steps are the following using Internet Explorer (if you use other browser the procedure is similar):

Procedure 4.2. Import a HTTPS server certificate

1. Export the certificate into a local file

   a. Point your HTTPS-aware Web browser to the repository URL. If this is your first visit to the repository it will be displayed a security alert stating that the security certificate presented by the server is not trusted.

Figure 4.7. Security alert - untrusted certificate

   b. Press the button "View Certificate".

   c. Select the "Details" tab.

   d. Press the button "Copy to file ...". This will start the Certificate Export Wizard on Windows

   e. Follow the indications of the wizard to save the certificate to a local file, for example *server.cer*.

2. Import the local file into the JRE running <oXygen/>

   a. Open a text-mode console.

   b. Go to the lib/security subdirectory of your JRE directory, that is of the directory where it is installed the JRE running <oXygen/>, for example on Windows *C:\Program Files\Java\j2re1.4.2\lib\security*

c. Run the following command:**..\..\bin\keytool.exe -import -trustcacerts -file local-file.cer -keystore cacerts** where *local-file.cer* is the file containing the server certificate, created during the previous step. Keytool requires a password before adding the certificate to the JRE keystore. The default password is "changeit". If somebody changed the default password then he is the only one who can perform the import. As a workaround you can delete the *cacerts* file, re-type the command and enter as password any combination of at least 6 characters. This will set the password for future operations with the key store.

3. Restart <oXygen/>

Saving Documents

There are three save methods:

- File-> Save (**Ctrl+S**) : Saves the current document. If the document does not have a file, displays the "Save As" dialog.
- File->Save All: Saves all open documents. If any document does not have a file, displays the "Save As" dialog.
- File->Save As: Displays the Save As dialog, used to name and save an open document to a file; or save an exiting file with a new name.

Closing Documents

To close documents use the following methods:

- File-> Close (**Ctrl+W**) : Closes only the selected tab. All other tab instances remain.
- File->Close All: Closes all open documents. If a document is modified or has no file, a prompt to save, not to save, or cancel the save operation is displayed.
- Close: Displayed when a tab is right-clicked. Closes the selected tab when selected.
- Close other files: Displayed when a tab is right-clicked. Closes the other files except the selected tab.
- Close all Tabs: Closes all open tabs within the panel.

XInclude

XInclude is a standard for building up XML documents from smaller pieces, being a replacement for the external entities. The external entities can be a problem because they cannot have a DOCTYPE declaration, and threfore are not valid documents on their own.

The main usage is in the documentation frameworks where the documents may be composed of many files, XInclude allowing the validation of the master file and also the validation of the individual sections.

The advantages of modular documentation are: reusable content units, smaller file units that are edited,

better version control, distributed authoring.

Here is an example of a chapter file and a book file that includes the chapter using XInclude.

Chapter file introduction.xml:

```
<?xml version="1.0"?>
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.3//EN"
                  "http://www.oasis-open.org/docbook/xml/4.3/docbookx.dtd">
<chapter id="introduction">
<title>Getting Started</title>
<section id="Installing <oXygen/>">
...
</chapter>
```

Main book file:

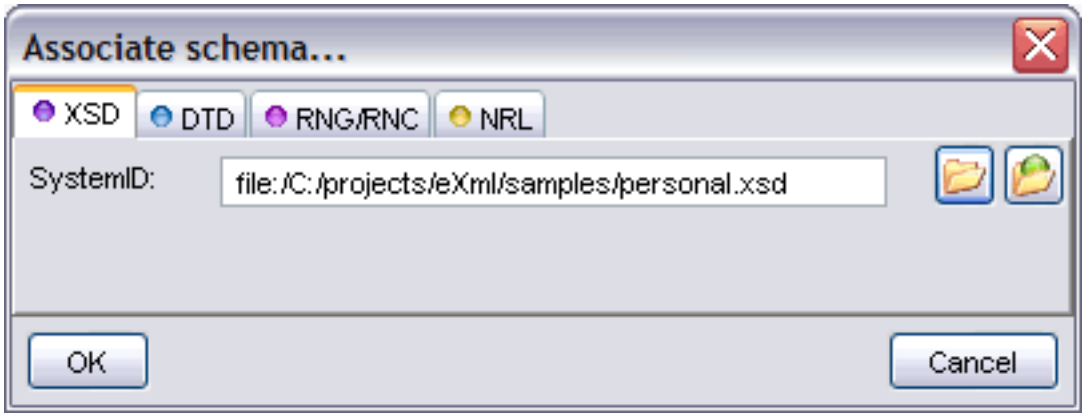
```
<?xml version="1.0"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.3//EN"
    "http://www.oasis-open.org/docbook/xml/4.3/docbookx.dtd">
<book>
<title>User Guide</title>
<para>This guide shows you how to use the editor.</para>
<xi:include
   xmlns:xi="http://www.w3.org/2001/XInclude"  href="introduction.xml" />
   ...
</book>
```

The XInclude support in <oXygen/> is turned off by default. You can turn it on by using the Options/ Preferences/XML/XML Parser Options, the entry Enable XInclude processing.

Associate schema

This is a dialog helping the user to easily associate a schema file with the edited document . Enables definition of a XML Document Prolog using the system identifier of a XML Schema, DTD, Relax NG (full or compact syntax) schema or NRL (Namespace Routing Language) schema.

Figure 4.8. Associate schema dialog



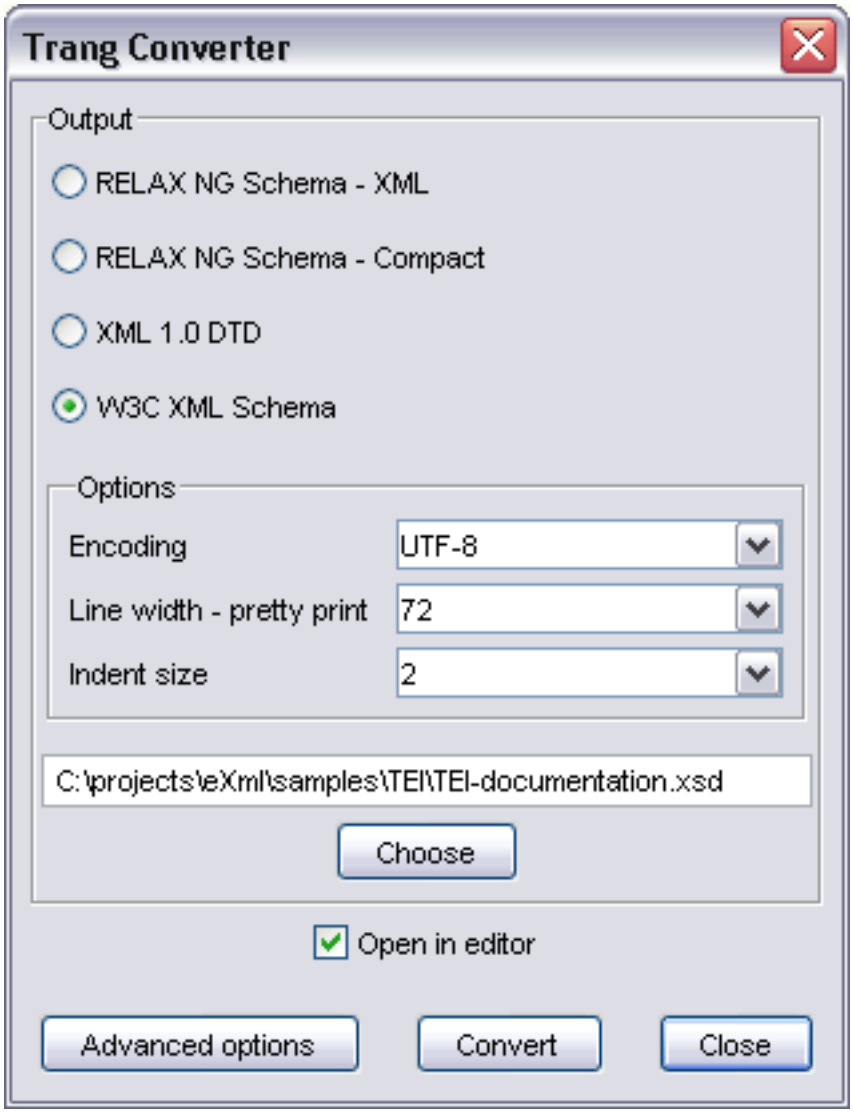
When associating a XML Schema to the edited document if the root element of the document defines a default namespace URI with a "xmlns" attribute the "Associate schema" action adds a xsi:schemaLocation attribute. Otherwise it adds a xsi:noNamespaceSchemaLocation attribute.

Converting Between Grammar Languages

The Trang converter allows you to convert a DTD or Relax NG (full or compact syntax) grammar or a set of XML files to an equivalent XML Schema, DTD or Relax NG (full or compact syntax) grammar. Where perfect equivalence is not possible due to limitations of the target language <oXygen/> will generate an approximation of the source grammar. The conversion functionality is available from Tools -> Trang Converter... .

The Trang converter is integrated in <oXygen/> : a grammar being edited can be converted with a toolbar button if that grammar can be the subject of a supported conversion. For example if you press the "Convert to ..." button while editing a DTD document the following dialog will show up:

Figure 4.9. Convert an edited grammar



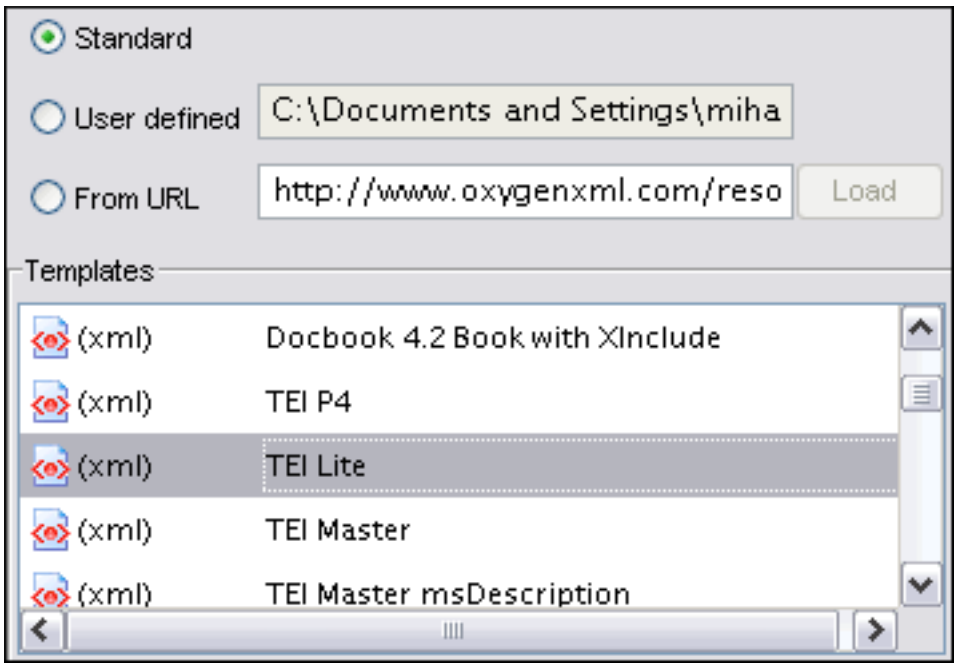
Here you can set the target language of the conversion and the target grammar name.

Creating Documents based on Templates

Templates are documents containing a predefined structure. They provide starting points on which one can rapidly build new documents that repeat the same basic characteristics. <oXygen/> installs a rich set of templates for a number of XML applications. You may also create your own templates and share them with other users.

The Templates dialog, enables you to select templates that have already been created in previous sessions or by other users. Open the Templates dialog by selecting File->New from Templates.

Figure 4.10. The Templates Dialog



Open a template using the following options:

Standard	Populates the Templates list to show templates supplied with the <oXygen/> installation package.
From File	Populates the Templates list to show previous saved personal templates.
From URL	Enables definition of a URL location containing Templates.
Templates List	Displays the available templates for Standard, From File and From URL options.

Procedure 4.3. Creating Documents based on Standard Templates

1. Select File->New from Templates. The Templates dialog is displayed.
2. Select the Standard option from the Load Templates Group. The Templates list displays standard <oXygen/> templates.
3. Scroll the Templates list and select the required Template Type.
4. Click OK. A new document is opened that already contains structure and content provided in the template starting point.

Procedure 4.4. Creating Documents based on Personal Template Files

1. Select File->New from Templates. The Templates dialog is displayed.
2. Select the From File option from the Load Templates Group. The Templates list displays person templates.
3. Scroll the Templates list and select the required Template Type.
4. Click OK. A new document is opened that already contains structure and content provided in the template starting point.

Procedure 4.5. Creating Documents based on URL Template Files

1. Select File->New from Templates. The Templates dialog is displayed.
2. Select the From URL option from the Load Templates Group. The From URL field is enabled.
3. Enter the URL location of the templates, then click Load. The list of templates is retrieved from the URL and displayed in the Templates list.
4. Scroll the Templates list and select the required Template Type.
5. Click OK. A new document is opened that already contains structure and content provided in the template starting point.

Creating New Templates

<oXygen/> enables user defined templates to be created. Templates are created by adding an existing document to the Template library.

Procedure 4.6. Creating New Templates

1. Open the document that will be used to create the Template.

2. Modify the structure and content as required.
3. Select File -> Add to Templates. The Add to Templates dialog is displayed.
4. Enter the name by which the template will be known. Click OK the document is added to the list of Personal Templates.
5. Test the template using the From File option.

Sharing Templates

<oXygen/> stores Personal Templates in an XML file called `.com.oxygenxml.templates.xml`, located in the Home folder of the <oXygen/> user. By copying this file to a Web Server folder and making it accessible via HTTP, other <oXygen/> users can use the From URL option to access the templates.

Procedure 4.7. Sharing Templates

1. Create one or more Personal Templates.
2. Copy `.com.oxygenxml\templates.xml` into an accesible directory on your web server.
3. Test the template using the From URL option.

Editing Documents

While editing a document is a simple procedure, there are some points of which you should be aware and which should make your editing more productive.

Working with Unicode

Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language. Unicode is an internationally recognized standard, adopted by industry leaders. The Unicode is required by modern standards such as XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, etc., and is the official way to implement ISO/IEC 10646.

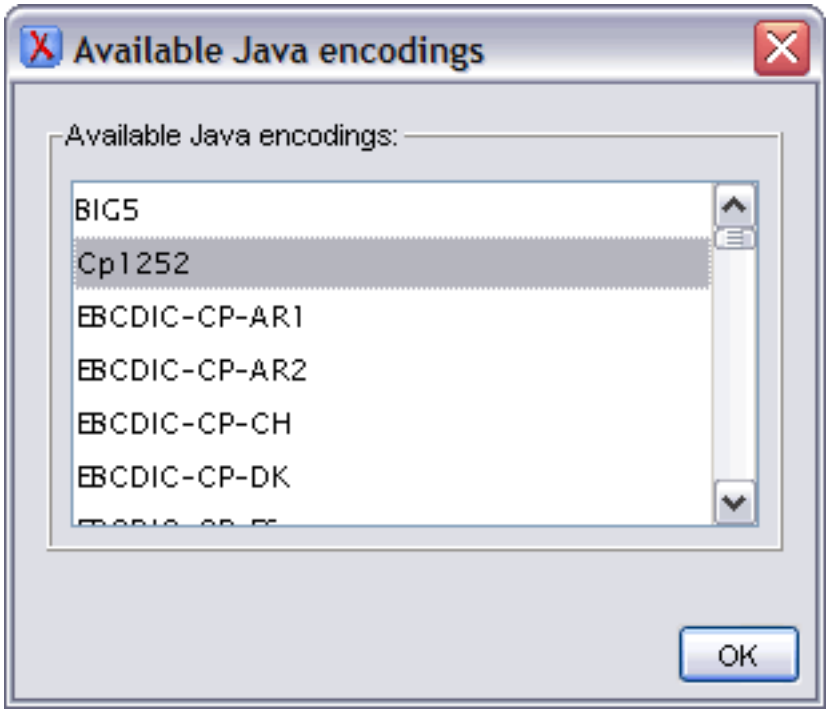
It is supported in many operating systems, all modern browsers, and many other products. The emergence of the Unicode Standard, and the availability of tools supporting it, are among the most significant recent global software technology trends. Incorporating Unicode into client-server or multi-tiered applications and websites offers significant cost savings over the use of legacy character sets.

As a modern XML Editor, <oXygen/> provides support for the Unicode standard. Enabling your XML application to be targeted across multiple platforms, languages and countries without re-engineering. Internally, the <oXygen/> Editor uses 16bit characters covering the Unicode Character set.

On loading documents of the type XML, XSL, XSD and DTD, <oXygen/> reads the document prolog to determine the specified encoding type. This is then used to instruct the Java Encoder to load support for and save using the code chart specified. In the event that the encoding type cannot be determined, <oXygen/> will prompt and display the "Available Java Encodings" dialog. The "Available Java Encod-

ings" dialog provides a list of all encodings supported by the Java platform.

Figure 4.11. Available Java Encodings Dialog



While in most cases you will use UTF-8, simply changing the encoding name will cause the file to be saved using the new encoding. The appendix Unicode Character Encoding provides a Matrix that matches common names with Java Names. It also explains what you should type in the XML prolog to cause the document to be saved as the required encoding.

To edit document written in Japanese or Chinese, you will need to change the font to one that supports the specific characters (a Unicode font). For the Windows platform, use of "Arial Unicode MS" or "MS Gothic" is recommended. Do not expect Wordpad or Notepad to handle these encodings. Use Explorer or Word to eventually examine XML documents.

Note

The naming convention used under Java does not always correspond to the common names used by the Unicode standard. For instance, while in XML you will use encoding="UTF-8", in Java the same encoding has the name "UTF8".

Streamline with Tag-Insight

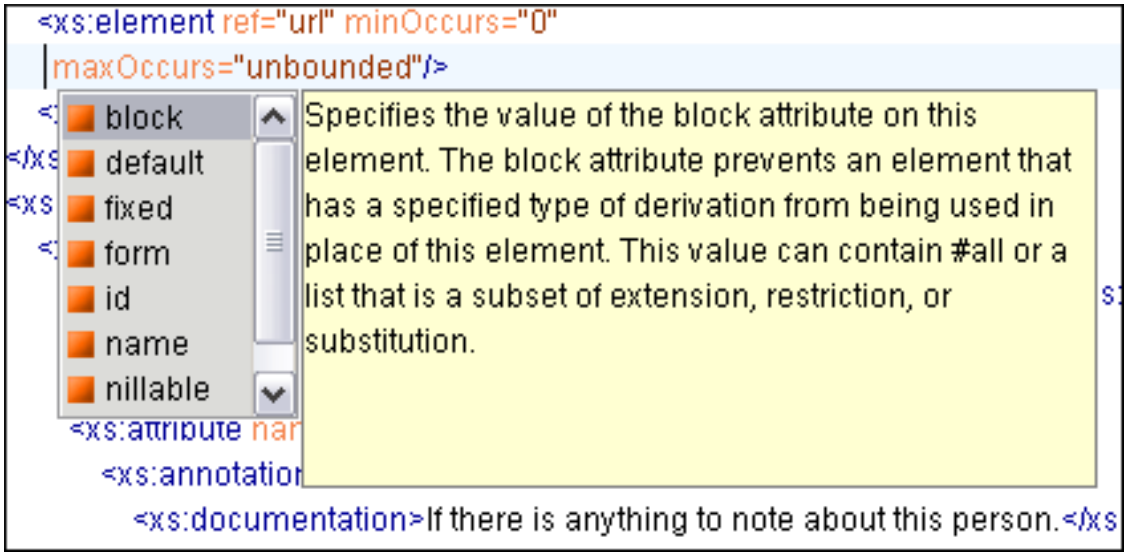
<oXygen/>'s intelligent Tag-Insight feature is an content assistant that enables rapid, in-line identification and insertion of structured language elements, attributes and in some cases their parameter options.

The Tag-Insight assistant is automatically displayed whenever the < character is entered into a document or by pressing **CTRL+Space** on a partial element or attribute name. Moving the focus to highlight an element and pressing the **Enter** key or the **Tab** key, inserts both the start and end parts of the highlighted element in to the document. If the feature Add Element Content of Tag-Insight is enabled all the elements that the new element must contain, as specified in the DTD or XML Schema, are inserted auto-

matically in the document. The Tag-Insight assistant can also add optional content and first choice particle, as specified in the DTD or XML Schema, for the element if the two options are enabled. After inserting, the cursor is positioned directly before the > character of the start tag, if the element has attributes, in order to enable rapid insertion of any attributed supported by the element, or after the > char of the start tag if the element has no attributes. Pressing the space bar, directly after element insertion will again display the assistant. In this instance the attributes supported by that element will be displayed. If an attribute supports a fix set of parameters, the assistant will display the list of valid parameter. If the parameter setting is user defined and therefore variable, the assistant will be closed to enable manual insertion. The values of the attributes can be learned from the same elements in the current document. If the XSD or DTD for the document contains element, attributes or attributes values annotations, these will be presented when the content completion window is displayed, if the coressponding option is enebled.

The content assistant can be invoked at any time by pressing **CTRL+Space** and the context-sensitive list of proposals will be shown in any position of the caret in the edited document in which element, attribute or attribute value insertion makes sense. Such positions are: anywhere within a tag name or at the beginning of a tag name in an XML document, XML Schema, DTD or Relax NG (full or compact syntax) schema, anywhere within an attribute name or at the beginning of an attribute name in any XML document with an associated grammar, and within attribute values or at the beginning of attribute values in XML documents where lists of possible values have been defined for that element in the grammar associated with the document.

Figure 4.12. Tag-Insight Assistant



The content of the Tag-Insight assistant is dependent on the element structure given in a given DTD, XML Schema, Relax NG (full or compact syntax) schema or NRL schema.

The number and type of elements displayed by the assistant is sensitive to the current position of the cursor in the structured document . The child elements displayed within a given element are defined by the structure of the specified DTD, XML Schema, Relax NG (full or compact syntax) schema or NRL schema. All elements that are not child elements of the current element are filtered out. This behavior is continued to element attributes and their parameters. For Relax NG schemas, all the elements are offered in any position of the document and inside a tag all the attributes possible for that tag are displayed by the Tag-Insight assistant (this will be improved in a future version).

The DTD, XML Schema, Relax NG schema or NRL schema used to populate the Tag-Insight assistant is specified in the following methods, in order of precedence:

- From the file specified in the external subset of the document prolog. In this case <oXygen/> reads the prolog and resolves the location of the DTD, XML Schema, Relax NG schema or NRL schema.
- From the file specified in the <oXygen/> Tag-Insight dialog. <oXygen/> will read the Tag-Insight settings when the prolog fails to provide or resolve the location of a DTD, XML Schema or Relax NG schema.

Creating DTDs

When working with documents that do not specify a DTD, or for which the DTD is not known or does not exist, <oXygen/> is able to learn and translate it to a DTD, which in turn can be saved to a file in order to provide a DTD. In addition to being useful for quick creation of a DTD that will be capable of providing an initialization source for the Tag-Insight assistant. This feature can also be used to produce DTDs for documents containing personal or custom element types.

Procedure 4.8. To create a DTD:

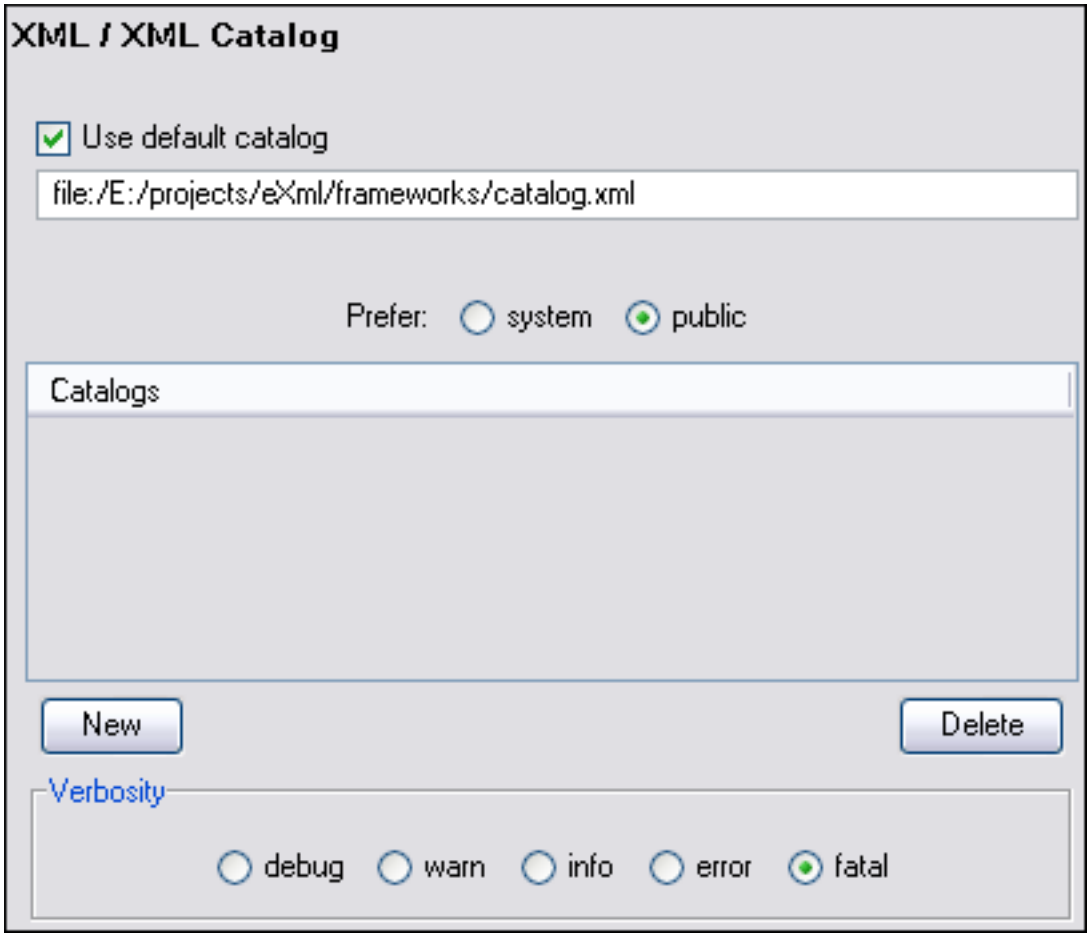
1. Open the structured document from which a DTD will be created.
2. Select Document->Learn Structure (**Ctrl+Shift+L**). <oXygen/> will learn the document structure, when finished displaying words "Learn Complete" in the Message Pane of the Editor Status bar.
3. Select Document->Save Structure (**Ctrl+Shift+S**) to save the DTD currently stored in memory to file.

Note

The resulting DTD is only valid for documents containing the elements and structures defined by the document used as the input for creating the DTD. If new element types or structures are defined in a document, they must be added to the DTD in order for successful validation.

Working with XML Catalogs

When Internet access is not available one or more XML catalogs can be added to the list in the dialog below and the local copies of the DTD, XML Schema, Relax NG schema and/or NRL schema files will be used. When you add or delete an XML catalog to the list of XML catalogs in the Options -> Preferences -> XML Catalog pane you must restart the application so that the changes take effect.



If "Use default catalog" option is checked <oXygen/> will use the built-in catalogs for DocBook, TEI and XHTML documents located in the *frameworks* subdirectory of the installation directory. Otherwise <oXygen/> will use the catalogs specified in the list.

The Prefer option is used to specify whether <oXygen/> will try to resolve first the PUBLIC or SYSTEM reference using the specified XML catalogs. If a PUBLIC reference is not mapped in any of the catalogs then a SYSTEM reference is looked up. The verbosity level specifies the types of output messages displayed and can have one of the values: debug, warn, info, error and fatal.

If the user has added no XML catalogs to this list then <oXygen/> will add by default the built-in catalogs for DocBook and TEI documents located in the *frameworks/docbook* and *frameworks/tei* subdirectories of the installation directory.

Formatting and Indenting Documents (Pretty Print)

In structured markup languages, the whitespace between elements that is created by use of the **Space bar**, **Tab** or multiple line breaks insertion from use of the **Enter**, is not recognized by the parsing tools. Often this means that when structured markup documents are opened, they are arranged as one long, unbroken line, what seems to be a single paragraph.

While this is perfectly acceptable practice, it makes editing difficult and increases the likelihood of errors being introduced. It also makes the identification of exact error positions difficult. Formatting and Indenting, also called "Pretty Print", enables such documents to be neatly arranged, in a manner that is consistent and promotes easier reading on screen and in print output.

Pretty print is in no way associated with the layout or formatting that will be used in the transformed document. This layout and formatting is supplied by the XSL style sheet specified at time of transformation.

Procedure 4.9. To format and indent a document:

1. Open or focus on the document that is to be formatted and indented.
2. Selecting Document->Format and Indent (**Ctrl+Shift+P**). While in progress the Status Panel will indicate "Pretty print in progress". On completion, this will read "Pretty print successful" and the document will be arranged.

Note

Pretty Print can format empty elements as an auto-closing markup tag (ex. <a/>) or as a regular tag (ex. <a>). It can preserve the order or attributes or order them alphabetically. Also the user may specify a list of elements for which white spaces are preserved exactly as before Pretty print and a one with elements for which white space is stripped. These can be configured from Options-> Preferences -> Editor -> Format.

Pretty Print requires that the structured document is "Well Formed". If the document is not "Well Formed" an error message is displayed. The message will usually indicate that a problem has been found in the form and will hint to the problem type. It will not highlight the general position of the error, to do this run "Well Formed" function by selecting Document->Check document form (**Ctrl+Shift+W**).

Using XPath Expressions

XPath is a language for addressing specific parts of an XML document. XPath, like the Document Object Model (DOM), models an XML document as a tree of nodes. An XPath expression is a mechanism for navigating through and selecting nodes from the XML document. An XPath expression is in a way analogous to a Structured Query Language (SQL) query used to select records from a database.

XPath models an XML document as a tree of nodes. There are different types of nodes, including element nodes, attribute nodes and text nodes. XPath defines a way to compute a string-value for each type of node.

XPath defines a library of standard functions for working with strings, numbers and Boolean expressions.

Examples:

child: : * Select all children of the root node.

.//name Select all elements having the name "name", descendants of the current node.

/catalog/cd[price>10.80]Selects all the cd elements that have a price element with a value larger than 10.80

To use XPath effectively requires, at least an understanding of the XPath Core Function Library. Once you have this knowledge the <oXygen/> XPath expression field part of the Editor toolbar can be used to aid you in XML document development.

To find out more about XPath, the following URL is recommended: http://www.w3.org/TR/xpath

In <oXygen/> the results of an XPath query are returned in the Message Panel. Clicking a record in the result list highlights the nodes within the editing panel.

Results are returned in a format that itself is a valid XPath expression:

- [FileName.xml] /node[value]/node[value]/node[value] -

Example 4.3. XPath Utilization with DocBook DTD

Our example is taken from a DocBook book based on the DocBook XML DTD. The book contains a number of chapters. DocBook defines that chapters as have a <chapter> start tag and matching </chapter> end tag to close the element. To return all the chapter nodes of the book enter **//chapter** into the XPath expression field, then Enter. This will return all the chapter nodes of the DocBook book, in the Message Panel. If your book has six chapters, their will be six records in the result list. Each record when clicked will locate and highlight the chapter and all sibling nodes contained between the start and end tags of the chapter.

If we used XPath to query for all example nodes contained in the section 2 node of a DocBook XML document we would use the following XPath expression **//chapter/sect1/sect2/example**. If an example node is found in any section 2 node, a result will be returned to the message panel. For each occurrence of the element node a record will be created in the result list.

In our example an XPath query on the file `oxygen.xml` determined that:

```
- [oxygen.xml] /chapter[1]/sect1[3]/sect2[7]/example[1]
```

Which means:

In the file `oxygen.xml`, first chapter, third section level 1, seventh section level 2, the example node found is the first in the section.

Tip

If your project is comprised of a main file with ENTITY references to other files, you can use XPath to return all the name elements of a certain type by querying the main file. The result list will query all referenced files.

Using Check Spelling

The Check Spelling option enables you to perform the check spelling on the current document:

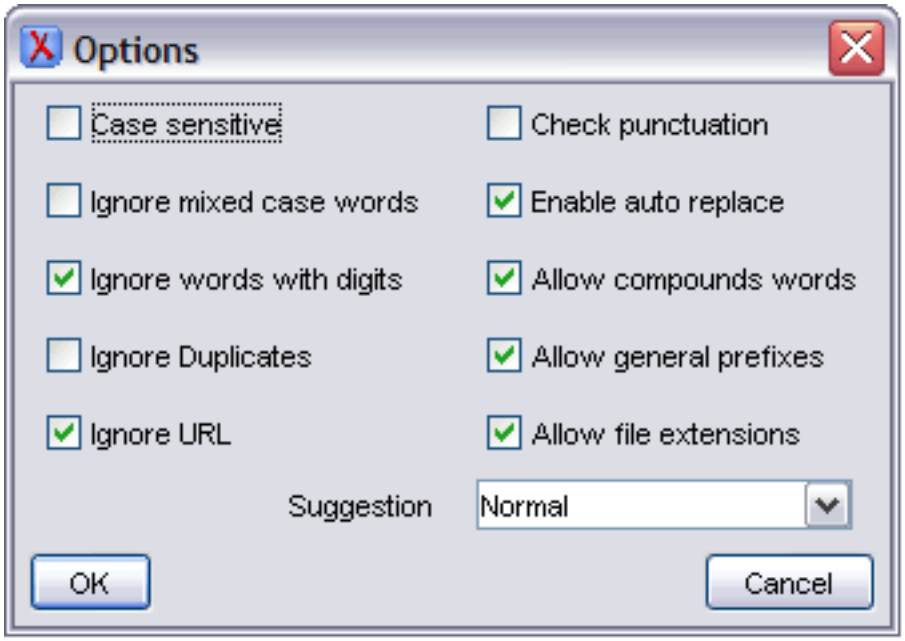
Figure 4.13. Check Spelling Dialog

Complete the dialog as follows:

Unrecognized Word	Contains the word that cannot be found in the selected dictionary. The word is also highlighted in the XML document.
Replace with	The character string which is suggested to replace the unrecognized word.
Guess	Displays a list of words suggested to replace the unknown word. Double clicking a word in this list automatically inserts it in the document and continues the spell checking process.
Dictionary	Displays a list with the available dictionaries.
Replace	Replaces the currently highlighted word in the XML document, with the selected word in the "Replace with" field.
Replace All	Replaces all occurrences of the currently highlighted word in the XML document, with the selected word in the "Replace with" field.

Ignore	Allows you to continue checking the document while ignoring the first occurrence of the unknown word. The same word will be flagged again if it appears in the document.
Ignore all	Ignores all instances of the unknown word in the whole document.
Learn	Includes the unrecognized word in the list of valid words so that the spell checker will no longer consider it for correction.
Options	Sets the configuration options of the Spell Checker.
Begin at caret postion	When checked, the spell checker begins checking from the current cursor position.
OK	Closes the Spell Checker dialog.

Figure 4.14. Options Dialog



The Options dialog contains the global check spelling options:

Case sensitive	When checked, operations ignore capitalization errors.
Ignore mixed case words	When checked, operations do not check words containing case mixing (e.g. "SpellChecker").
Ignore word with digits	When checked, the Spell Checker do not check words containing digits (e.g. "b2b").
Ignore Duplicates	When checked, the Spell Checker do not signal two successive identical words as an error.
Ignore URL	When checked, ignores words looking like URL or file names (e.g.

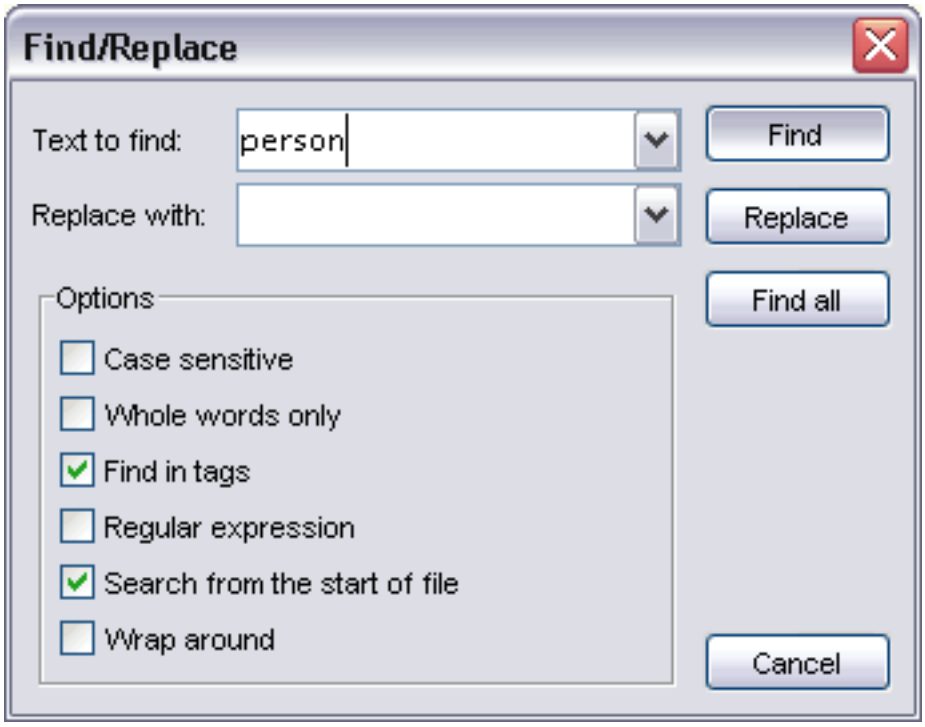
	"www.oxygenxml.com" or "c:\boot.ini") .
Check punctuation	When checked, punctuation checking is enabled: misplaced white space and wrong sequences, like a dot following a comma, are detected.
Enable auto replace	Enables the "Replace Always" feature.
Allow compounds words	When checked, all words formed by concatenating two legal words with an hyphen are accepted. If the language allows it, two words concatenated without hyphen are also accepted.
Allow general prefixes	When checked, a word formed by concatenating a registered prefix and a legal word is accepted. For example if "mini-" is a registered prefix, accepts "mini-computer".
Allow file extensions	When checked, accepts any word ending with registered file extensions (e.g. "myfile.txt", "index.html" etc.).
Suggestion	This option indicates the type of spell checker accuracy, which may be: "Favour speed over quality", "Normal" and "Favour quality over speed".

Using Search and Replace

The Search and Replace option enables you to perform the following operations on the current document:

- find occurrences of a word or string of characters including white spaces and highlight the position in the editor.
- replace occurrences of target defined in the "Find" field with a word or string of characters, including white spaces, defined in the "Replace" field.
- find all occurrences of a word or string of characters including white spaces and return a result list to the Message Panel.
- replace all occurrences of a word or string of characters including white spaces.

Figure 4.15. Find/Replace Dialog



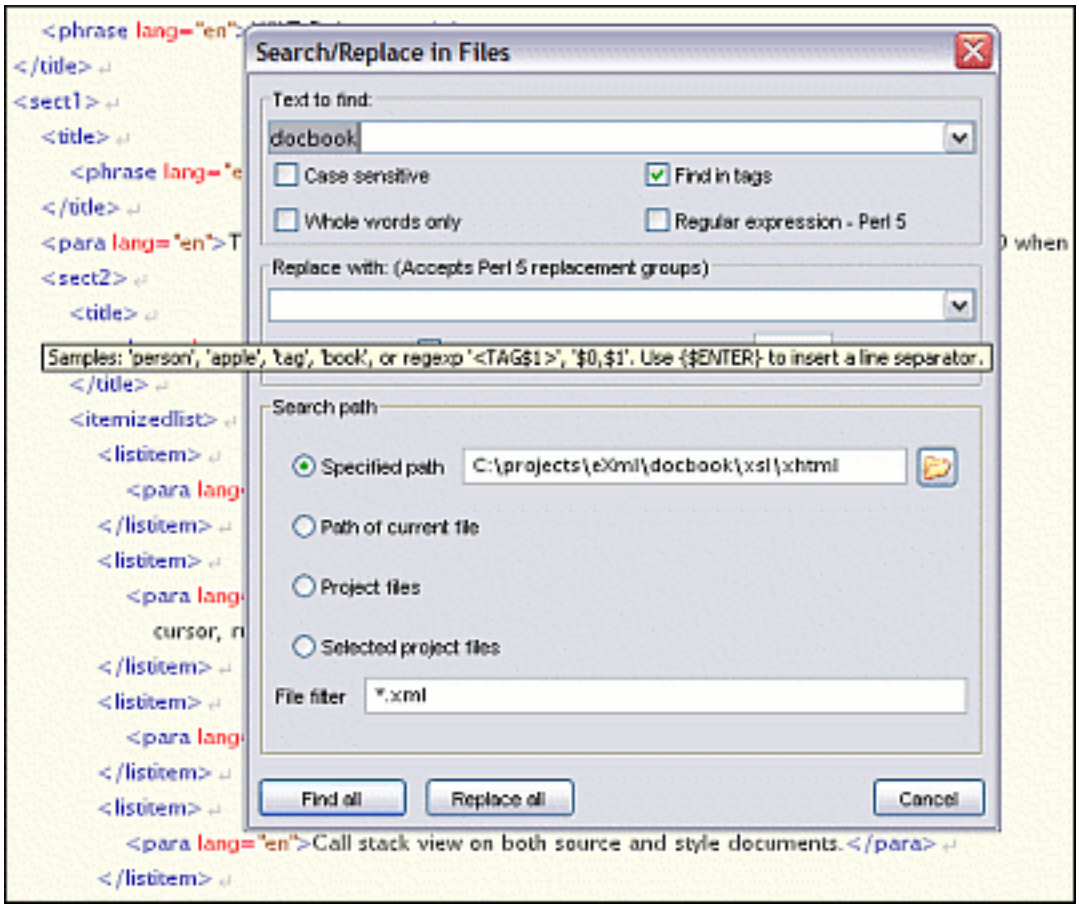
Complete the dialog as follows:

Text to find	The target character string to search for.
Replace with	The character string with which to replace the target. It may contain '{$ENTER}' which at the replace time will insert a new line character.
Find	Execute a find operation for the next occurrence of the target and stop.
Replace	Execute a replace operation for the target and stop.
Find all	Executes a find operation and returns all results to the Message Panel.
Case sensitive	When checked, operations are case sensitive.
Whole words only	When checked only whole occurrences of a word will be included in the operation.
Find in tags	When checked, operation will include content of the start and end tags of the XML elements.
Regular Expression	When checked allows using any regular expression in PERL syntax.
Search from file start	Starts the operation from start of file, position 0:0.
Wrap around	Continues the find from the start of the document after reaching the end.

Using Search and Replace in Files Dialog

The Search and Replace in Files option enables you to perform the same operations as the Search/Replace option on any number of files located in a given path.

Figure 4.16. Search/Replace in Files



Complete the dialog as follows:

Text to Find	The target character string to search for.
Case Sensitive	When checked, operations are case sensitive.
Whole words only	When checked only whole occurrences of a word will be included in the operation.
Find in tags	When checked, operation will include content of the start and end tags of the XML elements.
Regular Expression	When checked allows using any regular expression in PERL syntax.
Replace with	The character string with which to replace the target. It may contain '{$ENTER}' which at the replace time will insert a new line

	character.
Make Backups with extension	In the replace process <oXygen/> makes backup files of the modified files. The default extension is *bak, but you can change extension as you prefer.
Specified Path	Choose the search path
Path of current file	Use the path of the current file
Project Files (File Filter)	Search the files from the current project using the specified file filter.
Selected project files	Search only in the selected files of the current opened project

Note

The search is performed only on local files. If you have added to the project remote files from an FTP or Webdav server these will be skipped from the search.

Find All	Executes a find operation and returns the result list to the Message Pane
Replace All	Replaces all occurrences of the target contained in the specified files.

Use this option with caution.

Global search and replace across all project files does not open the files containing the targets, nor does it prompt on a per occurrence basis, to confirm that a replace operation must be performed. As the operation simply matches the string defined in the find field, this may result in replacement of matching strings that were not originally intended to be replaced.

Working with Large Documents

The problem

Let's consider the case of documenting a large project. It is likely to be several people involved. The resulting document can be few megabytes in size. How to deal with this amount of data in such a way the work parallelism would not be affected ?

Fortunately, XML provides a solution for this. It can be created a master document, with references to other documents, containing the document sections. The users can edit individually the sections, then apply FOP or XSLT over the master and obtain the result files, let say PDF or HTML.

- The master should declare the DTD to be used and the external entities - the sections. A sample document is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE book SYSTEM "../xml/docbookx.dtd" [ <!ENTITY testing SYSTEM "testing.x
 <book>
      <chapter> ...
```

At a certain point in the master document there can be inserted the section "testing.xml" entity:

... &testing; ...

- The document containing the section must not define again the DTD.

  <section> ... here comes the section content ... </section>

Note

The indicated DTD and the element names ("section", "chapter") are used here only for illustrating the inclusion mechanism. You can use any DTD and element names you need.

Using the project panel

When you have a large number of files to edit and to organize, you may use the project panel.

Note

The operations can be accessed using the toolbar buttons.

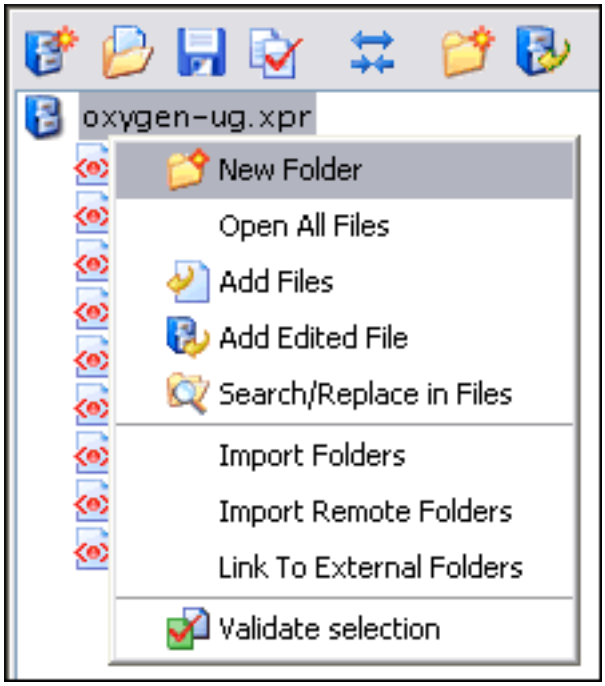
Creating a project

Choose File / New Project to create a new project. Make sure the project panel is visible by checking the View / Show Project item. (A check mark should be displayed in the menu.)

Creating project folders

We can organize the project as a collection of folders. These are logical folders, they do not have any connection with directories on the disk. Right click on the icon of the project, in the project panel. A popup menu will be shown.

Figure 4.17. Project panel popup menu



Choose the first option, "New folder". Enter a name of the folder.

Figure 4.18. Project panel new folder dialog

Adding files to a project

To add one or more files to the newly created folder, right click on it, and choose "Add file".

A shortcut for adding the edited file to the selected folder is to press the right-most button from the project panel toolbar.

Figure 4.19. Project panel toolbar

Removing files or project folders

Right click on the item you want to remove. Choose the remove option.

Setting a schema for the Tag-Insight

In case you are editing document fragments, for instance the chapters from a book each one in a separate file, you can activate the Code Completion for these fragments in two ways:

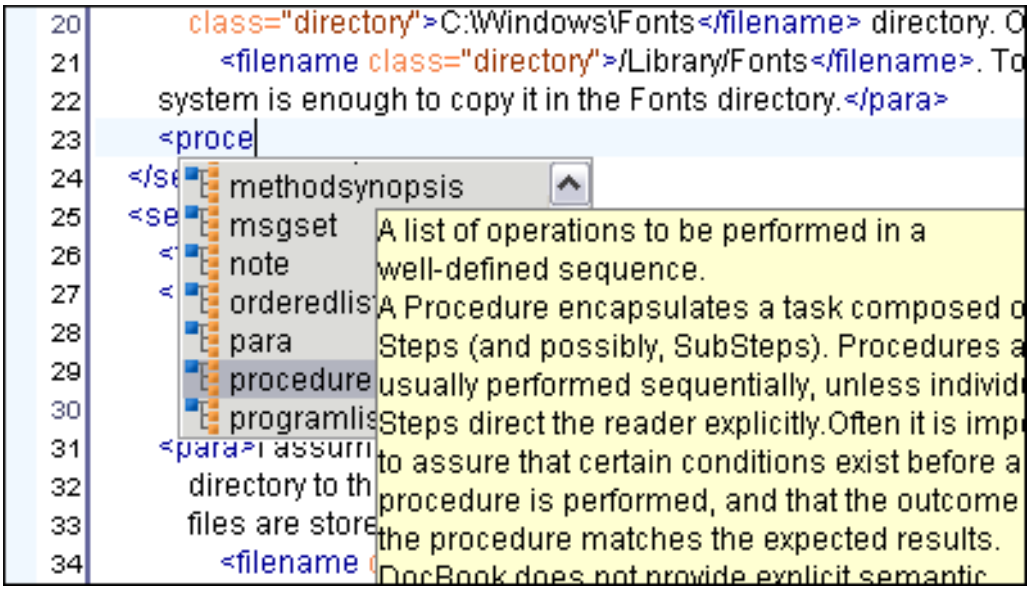
Setting a default DTD

As explained above, when splitting a large document, only one file will contain the Document Type Definition (the DTD) and will include the others. The included sections cannot define again the DTD because the main document will not be valid.

Important

The editor is creating the Tag-Insight lists by analysing the specified DTD and the current context (the position in the editor). If you change the DTD you can observe that the list of tags to be inserted is changing.

Figure 4.20. Tag-Insight driven by a Docbook DTD



To offer Tag-Insight on the included files, you can specify a DTD or XML Schema to be used when the documents do not specify one.

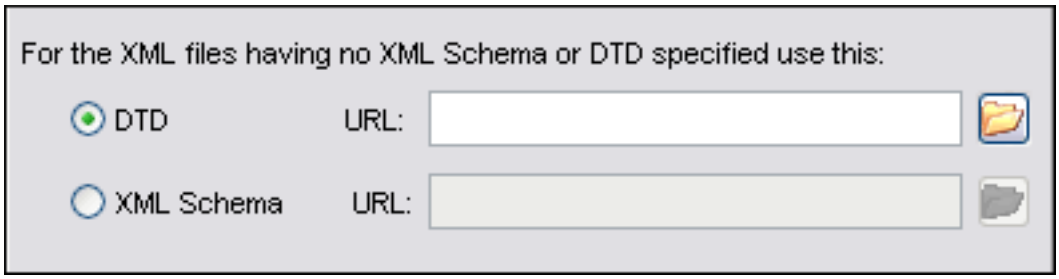
Changing the default DTD

From the Options menu, Preferences dialog, choose Tag-Insight/Default.

The displayed panel has two radio buttons: one for DTD and the other for XML Schema.

If you are creating documentation with Docbook then is a good choice to set the docbookx.dtd file.

Figure 4.21. Tag-Insight configuration dialog



Setting a Processing Instruction

The same effect is obtained by configuring a processing instruction that specifies the DTD to be used. The advantage of this method is that you can configure the TagInsight for each file. The processing instruction must be added at the beginning of the document, just after the XML prologue: <?oxygen DTDSystemID="system" DTDPublicID="public"?>

Note

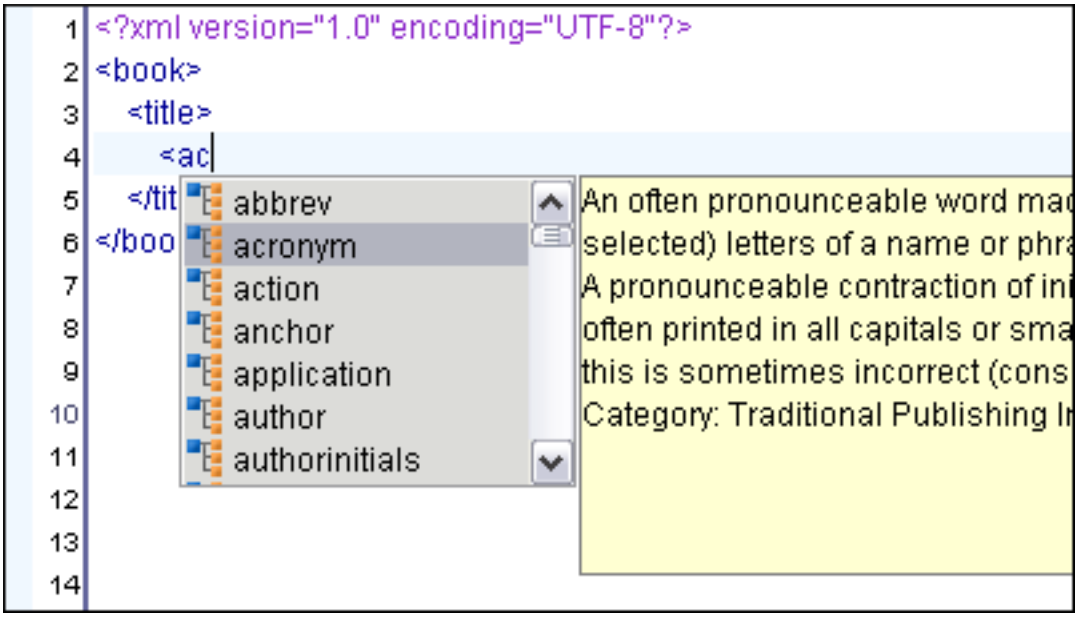
The system and public values must be the same as for a DOCTYPE declaration.

Creating a included file - a section.

Select File / New. Choose the XML type, but with no DTD.

Make sure that in the Tag-Insight option you have choose the correct DTD. Now you can type in the edited document the root element of your section. For example, if you are using docbook it can be "<chapter></chapter>" or "<section></section>". Now if you are moving the cursor between the tags and press "<", you will see the list of inserable element names.

Figure 4.22. Tag-Insight list over a document with no DTD



Note

The validation will not work on a included file, as no DTD is set. The validation can be done only from the master file. At this point you can only check the document to be well-formed.

Debugging Your Documents

The W3C XML specification states that a program should not continue to process an XML document if it finds a validation error. The reason is that XML software should be easy to write, and that all XML documents should be compatible. With HTML it was possible to create documents with lots of errors (like when you forget an end tag). One of the main reasons that HTML browsers are so big and incompatible, is that they have their own ways to figure out what a document should look like when they encounter an HTML error. With XML this should not be possible.

However, when creating an XML document, errors are very easily introduced. When working with large projects or many files, the probability that errors will occur is even greater. Determining that your project is error free can be time consuming and even frustrating. For this reason <oXygen/> provides functions that enable easy error identification and rapid error location.

Checking XML Form

XML with correct syntax is "Well Formed XML" and has correct XML syntax.

A "Well Formed XML" document is a document that conforms to the XML syntax rules.

- All XML elements must have a closing tag.
- XML tags are case sensitive.
- All XML elements must be properly nested.
- All XML documents must have a root element.
- Attribute values must always be quoted.

- With XML, white space is preserved.

Using the Check XML Form function checks your project for any deviation from these rules. If any error is found the result is returned to the Message Panel. Each error is one record in the Result List and is accompanied by an error message. Clicking the record will open the document containing the error and highlight the approximate location.

Example 4.4. Check XML Form Error Message

In our example we will use the case where an end tag is missing from a DocBook listitem element. In this case running Check XML Form will return the following error.

```
F The element type
            "listitem" must be terminated by the matching end-tag "</listitem>".
```

To resolve the error, click in the result list record which will locate and highlight the errors approximate position. Review the "listitems", identify which is missing an end tag and insert </listitem>.

Validating Documents

A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a Document Type Definition (DTD) or XML Schema, which defines the legal elements of an XML document.

The purpose of a DTD is to define the legal building blocks of an XML document. It defines the document structure with a list of legal elements.

The <oXygen/> Validate XML function ensures that your document is compliant with the rules defined by an associated DTD, XML Schema or Relax NG schema.

Example 4.5. Validate XML Error Message

In our example we will use the case where a DocBook listitem element does not match the rules of the `docbookx.dtd`. In this case running Validate XML will return the following error.

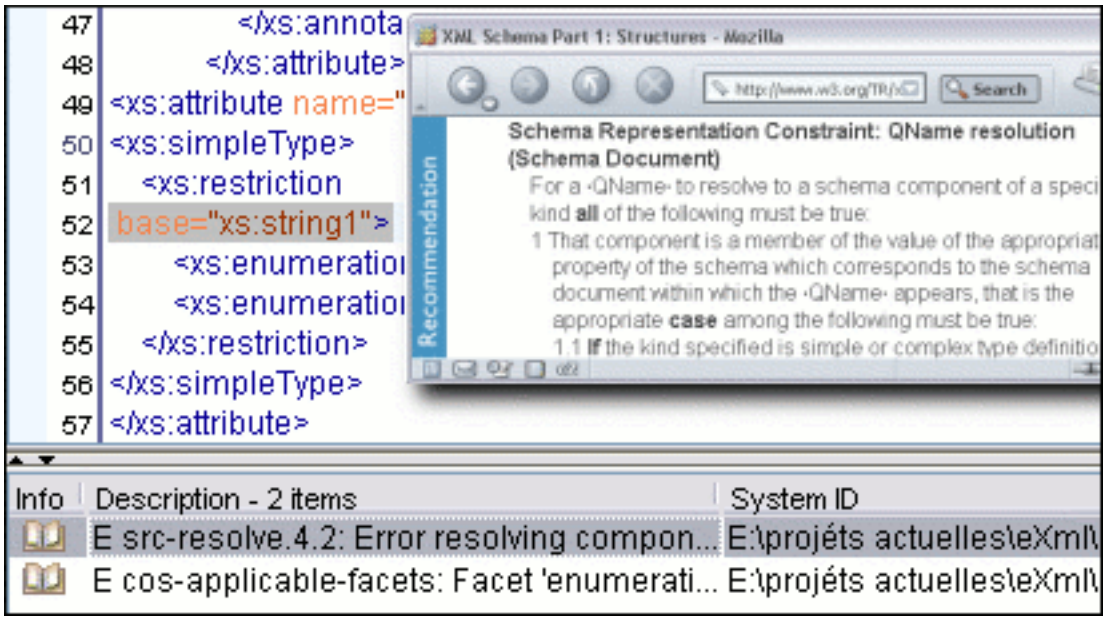
```
E The content of element type "listitem"
            must match"(calloutlist|glosslist|itemizedlist|orderedlist|segmentedli
            simplelist|variablelist| caution|important|note|tip|warning|
            literallayout|programlisting|programlistingco|screen|
            screenco|screenshot|synopsis|cmdsynopsis| funcsynopsis|classsynopsis|f
            constructorsynopsis| destructorsynopsis|methodsynopsis|formalpara|para
            address|blockquote|graphic|graphicco|mediaobject| mediaobjectco|inform
            informalexample| informalfigure|informaltable|equation|example|
            figure|table|msgset|procedure|sidebar|qandaset|anchor|
            bridgehead|remark|highlights|abstract|authorblurb|epigraph| indexterm|
```

As you can see, this error message is a little more difficult to understand, so understanding of the syntax or processing rules for the DocBook XML DTD's "listitem" element is required. However, the error message does give us a clue as to the source of the problem, but indicating that "The content of element type "listitem" must match".

Luckily most standards based DTD's, XML Schema's and Relax NG schemas are supplied with reference documentation. This enables us to lookup the element and read about it. In this case we would want to learn about the child elements of "listitem" and their nesting rules. Once we have correctly inserted the required child element and nested it in accordance with the XML rules, the document will become valid on the next validation test.

At the XML Schema validation <oXygen/> indicates the specification reference for the XML Schema errors. The error messages contain an Info field that when clicked will open the browser on the "XML Schema Part 1:Structures" specification at exactly the point where the error is described thus allowing you to understand the reason for that error.

Figure 4.23. Link to specification for XML Schema errors



Quick Document Browsing Using Bookmarks

The concept of bookmark is the same as in other IDEs: the user can mark a position in one edited document so that he can quickly return after further editing and browsing through one or more documents opened at the same time. Up to nine distinct bookmarks can be placed in any opened document. Configurable shortcut key strokes are available for placing bookmarks and for quick return to any of the marked positions.

The key strokes can be configured from Options-> Preferences->Menu shortcut keys.

A bookmark can be placed from Edit-> Bookmarks->Create, from Edit-> Bookmarks->Bookmarks quick creation and by clicking in the margin of the editing area, to the left of the line number area, reserved for bookmarks.

Quickly switching to a position marked by a bookmark can be done by Edit-> Bookmarks->Go to.

Chapter 5. Transforming Documents

XML is designed to store, carry, and exchange data, not to display data. When we want to view the data we must either have an XML compliant user agent or transform it to a format that can be read by other user agents. This process is known as transformation.

Within the current version of <oXygen/> you can transform your XML documents to the following formats without having to exit from the application. For transformation to formats not listed simply install the tool chain required to perform the transformation and process the xml files created with <oXygen/> in accordance with the processor instructions.

PDF — Adobe Portable Document Format (PDF) is a compact binary file format that can be viewed and printed by anyone, anywhere across a broad range of hardware and software using the free PDF Viewer from Adobe [http://www.adobe.com/products/acrobat/readstep.html].

PS — PostScript is the leading printing technology from Adobe [http://www.adobe.com:80/products/postscript/main.html] for high-quality, best-in-class printing solutions ranging from desktop devices to the most advanced digital presses, platemakers, and large format image setters in the world. Postscript files can be viewed using viewers such as GhostScript, but are more commonly created as a prepress format.

TXT — Text files are Plain ASCII Text and can be opened in any text editor or word processor.

XML — XML stands for EXtensible Markup Language and is a W3C [http://www.w3c.org/XML/] standard markup language, much like HTML, which was designed to describe data. XML tags are not predefined in XML. You must define your own tags. XML uses a Document Type Definition (DTD), an XML Schema or a Relax NG schema to describe the data. XML with a DTD, XML Schema or Relax NG schema is designed to be self-descriptive. XML is not a replacement for HTML. XML and HTML were designed with different goals:

- XML was designed to describe data and to focus on what data is.
- HTML was designed to display data and to focus on how data looks.
- HTML is about displaying information, XML is about describing information.

XHTML — XHTML stands for EXtensible HyperText Markup Language, a W3C [http://www.w3c.org/MarkUp/] standard. XHTML is aimed to replace HTML. While almost identical to HTML 4.01, XHTML is a stricter and cleaner version of HTML. XHTML is HTML defined as an XML application.

All formatting during a transformation is provided under the control of an Extensible Stylesheet (XSLT). Specifying the appropriate XSLT enables transformation to the above formats and preparation of output files for specific user agent viewing applications, including:

HTML — HTML stands for Hyper Text Markup Language and is a W3C Standard [http://www.w3c.org/MarkUp/] for the World Wide Web. HTML is a text file containing small markup tags. The markup tags tell the Web browser how to display the page. An HTML file must have an htm or html file extension. An HTML file can be created using a simple text editor.

HTML Help — Microsoft HTML Help [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/htmlhelp/html/vsconHH1Start.asp?frame=true] is the standard help system for the Windows platform. Authors can use HTML Help to create online help for a software application or to create content for a multimedia title or Web site. Developers can use the HTML Help API to program a host application or hook up context-sensitive help to an application.

JavaHelp — JavaHelp software is a full-featured, platform-independent, extensible help system from Sun Microsystems [http://java.sun.com/products/javahelp/index.html] that enables developers and authors to incorporate online help in applets, components, applications, operating systems, and devices. JavaHelp is a free product and the binaries for JavaHelp are redistributable.

Many other target formats are possible, these are the most popular. The basic condition for transformation to any format is that your document is valid against a given DTD and that the XSLT (XSL), used for transformation is compatible with the DTD.

An XSL stylesheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an output document by using special formatting vocabulary.

<oXygen/> supports XSLT/XPath version 1.0 using Saxon 6.5.3, Xalan and XSLT/XPath 2.0 by using Saxon 8.1B. The editor switches between the tag-insight list of elements for the two standards automatically by examining the version attribute of the stylesheet. Also the validation is done in function of the stylesheet version.

XSL consists of three parts:

XSL Transformations — XSLT is a language for transforming XML documents.

XML Path Language — XPath is an expression language used by XSLT to access or refer parts of an XML document. (XPath is also used by the XML Linking specification).

XSL Formatting Objects — XSL-FO is an XML vocabulary for specifying formatting semantics.

The <oXygen/> installation package is distributed with the Apache [http://www.apache.org] FOP [http://xml.apache.org/fop/index.html] (Formatting Objects Processor) for rendering your XML documents to PDF. FOP is a print and output independent formatter driven by XSL Formatting Objects. FOP is implemented as a Java application that reads a formatting object tree and renders the resulting pages to a specified output.

Tip

To include PNG images in the final PDF document you need the JIMI [http://java.sun.com/products/jimi/] or JAI [http://java.sun.com/products/java-media/jai/] libraries. For TIFF images you need the JAI [http://java.sun.com/products/java-media/jai/] library. The JIMI and JAI libraries are not bundled with <oXygen/> due to Sun's licensing. Using them is as easy as downloading them and copying the necessary jar files (required by the library documentation) in the lib subdirectory of the <oXygen/> installation directory. This means JimiProClasses.zip for JIMI and jai_core.jar, jai_codec.jar and mlibwrapper_jai.jar for JAI. For the JAI package you also need to include the directory containing the native libraries (mlib_jai.dll and mlib_jai_mmx.dll on Windows) in the PATH system variable.

The MacOS X version of the JAI library can be downloaded from http://www.apple.com/downloads/macosx/apple/java3dandjavaadvancedimagingupdate.html. In order to use it, install the downloaded package.

Other FO processors can be configured in the Preferences -> FO Processors option for use in document transformation.

Transformation Scenarios

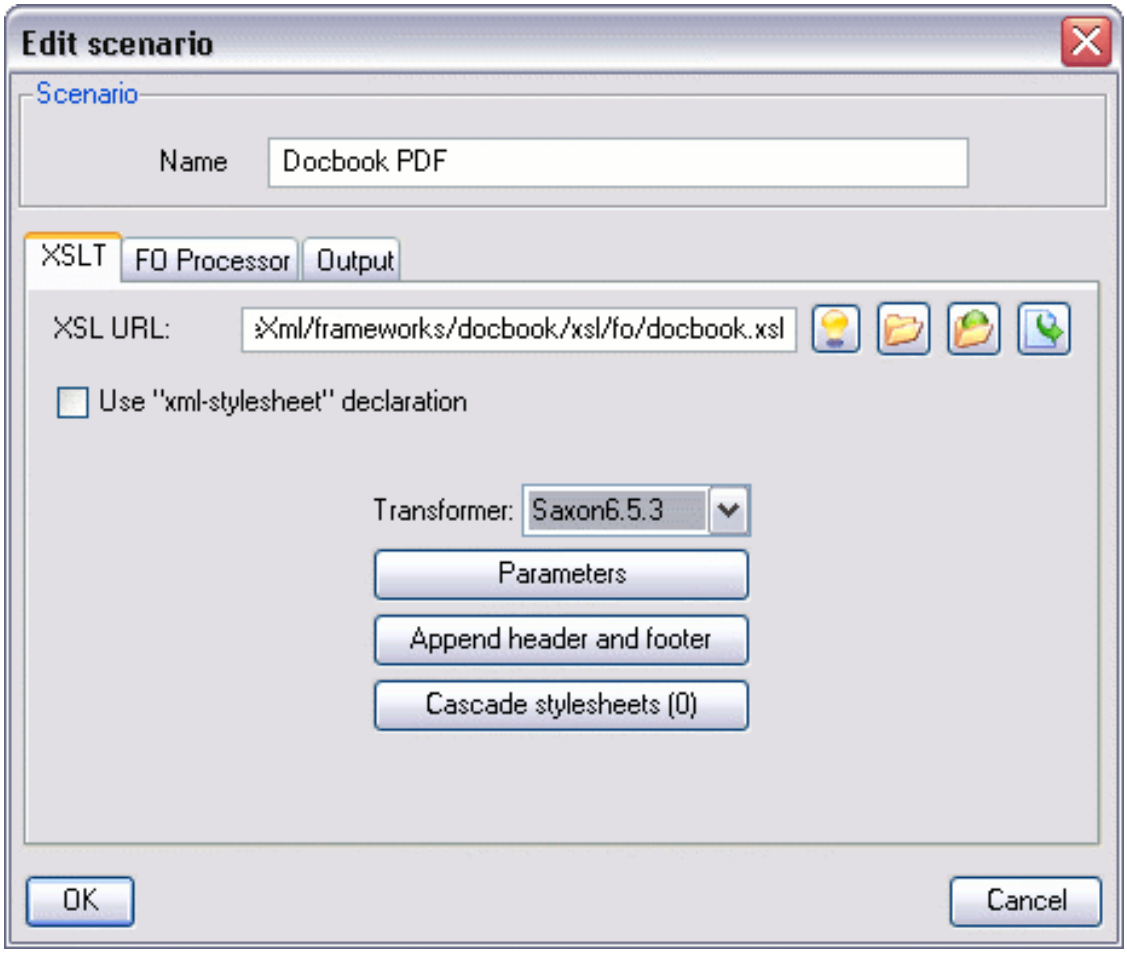
Before transforming the current edited XML document in <oXygen/> one must define a transformation scenario to apply to that document. A scenario is a set of values for various parameters defining a transformation. It is not tied to any particular document but to a document type:

Scenarios that apply to XML files	Such a scenario contains the location of an XSLT stylesheet that is applied on the edited XML document and other transform parameters.
Scenarios that apply to XSL files	Such a scenario contains the location of an XML document that the edited XSL file is applied on and other transform parameters.

The Configure Scenario dialog is used to associate a scenario from the list of all scenarios with the edited document by selecting an entry from the list. The dialog is opened by pressing the Configure Transformation Scenario button on the toolbar of the document view. Once selected the scenario will be applied with only one click on the Apply Transformation button on the same toolbar. Pressing the Apply Transformation button before associating a scenario with the edited document will invoke first the Configure Scenario dialog and then apply the selected scenario.

Open the Configure Transformation dialog by selecting Document->Configure transformation scenario (**Ctrl+Shift+C**).

Figure 5.1. The Configure Transformation Dialog



Complete the dialog as follows:

XSLT Tab

Use the XSLT tab to specify an input XSL file to be used for the transformation. You can also add XSLT parameters and append header and footer URL's to be included in the transformation. To apply a cascade of stylesheets the user can set the list of stylesheets applied after the stylesheet from the XSL URL field in the dialog displayed after pressing the "Cascade Stylesheets" button.The user can choose between Xalan and Saxon when configuring the transformation. Saxon is faster on Docbook stylesheets.

FOP Tab

Use the FOP tab to enable/disable use of FOP during a transformation. FOP input may be provided from the XSLT output or the edited document source. <oXygen/> is supplied with the Apache FOP, but supports definition and use of any third party processor. Default output method is set to use PDF, but PS and TXT are also configured. You may add and define any method supported by your FOP.

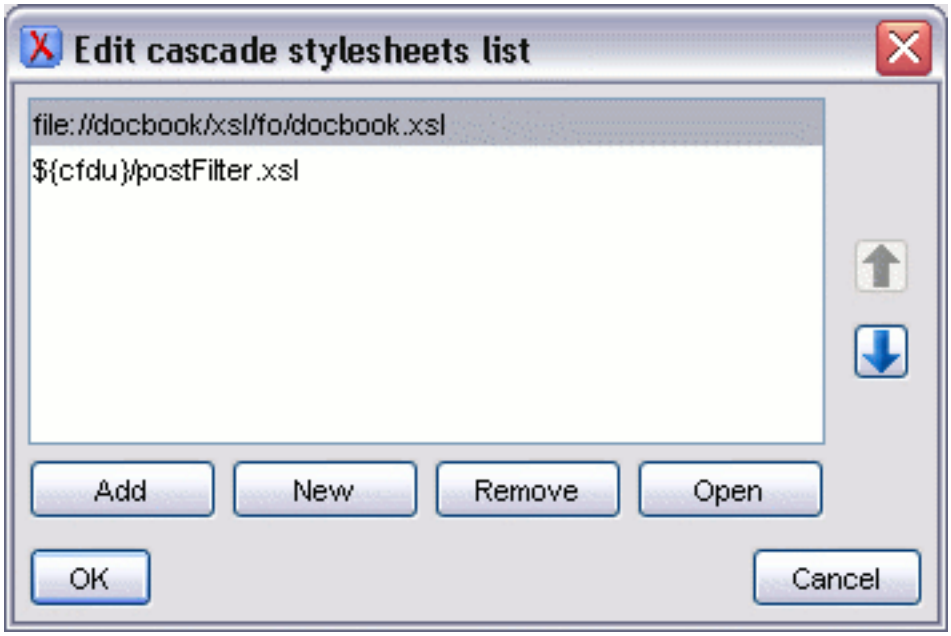
Output Tab

Use the Output Tab to specify the output path where target output files will be saved. When performing an XHTML transformation the relative path for image locations must be provided in order to ensure that image paths will be correctly resolved in order to be displayed in the output files. When using FOP this is not required as images will be embedded within the output PDF or PS. This option will therefore be

disabled during FOP transformations.

The list of cascade stylesheets can be edited in the dialog opened by the button "Cascade Stylesheets".

Figure 5.2. Edit cascade stylesheets list dialog



Add	Adds a stylesheet in the "Cascade stylesheets" list using a file browser dialog , also you can type a macro in the file name field of the browser dialog. The name of the stylesheet will be added in the list after the current selection.
New	Opens a dialog in which you can type the name of a stylesheet.The name is considerd relative to the URL of the current edited XML document..You can use macros in the name of the stylesheet. The name of the stylesheet will be added in the list after the current selection.
Remove	Deletes the selected stylesheet from the "Cascade stylesheets" list.
Open	Opens the selected stylesheet in a separate view .
Up	Move the selected stylesheet up in the list.
Down	Move the selected stylesheet down in the list.

In the fields reserved for: input URL (XSL URL or XML URL, depending on scenario type), header URL, footer URL, the URLs in the list of cascade stylesheets, image base URL, the user can use the following macros:

${home}	the path of the user home
${cfdu}	current file directory url - the path of the current edited document up to the name of the parent directory as URL

${cfn} current file name - the name of the current edited document without extension and parent directory

In the Save As field from the Output tab, the user can use the following macros: ${home}, ${cfd}, ${cfn}.

${cfd} current file directory - the path of the current edited document up to the name of the parent directory

The macros defined here can also be used in the values set for the parameters of the transformation (e.g. base.dir).

Creating a Scenario

Use the following procedure to create a scenario.

1. Select Document->Configure transformation scenario (**Ctrl+Shift+C**) to open the Configure Transformation dialog.
2. Click the Duplicate Scenario icon to the right of the top combo box to create a copy of the current "Scenario".
3. Double-click in the "Name" field to select the exiting text.
4. Type a new name.
5. Click OK or Transform Now to save the "Scenario".

The default scenario

If one presses the Apply Transformation Scenario toolbar button, currently there is no scenario associated with the edited document and the edited document contains a "xml-stylesheet" processing instruction reffering to a XSLT stylesheet (commonly used for display in Internet browsers), then <oXygen/> will prompt the user and offer him the option to associate the document with a built-in default scenario containing in the *XSL URL* field the URL from the *href* attribute of the processing instruction. This scenario will have the "Use xml-stylesheet declaration" checkbox set by default, will use Saxon as transformation engine, will perform no FO processing and will store the result in a file with the same URL as the edited document except the extension which will be changed to html. The name and path will be preserved because the output file name is specified in terms of two macros: ${cfd} and ${cfn}.

Import/Export Transformation Scenarios

The option to Export Transformation Scenarios is used to store all the scenarios in a separate file , a properties file. In this file will also be saved the associations between document urls and scenarios. The saved urls are absolute . You can load the saved scenarios using Import Transformation Scenarios option. All the imported scenarios will have added to the name the word 'import'.

Example Transformation Scenarios

The following examples use the DocBook XSL Stylesheets to illustrate how to configure <oXygen/> for transformation to the various target formats.

The following steps are common to all the example procedures below.

1. Set the editor focus to the document to be transformed.
2. Select Document->Configure transformation scenario (**Ctrl+Shift+C**) to open the Configure Transformation dialog.
3. Select the XSLT tab.
4. Click the "Browse for an input XSL file button". The Open dialog is displayed.

Note

During transformations the Editor Status Bar will show "Transformation - in progress". The transformation is successfully complete when the message "XSL transformation successful" displays. If the transform fails the message "XSL transformation failed" is displayed as an error message in the Messages Panel. The user can stop the transformation process at any point by pressing the "Stop transformation" button. In this case the message displayed in the status bar will be "Transformation stopped by user".

PDF Output

1. Change directory to **[oxygen]/frameworks/docbook/xsl/fo/**.
2. Select `docbook.xsl`, click Open. The dialog closes.
3. Select the FOP tab.
4. Check the Perform FOP option. The remaining options are enabled.
5. Select the following options:
   a. XSLT result as input.
   b. PDF as method.
   c. Built-in(Apache FOP) as processor.
6. Select the Output tab.
7. In the "Save As" field enter the output file name relativ to the current directory (`YourFile-Name.pdf`) or the path and output file name (`C:\FileDirectory\YourFileName.pdf`).
8. Optionally, uncheck the XHTML and XML check boxes in the Show As group.
9. Click Transform Now. The transformation is started.

PS Output

1. Change directory to **[oxygen]/frameworks/docbook/xsl/fo/**.
2. Select `docbook.xsl`, click Open. The dialog closes.
3. Select the FOP tab.
4. Check the Perform FOP option. The remaining options are enabled.
5. Select the following options:
   a. XSLT result as input.
   b. PS as method.
   c. Built-in(Apache FOP) as processor.
6. Select the Output tab.
7. In the "Save As" field enter the output file name relativ to the current directory (`YourFileName.ps`) or the path and output file name (`C:\FileDirectory\YourFileName.ps`).
8. Optionally, uncheck the XHTML and XML check boxes in the Show As group.
9. Click Transform Now. The transformation is started.

TXT Output

1. Change directory to **[oxygen]/frameworks/docbook/xsl/fo/**.
2. Select `docbook.xsl`, click Open. The dialog closes.
3. Select the FOP tab.
4. Check the Perform FOP option. The remaining options are enabled.
5. Select the following options:
   a. XSLT result as input.
   b. TXT as method.
   c. Built-in(Apache FOP) as processor.
6. Select the Output tab.
7. In the "Save As" field enter the output file name relativ to the current directory (`YourFileName.txt`) or the path and output file name (`C:\FileDirectory\YourFileName.txt`).

8. Optionally, uncheck the XHTML and XML check boxes in the Show As group.
9. Click Transform Now. The transformation is started.

HTML Output

1. Change directory to **[oxygen]/frameworks/docbook/xsl/html/**.
2. Select `docbook.xsl`, click Open. The dialog closes.
3. Select the FOP tab.
4. Uncheck the Perform FOP option. The FOP options are disabled.
5. Select the Output tab.
6. In the "Save As" field enter the output file name relativ to the current directory (`YourFileName.html`) or the path and output file name (`C:\FileDirectory\YourFileName.html`).

    a. If your pictures are not located relative to the out location, check the XHTML check box in the Show As group.

    b. Specify the path to the folder or URL where the pictures are located

7. Click Transform Now. The transformation is started.

HTML Help Output

1. Change directory to **[oxygen]/frameworks/docbook/xsl/htmlhelp/**.
2. Select `htmlhelp.xsl`, click Open. The dialog closes.
3. Set the XSLT parameter base.dir, it identifies the output directory. (If not specified, the output directory is system dependent.)
4. Select the FOP tab.
5. Uncheck the Perform FOP option. The FOP options are disabled.
6. Click Transform Now. The transformation is started.

JavaHelp Output

1. Change directory to **[oxygen]/frameworks/docbook/xsl/javahelp/**.
2. Select `javahelp.xsl`, click Open. The dialog closes.
3. Set the XSLT parameter base.dir, it identifies the output directory. (If not specified, the output directory is system dependent.)
4. Select the FOP tab.
5. Uncheck the Perform FOP option. The FOP options are disabled.
6. Click Transform Now. The transformation is started.

XHTML Output

1. Change directory to **[oxygen]/frameworks/docbook/xsl/xhtml/**.
2. Select `docbook.xsl`, click Open. The dialog closes.
3. Select the FOP tab.
4. Uncheck the Perform FOP option. The FOP options are disabled.
5. Select the Output tab.
6. In the "Save As" field enter the output file name relativ to the current directory (`YourFileName.html`) or the path and output file name (`C:\FileDirectory\YourFileName.html`).

    a. If your pictures are not located relative to the out location, check the XHTML check box in the Show As group.

    b. Specify the path to the folder or URL where the pictures are located

7. Click Transform Now. The transformation is started.

Configuring the extension paths for transformer processors.

Both the Xalan and Saxon processors support the use of extension elements and extension functions. Unlike a literal result element, which the stylesheet simply transfers to the result tree, an extension element performs an action. The extension is usually used because the xslt stylesheet fails in providing adequate functions to the user for accomplishing a more complex task.

Samples on how to use extensions can be found at:

- http://xml.apache.org/xalan-j/extensions.html

- http://saxon.sourceforge.net/saxon6.5.2/extensions.html

The following needs to be accomplished in order for the transformer to find and use succesfully the Java extension classes:

- Set the property "com.oxygenxml.additional.classpath" to contain the additional paths to the directories containing the used Java extension classes or jars.

Example of setting two directories called "test1" and "test2" located in the <oXygen/> root directory as extension paths (containing the extension classes or jars):

- For users who use a script (bat or sh) to start <oXygen/> add the following parameter "-Dcom.oxygenxml.additional.classpath=test1;test2" to the java command line in your script file (oxygen.bat or oxygen.sh). Example: "java -Xmx256m -Dcom.oxygenxml.additional.classpath=test1;test2; -cp %CP% ro.sync.exml.Oxygen %1 %2 %3 %4 %5" .
- For users who use an executable (exe) to start <oXygen/> add the following parameter "com.oxygenxml.additional.classpath=test1;test2" to the "oxygen.lax" file situated in the <oXygen/> root directory.

After the parameter is set, Java classes and jars from the extension paths are dinamically loaded and can be used by the transformer processors with no extra setting.

Chapter 6. XSLT Debugger

Overview

The main workspace of <oXygen/> supports two perspectives. The first is the standard editing perspective that provides general features and functions for the development of XML documents and other programming languages. The second is the Debugger perspective. The Debugger perspective is started by clicking the Debugger button located on the <oXygen/> main toolbar. To switch back to Editor perspective simply click the Editor button that is adjacent to the Debugger button on the toolbar. Users can toggle between Debugger and Editor perspective as required by clicking either buttons.Table 3.12, " Description of Main Toolbar Buttons "

This chapter explains the Debugger mode functionality, which provides a rich set of features for development, testing and solving of XSL problems, including:

- Support for Saxon and Xalan XSLT engines.
- Stepping capabilities: step in, step over, step out, run, run to cursor, run to end, pause, stop.
- Back mapping between every piece of output and style element /source context who generate it .
- Breakpoints on both source and style documents.
- Call stack view on both source and style documents.
- Trace history on both source and style documents.
- Support for XPath expression evaluation during debugging.
- Step into imported/included stylesheets as well as included source entities.
- Available templates and hits count.
- Variables view.
- Dynamic output generation.

Layout

When the Debugger perspective is selected the main workspace of the <oXygen/> editor perspective is replaced by the Debugger perspective interface. An example of what the Debugger interface might look like is shown below. This interface is comprised of four panes as follows:

1. Source document view (XML)
2. Stylesheet document view (XSL)
3. Output View
4. Control view

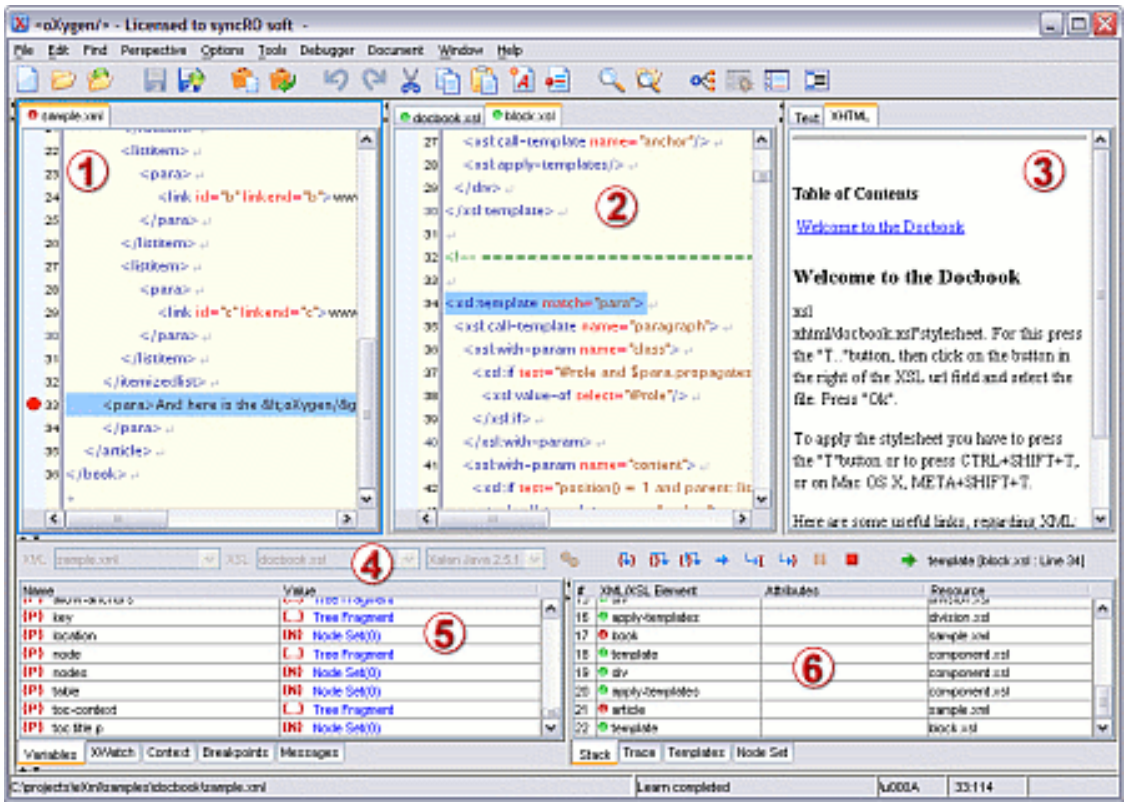
XML documents and XSL stylesheets that are opened in Editor perspective are automatically sorted into

the first two panes. When multiple files of each type are opened, the individual documents/stylesheets are separated using the familiar tab management system of the Editor perspective. Selecting a tab brings the document/stylesheet into focus and enables editing without toggling back to the Editor perspective.

When editing in the Editor perspective the editor toolbar is displayed. In Debugger mode this toolbar is not available, however the functions are still accessible from the Document menu same as the context menus that are activated by a right click of the mouse. Bookmarks cannot be set in Debugger perspective. The bookmark functionality is replaced instead with functionality required for setting and removing breakpoints.

During debugging the current execution node is highlighted on both document (XML) and stylesheet (XSL) views.

Figure 6.1. Debugger Mode Interface



Source document view (XML)

Displays and allows editing of data or document oriented XML files (documents).

Stylesheet document view (XSL)

Displays and allows editing of XSL files(stylesheets).

Output document view

Displays the transformed output that results from the input of a selected document (XML) and selected

stylesheet (XSL) to the transformer. The result of transformation is dynamically written as the transformation is processed.

There are two views for the output: a text view (with XML syntax highlight) and an XHTML view. For large output the XHTML view can be can disabled (see Debugger Settings).

Control view

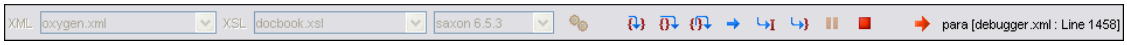
The control view provides functionality for configuration and control of debugging operations. It also provides a series of Information View types. This pane is comprised of two parts:

- Control Toolbar
- Information View

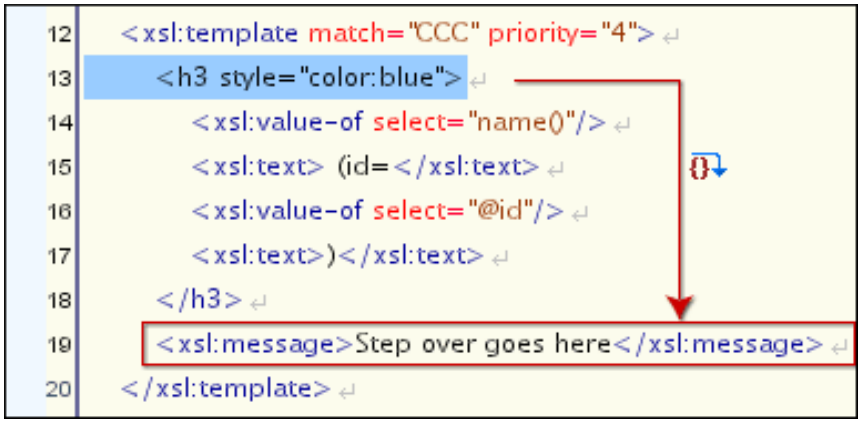
Control Toolbar

The toolbar contains all actions needed in order to configure and control the debug process. Items are described below from left to right as they appear in the toolbar.

Figure 6.2. Control Toolbar



XML source selector	The selection represents the source document to be used as input by the transformation engine. The selection list is filled-in with all opened files (the XML ones being emphasized). This gives you the possibility to use other file types as source.
XSL stylesheet selector	The selection represents the stylesheet document to be used by the transformation engine. The selection list is filled-in with all opened files (the XSL ones being emphasized).
XSLT engine selector	Lists the available XSLT processors
XSLT parameters	XSLT parameters to be used by the transformation.
Step into	Starts the debugging process and runs until the next stylesheet node (next step in transformation).
Step over	Executes the current stylesheet node (including its sub-elements) and goes to next node in document order (usually the next sibling of the current node).



Step out

Steps out to the parent node (equivalent to the Step over on the parent).

Run

Starts the debugging process and runs until the first breakpoint is encountered or until the end of transformation occurs, if no breakpoints are encountered (see Breakpoints view).

Run to cursor

Starts the debugging process and runs until one of the following conditions occur: the line of cursor is reached, a valid breakpoint is reached or end of execution.

Run to end

Runs the transformation until the end, without taking into account any enabled breakpoints that might be set.

Pause

Interrupts the current transformation. This is useful for long transformations (Docbook for instance) when you want to find out what point the transformation has reached. The transformation can be resumed after.

Stop

Ends the transformation process.

Show current context

Highlights the current execution nodes in both the document and stylesheet files. This feature is useful when you lost the current selection.

Current step info

Shows information about the current node reached by the debugging process. The details shown are:

- Icon to show the action (entering or leaving node).
- Node name.

- Resource file where the node is located.
- Line number inside resource file where the node is located.

Note

Accelerator key combinations can be associated with debugger actions in the <oXygen/> preference dialog. Select Options->Preferences+Menu Shortcut Keys->Debugger category

Information View

The information view is comprised of two panes that are used to display various types of information that can be used to understand the transformation process. For each information type there is a corresponding tab. While running a transformation, relevant events are displayed in the various information views. This enables the developer to obtain a clear view of the transformation progress. Using the Debug controls developers can easily isolate parts of stylesheet therefore they may be understood and modified. The information types include (for a more detailed discussion on each information type see Understanding Information Views):

Left side Information View Classes

- Context node view
- XPath watch view
- Breakpoints view
- Messages view
- Variables view

Right side Information View Classes

- Stack view
- Trace history view
- Templates view
- Node set view

Working with Debugger

This section explains the working process involving the use of Debugger perspective.

Getting Started

<oXygen/> provides two perspectives, Editor and Debugger. <oXygen/> starts by default in the Editor perspective. Switching between Editor and Debugger perspectives is easy and can be done at any time

during a working session even when no files have been opened. To switch to Debugger perspective click the  button .

Unlike Editor perspective, Debugger perspective requires that at least one document (XML) and one stylesheet (XSL) are opened before the debug functionality and features become of any use. These files can be opened while in Editor perspective before switching to Debugger perspective, or directly from within the Debugger perspective. You can switch back to Editor perspective by clicking the Editor  button .

When switching from Editor perspective to Debugger perspective, the opened files are sorted by extension into the Source document view (XML) and Stylesheet document view (XSL) panes.

The Debug Process

The debug procedure described below (see Typical Debug Process), assumes <oXygen/> is already in Debugger perspective and at least one document (XML) and one stylesheet (XSL) are already opened. Some samples have been provided in order to get used with the XSLT debugging process. They can be found in the `samples/debugger` subdirectory of your <oXygen/> installation.

When a debug process is running, it is adviseable to stop the process before attempting to edit source documents or stylesheets. Editing during an active debug process will result in inaccurate informations being displayed in the Information View. The process must also be stopped in order to switch to Editor perspective.

During the debug process, if the transformation engine reaches a node from a file that has not been opened, this file will be opened into the corresponding pane and the node will be highlighted. This is most likely to happen in the cases of XML entity files or XSL imported/included files.

Errors encountered during debugging are reported in the <oXygen/> results panel.

At the end of debugging process, only the content from the following views is preserved (all other views are cleared):

- XPath watch view
- Messages view
- Trace history view
- Templates view

Procedure 6.1. Typical Debug Process

1. From Source document view (XML) select a source document.
2. From Stylesheet document view (XSL) select a stylesheet document.
3. From the Control Toolbar use the XML source selector control to select a source document.
4. From the Control Toolbar use the XSL stylesheet selector control to select a stylesheet.

5. From the Control Toolbar use the XSLT engine selector control to select one of the available processing engines.

6. Configure the XSLT parameters. Once set, these parameters are preserved between debugging sessions.

7. Start the debugging using the active control buttons (see Control view for description of control functions).

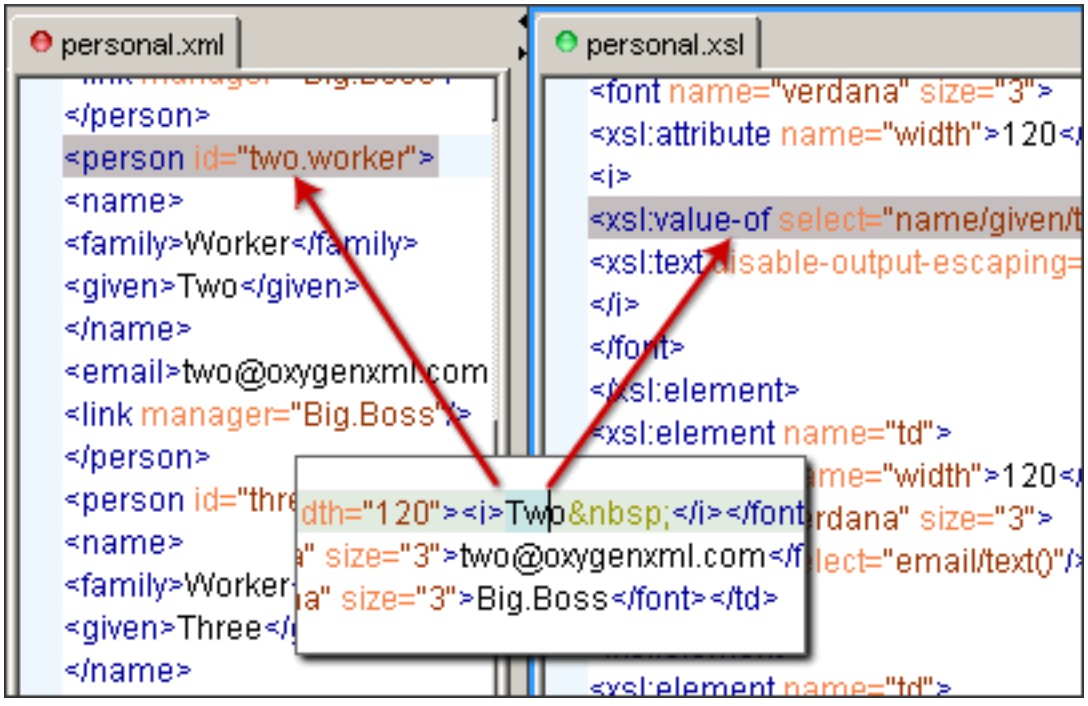
Output to Source Mapping

Every section of the output it is generated by an XSL stylesheet element in the context of an XML source node.

During debugging, it is important to know this mapping from output to source in order to quickly spot the templates with problems. Some of the debugging capabilities, for example "Step in" can be used for this purpose. Using "Step in" you can see how output is generated and link it with the style element being executed in the current source context. However, this can become difficult on complex stylesheets that generates a large output.

Output to source mapping is a powerful feature that makes this mapping persistent that is you can click on th text from the Output document view and the editor will select the XML source context and the XSL element that generated the text.

Figure 6.3. Output to Source Mapping

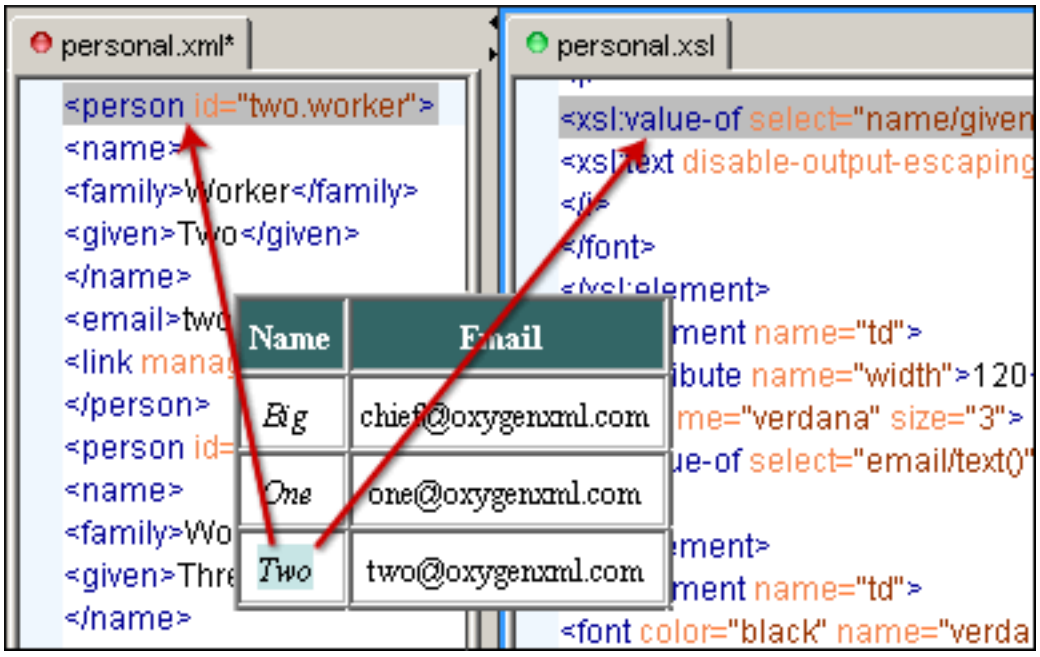

personal.xml
</person>
<person id="two.worker">
<name>
<family>Worker</family>
<given>Two</given>
</name>
<email>two@oxygenxml.com
<link manager="Big.Boss"/>
</person>
<person id="thre
<name>
<family>Worker
<given>Three</
</name>
personal.xsl

<xsl:attribute name="width">120</

<xsl:value-of select="name/given/t
<xsl:text disable-output-escaping=

</xsl:element>
<xsl:element name="td">
dth="120">Two two@oxygenxml.com</f
a" size="3">Big.Boss</td>
me="width">120</
rdana" size="3">
lect="email/text()"/>
<xsl:element name="td">
Additionaly, you can inspect the mapping by clicking a section of the output from the XHTML view tab of the Output document view to have the stylesheet element and the source context highlighted.

Figure 6.4. XHTML Output Mapping



Understanding Information Views

Detailed informations about the debugger status are provided using the information views.

Context node view

The context node is a source node corresponding to the XSL expression being evaluated. It is also called the context of execution. The context node implicitly changes as the processor hits various steps (at the point where XPath expressions are evaluated). This node has the same value as evaluating '.' (dot) XPath expression on XPath watch view.

Figure 6.5. The Context node view

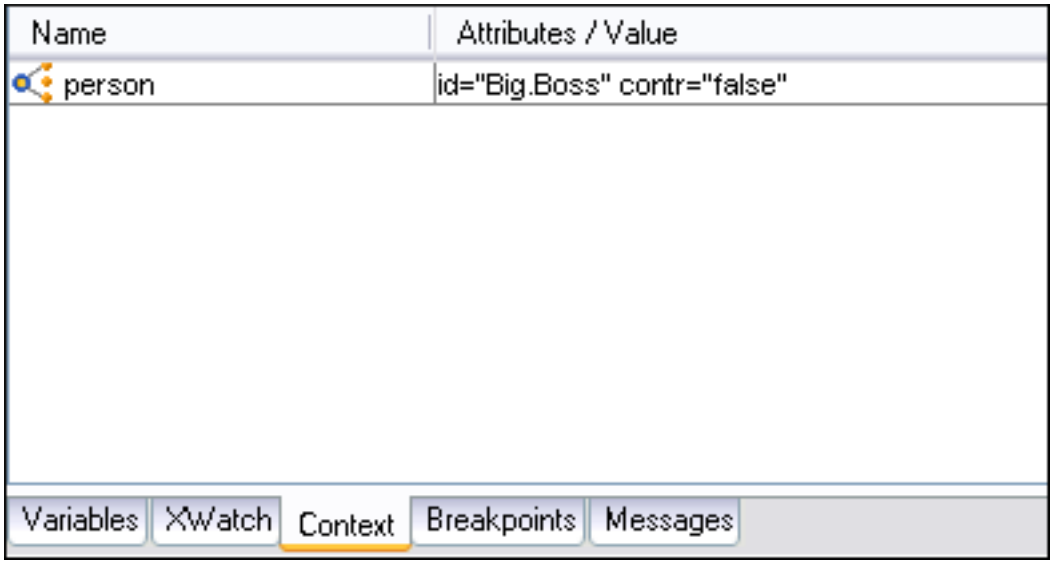


Table 6.1. Context node details

Column	Description
Name	Name of source (XML) node.
Attributes/Value	Attributes or value of the XML node. If attributes exist, they are shown under the form of `attributeName="attributeValue"`, otherwise the text content of the node is shown.

XPath watch view

Shows XPath expressions to be evaluated during debugging. Expressions are evaluated dynamically as the processor changes its source context.

Figure 6.6. The XPath watch view

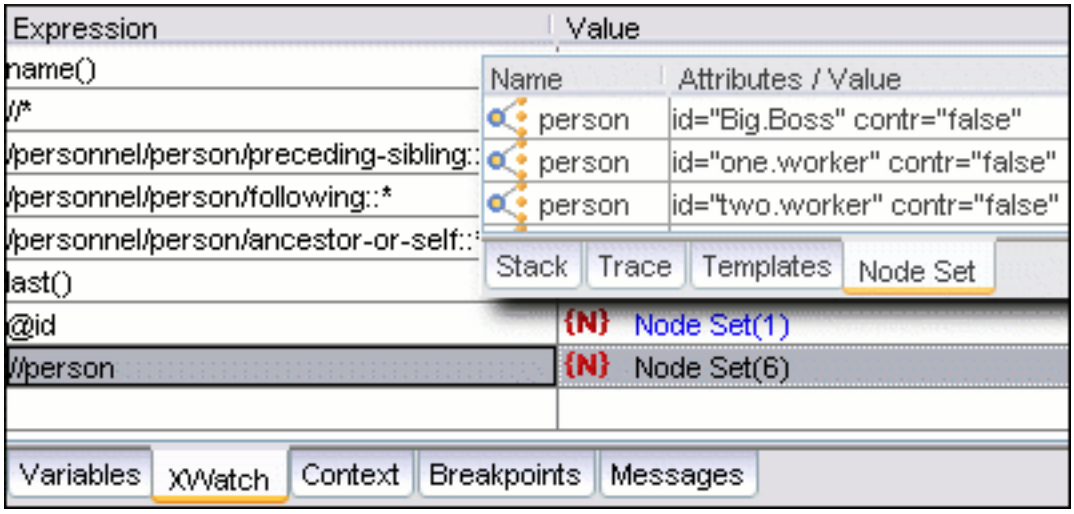


Table 6.2. XWatch details

Column	Description
Expression	XPath expression to be evaluated (should be XPath 1.0 or 2.0 compliant).
Value	Result of XPath expression evaluation. Value has a type (see Possible Values in the section Variables view). For *Node Set* results the number of nodes in the set is shown in parenthesis.

Remarks

- Expressions reffering to variables names are not evaluated. In case of an XPath error, you get an Error line.
- The expression list is not deleted at the end of transformation (it is preserved during sessions).
- To insert a new expression click the last line on the expression column and enter it. Press enter on cell to add and evaluate.
- To delete an expression click on its Expression column and delete its content. Press enter on cell to commit changes.
- If the expression result type is a Node Set you can click on it (Value column) and you will see on the right side its value. (see Node set view).

Breakpoints view

Lists all breakpoints set on opened documents. Once you set a breakpoint it is automatically added in this list. Breakpoints can be set on both XML and XSL documents.

Figure 6.7. The Breakpoints view

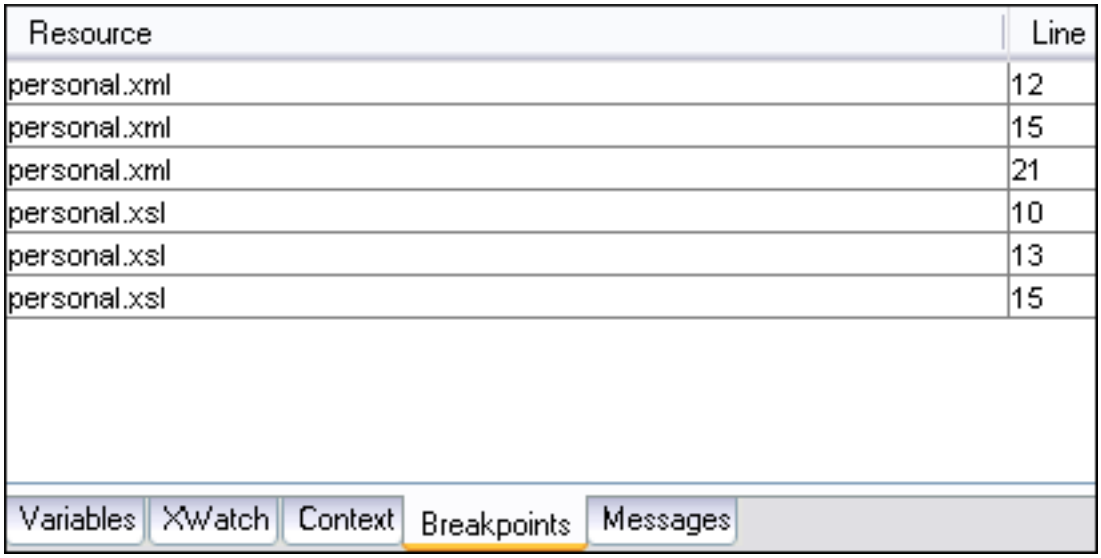


Table 6.3. Breakpoints details

Column	Description
Resource	Resource file where the breakpoint is set. Entire path of resource file is available as tooltip.
Line	Line number inside resource where the breakpoint is set.

Valid Breakpoint

- Not all set breakpoints are valid. For example if the breakpoint is set on one empty or commented line or the line is not reached by the processor (no template to match it, line containing only an end tag), that breakpoint is invalid.
- The contextual menu on table has "Remove" and "Go to" options.
- Clicking a record highlights the breakpoint line into the document.

Messages view

`<xsl:message>` instructions are one way to signal special situations encountered during transformation as well as a raw way of doing the debugging. This view shows all `<xsl:message>` calls executed by the XSLT processor during transformation.

Figure 6.8. The Messages view

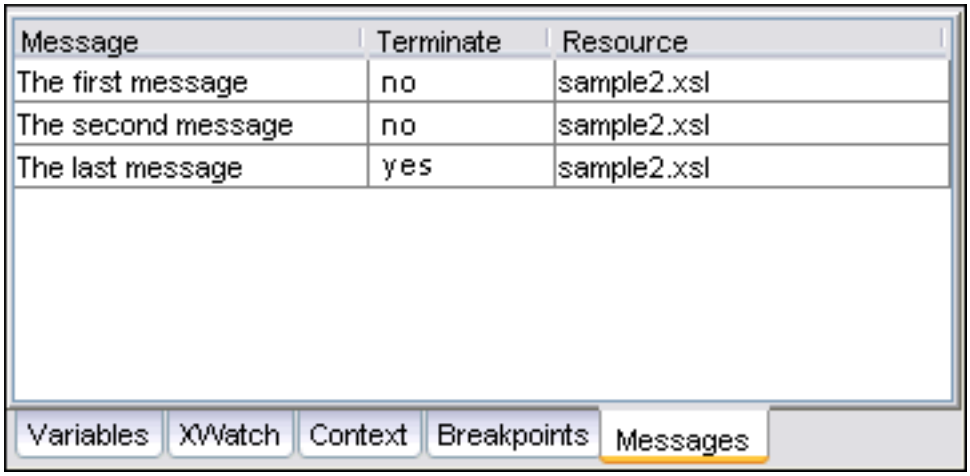


Table 6.4. Messages details

Column	Description
Message	Message content.
Terminate	Signals if processor will terminate the transformation or not once it encounters the message (true/false respectively)
Resource	Resource file where `<xsl:message>` instruction is defined. The complete path of the resource is available as tooltip.

Remarks

- Clicking a record from the table highlights the `<xsl:message>` declaration line.

Stack view

Shows the current execution stack of both source and style nodes. During transformation two stacks are managed: one of source nodes being processed and the other for stylesheet nodes being processed. <oXygen/> shows both node types into one common stack. The source (XML) nodes are preceded by a red color icon while stylesheet nodes are preceded by a green color icon. The advantage of this approach is that you can always see the source scope on which a stylesheet instruction is executed (the last red color node on the stack). The stack is oriented upside down.

Figure 6.9. The Stack view

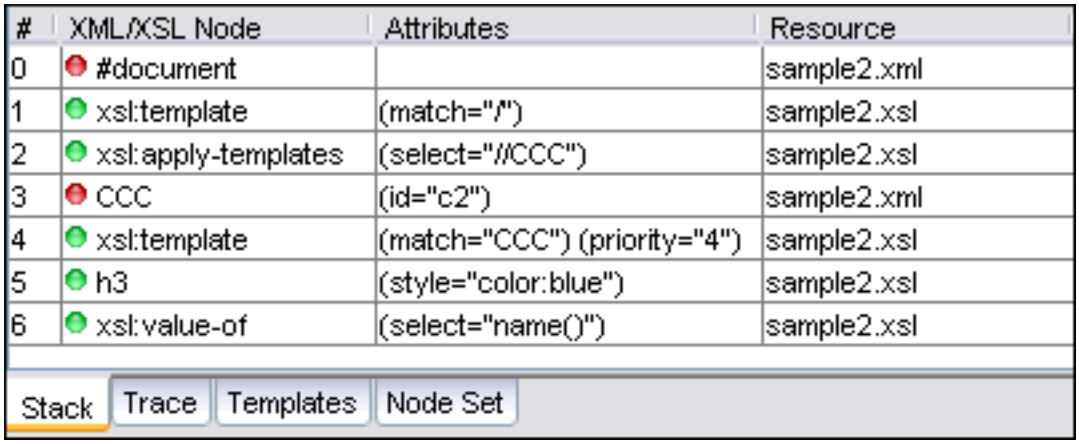


Table 6.5. Stack details

Column	Description
#	Order number, represents the depth of the node (0 is the stack base).
XML/XSL Node	Node from source or stylesheet document currently being processed. One particular stack node is the document root, noted as #document.
Attributes	Attributes of the node (list of `id="value"` pairs).
Resource	Resource file where the node is located. Entire path is available as tooltip.

Remarks

- Clicking a record from the stack highlights that node's location inside resource.
- Using Saxon, the stylesheet elements are qualified with XSL proxy, while on Xalan you only see their names. (example `<xsl:template>` on Saxon and template on Xalan).
- Only Saxon processor shows element attributes.
- Xalan processor shows the "built-in" rules.

Trace history view

Usually the XSLT processors signal the following events during transformation:

- entering a source (XML) node.
- leaving a source (XML) node.
- entering a stylesheet (XSL) node.
- leaving a stylesheet (XSL) node.

The trace history catches all these events, so you can see how the process evolved. The red icon lines denote source nodes while the green icon lines denote stylesheet nodes.

Figure 6.10. The Trace History View

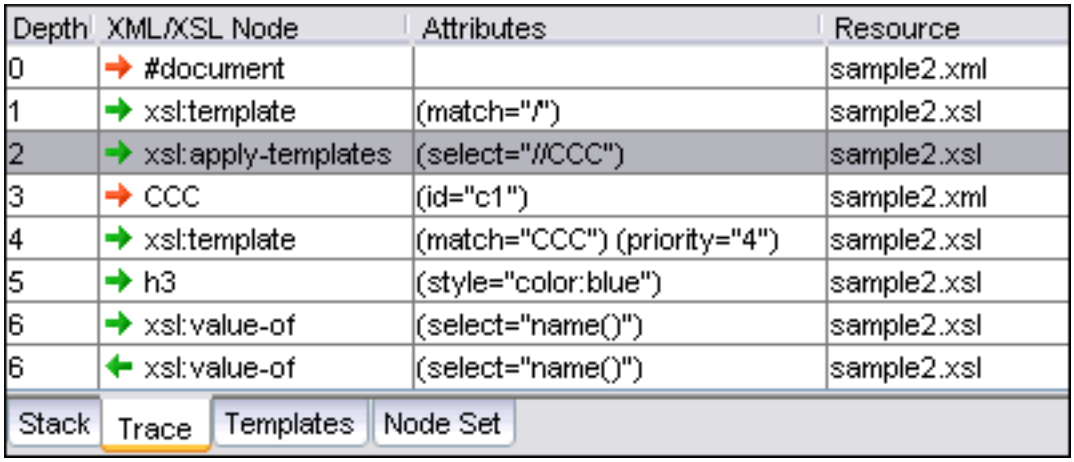


Table 6.6. Trace History details

Column	Description
Depth	Starts from 0 and represents the level of overlapping for that node. This is similar with the # order number from stack at the moment the node was processed.
XML/XSL Node	Represents the node from the processed source or stylesheet document. One particular node is the document root, noted as #document. Every node has an arrow in front of it representing what action was performed on it (entering or leaving).
Attributes	Attributes of the node (list of `id="value"` pairs).
Resource	Resource file where the node is located. Complete path to resource file is provided as tooltip.

Remarks

- Clicking a record highlights that node's location inside the resource.
- Only Saxon processor shows element attributes.
- Xalan processor shows the "built-in" rules.

Templates view

The `<xsl:template>` is the basic element for stylesheets transformation. This view shows all `<xsl:template>` instructions used by the transformation. By seeing the number of hits for each of

the templates you get an idea of the stylesheet coverage by template rules with respect to the input source.

Figure 6.11. The Templates view

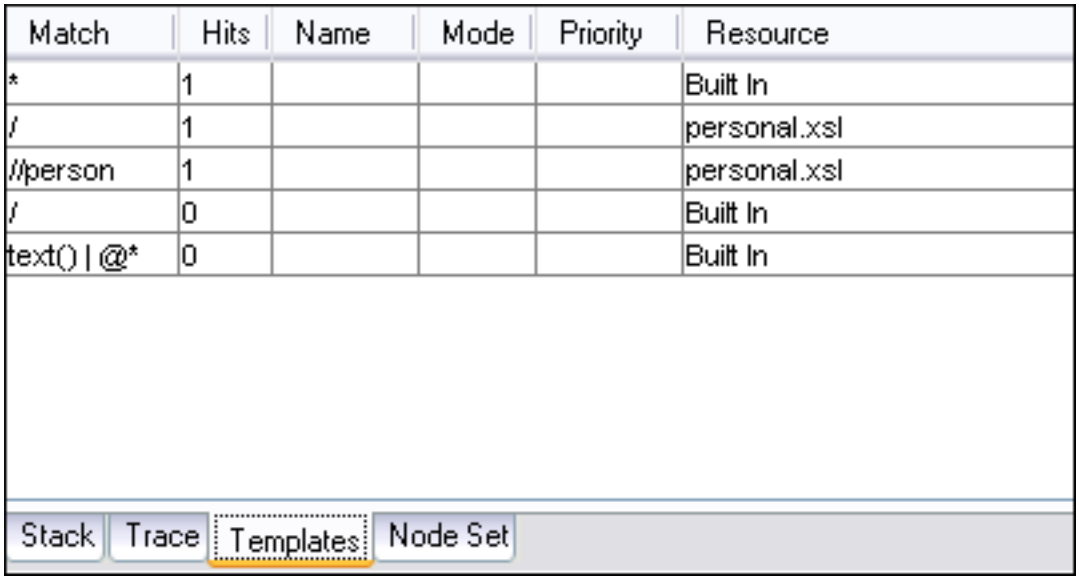


Table 6.7. Templates details

Column	Description
Match	Match attribute of the `<xsl:template>`.
Hits	Number of hits for the `<xsl:template>`. Shows how many times the XSLT processor used this particular template.
Priority	Template priority as established by XSLT processor.
Mode	Mode attribute of the `<xsl:template>`.
Name	Name attribute of the `<xsl:template>`.
Resource	Resource file where template is located. Complete path of resource file is available as tooltip.

Remarks

- Clicking a record highlights that template definition inside resource.
- Saxon only shows the applied templates having at least one hit from the processor. Xalan shows all defined templates, with or without hits.
- The template list is sorted descending on the number of hits.
- Xalan shows the "built-in" rules.

Node set view

This view is always used in relation with Variables view and XPath watch view and shows a nodeset value. Once you click a variable having as value a nodeset or tree fragment or an XPath expression evaluated to a nodeset in the above views the node set view gets updated with the respective value.

Figure 6.12. The Node Set view

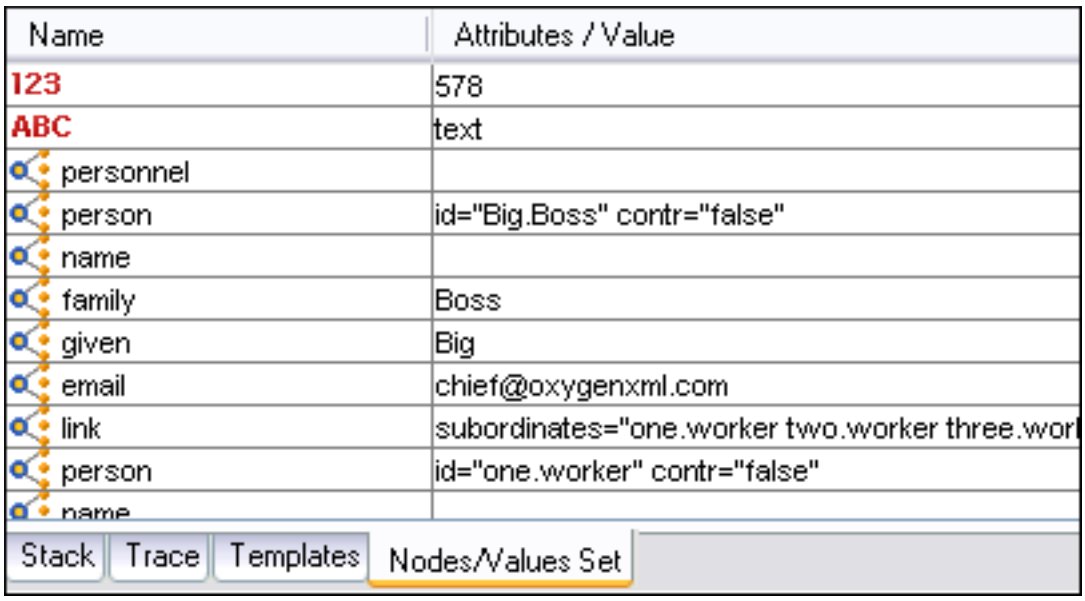


Table 6.8. Node set details

Column	Description
Name	Name of source (XML) node.
Attributes/Value	Attributes or text `content(Value)` of the XML node. If attributes exist, these are shown under the form of `attributeName="attributeValue"`, otherwise the text content of the node is shown.

Remarks

- In case of longer values for Value/Attributes column content, the interface shows three suspension points (...) at the end. A more detailed value is available as tooltip.
- Clicking a record highlights the location of that node into the source or stylesheet view.

Variables view

During transformation variables and parameters play an important role.

<oXygen/> uses the following icons to differentiate variables/parameters:

- V{} Global variable.
- {V} Local variable.
- P{} Global parameter.
- {P} Local parameter.

The values types of a variable are marked by icons explained below:

Possible Values

- 1/0 Boolean.
- ABC String.
- 123 Numeric.
- {N} Node set.
- {...} Tree fragment.
- □ Object.
- ? Any.

Figure 6.13. The Variables view

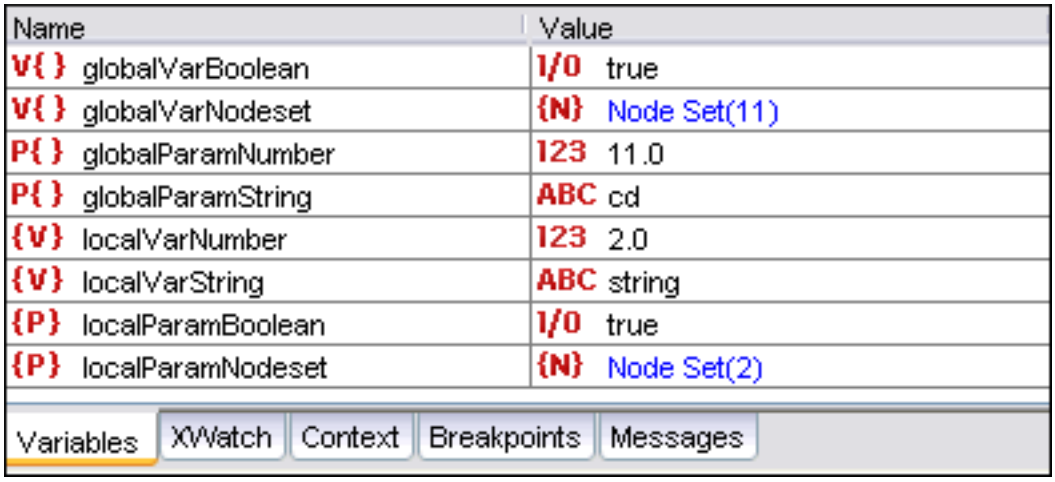


Table 6.9. Variables details

Column	Description
Name	Name of the variable/parameter.
Value	Current value for the variable/parameter.

Remarks

- Clicking a record highlights the variable definition line.
- Variable values could differ depending on the transformation engine used or stylesheet version set.
- If the value of the variable is a node-set or a tree-fragment, clicking on it causes the Node set view to be shown with corresponding set of values.

Chapter 7. WSDL Support

Web Services Description Language Overview

Web Services Description Language (WSDL) is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

<oXygen/> offers the following facilities for WSDL support :

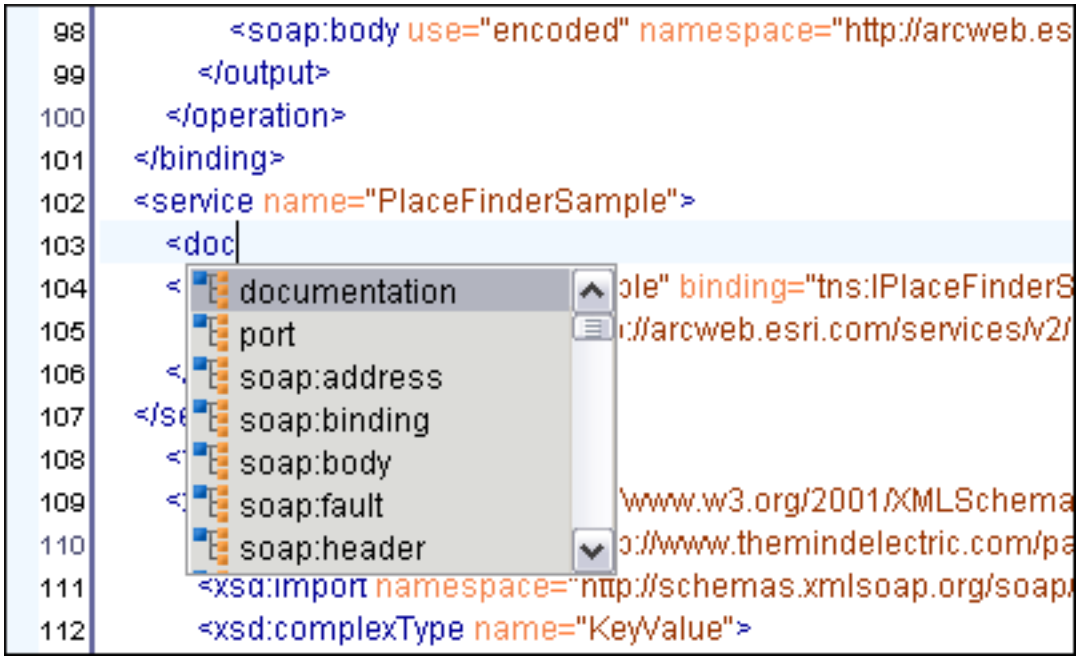
- Editing WSDL files.
- Validating WSDL files.
- Analysing and testing WSDL files.

Editing WSDL files

The WSDL files contain information about the published services, like the name, the message types and the bindings. The editor is offering a way to edit the WSDL files that is similar to editing XML, the tag-insight being driven by a mix of the WSDL and SOAP Schema.

To create a WSDL file, use the File/New and then choose WSDL file.

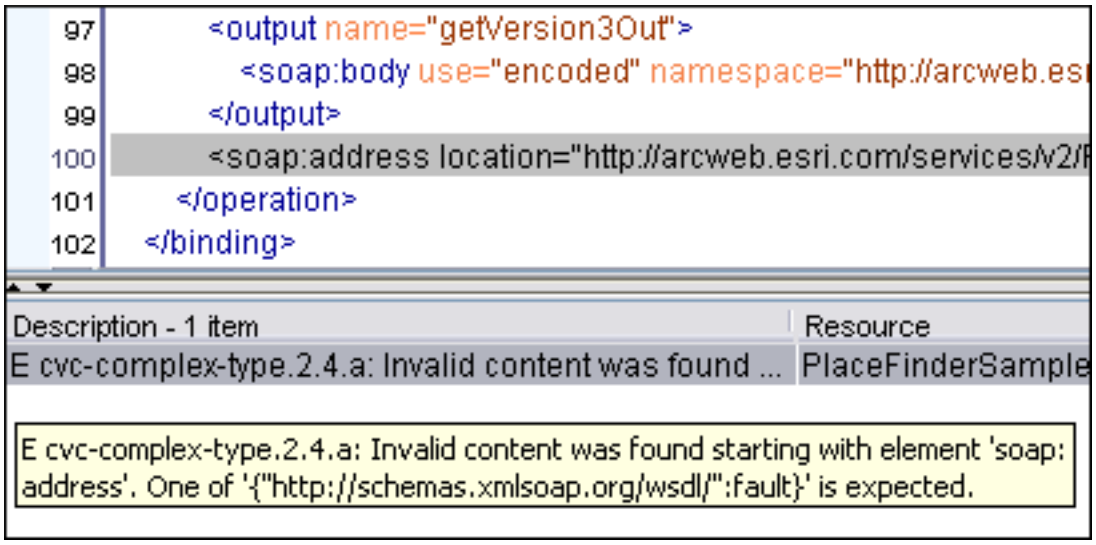
Figure 7.1. Tag insight for WSDL



Validating WSDL files

While editing the Web-Services descriptors you can check their conformance to the WSDL and SOAP schema. You do not need to specify the schema location for the WSDL standard namespaces. In the following example you can see how the errors are reported.

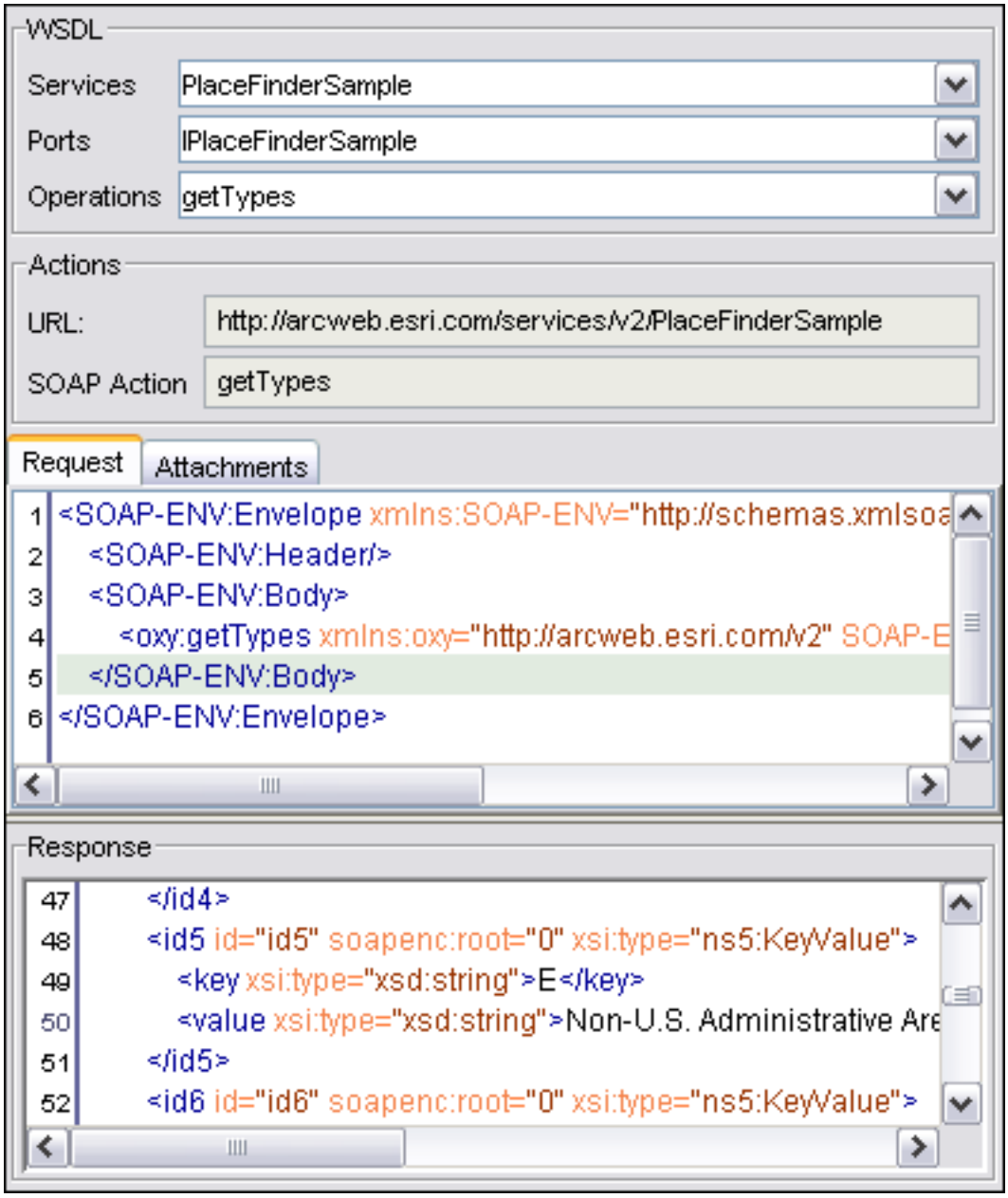
Figure 7.2. Validating a WSDL file



Analysing and testing WSDL files.

After defining the descriptor you can check it to see if the defined messages are accepted by the Web Services server. <oXygen/> is providing two ways of testing, one for the currently edited WSDL file and other for the remote WSDL files that are published on a web server.

Figure 7.3. WSDL Analyser



In case of a remote file you must use the menu option "WSDL SOAP Analyser". In case of the edited document, you can start the analyser from the first button of the tool bar.

The analyser fields are:

- The List of Services. The list of services defined by the WSDL file.
- The List of Ports. The ports for the selected service.
- The List of Operations. The list of available operations for the selected service.
- The Action URL. This is not editable and it shows the script that serves the operation.

- The SOAP Action. This is not editable and identifies the action performed by the script.
- The Request Editor. It allows you to compose the web service request. When an action is selected, <oXygen/> tries to generate as much content as possible for the call skeleton. Usually you just have to change few values in order for the request to be valid. The tag-insight is available for this editor and is driven by the schema that defines the type of the current message.
- The Attachments List. You can define a list of file's URLs to be attached to the request.
- The Response Area. It presents the message received from the server in response to the Web Service request. It may show also error messages.
- The Errors List. There may be situations in which the WSDL file is respecting the WSDL XML Schema, but it fails to be valid for example in the case of a message that is defined by means of an element that is not found in the types section of the WSDL. In such a case, the errors will be listed here. This list is presented only when there are errors.
- The Send Button. Executes the request. A status dialog is shown when <oXygen/> is connecting to the server.

The testing of a WSDL file is straight-forward, you just have to click on the WSDL analysis button, then select the service, the port and the operation. The editor will generate the skeleton for the request. You can edit the request, eventually attach files to it and send it to the server. Watch the server response in the response area.

Chapter 8. XQuery Support

XQuery Overview

XQuery is the query language for XML. The many benefits of XQuery include:

- XQuery allows you to work in one common model no matter what type of data you're working with: relational, XML, or object data.
- XQuery is ideal for queries that must represent results as XML, to query XML stored inside or outside the database, and to span relational and XML sources.
- XQuery allows you to create many different types of XML representations of the same data.
- XQuery allows you to query both relational sources and XML sources, and create one XML result.

XQuery is currently under development at the W3C.

<oXygen/> XML Editor includes an XQuery editor featured with:

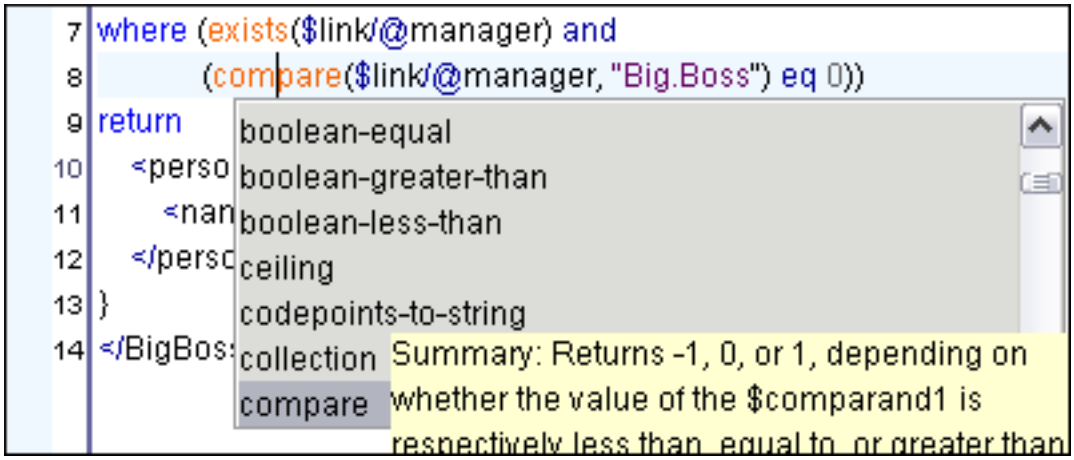
- syntax highlight for XQuery documents
- code insight for XQuery functions, operators and keywords
- XQuery validation and execution
- support for applying your queries on XML documents

To create a new XQuery document you can select File-> New (**Ctrl+N**) and when the New Document dialog appears select XQuery entry.

Syntax Highlight and Content Completion

Once you created the new document <oXygen/> provides syntax highlight for keywords and all known XQuery functions and operators. Also for these there is available a code-insight component that can be activated by pressing Ctrl+Space keys. The functions and operators are presented together with a comment about parameters and functionallity.

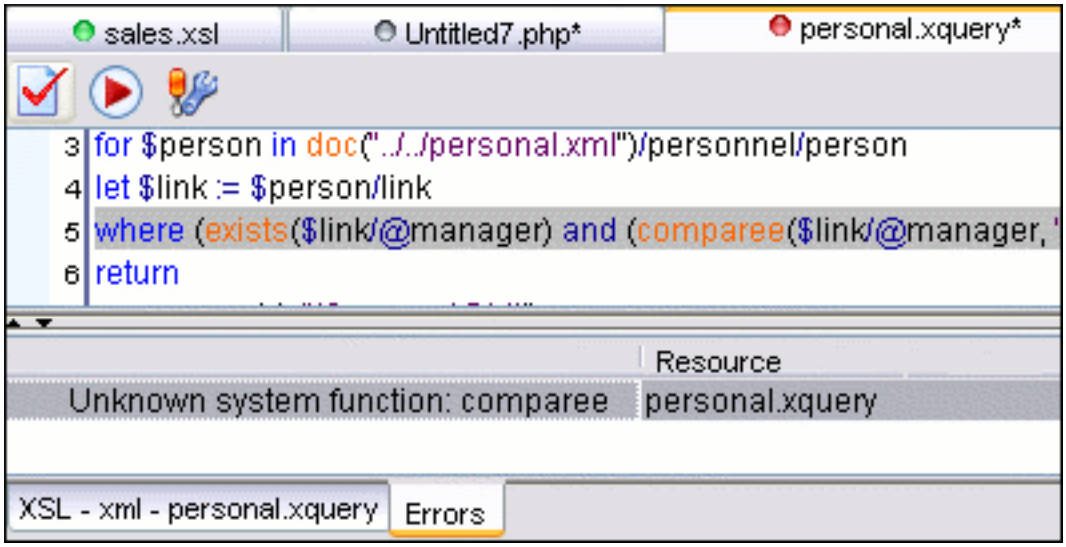
Figure 8.1. XQuery Tag Insight



XQuery Validation

With <oXygen/> you can validate your documents before using them in your transformation scenarios. The validation uses the Saxon 8.1B processor. This is conformant to the XQuery Working Draft http://www.w3.org/TR/xquery/. The processor is used in two cases: validation of the expression and execution. Although the execution implies a validation, it is faster to syntactically check the expression without executing it. The errors that occured in the document are presented in the messages view at the bottom of editor window, with a full description message. As with all error messages, if you click on one entry, the line where the error appeared is highlighted.

Figure 8.2. XQuery Validation



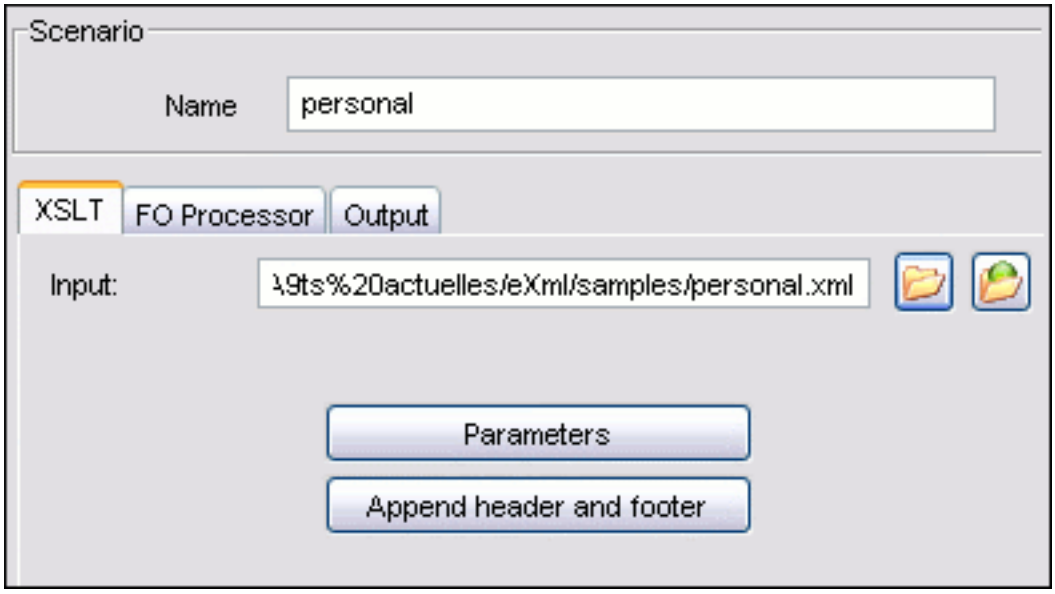
Transforming XML Documents Using XQuery

XQueries are very similar to the XSL stylesheets in the sense they both are capable of transforming an XML input into another format. You can define transformation scenarios that specify the input URL, the preview mode, XML or XHTML. The result can be saved and opened in the associated application. You can even run a FO processor on the output of an XQuery. The transformation scenarios may be shared

between many XQuery files, and are exported at the same time with the XSLT scenarios.

The Transformation Scenario Edit dialog is illustrated below. The transformation performed can be based on the XML document specified in the Input field, or, if this field is empty, the documents reffered from the query expression are used instead.

Figure 8.3. XQuery Transformation

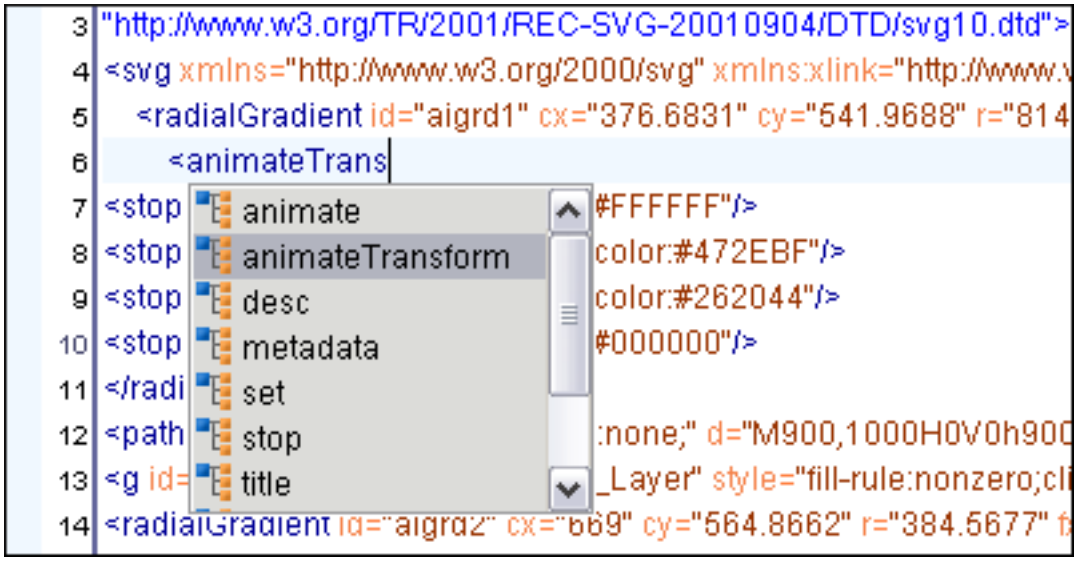


Chapter 9. SVG Editor

SVG is a platform for two-dimensional graphics. It has two parts: an XML-based file format and a programming API for graphical applications. Just to enumerate some of the key features: shapes, text and embedded raster graphics with many painting styles, scripting through languages such as ECMAScript and support for animation.

SVG is a vendor-neutral open standard that has important industry support. Companies like Adobe, Apple, IBM and others have contributed to the W3C specification. Many documentation frameworks, including Docbook have support for SVG by means of defining the graphics directly in the document.

Figure 9.1. SVG Tag Insight



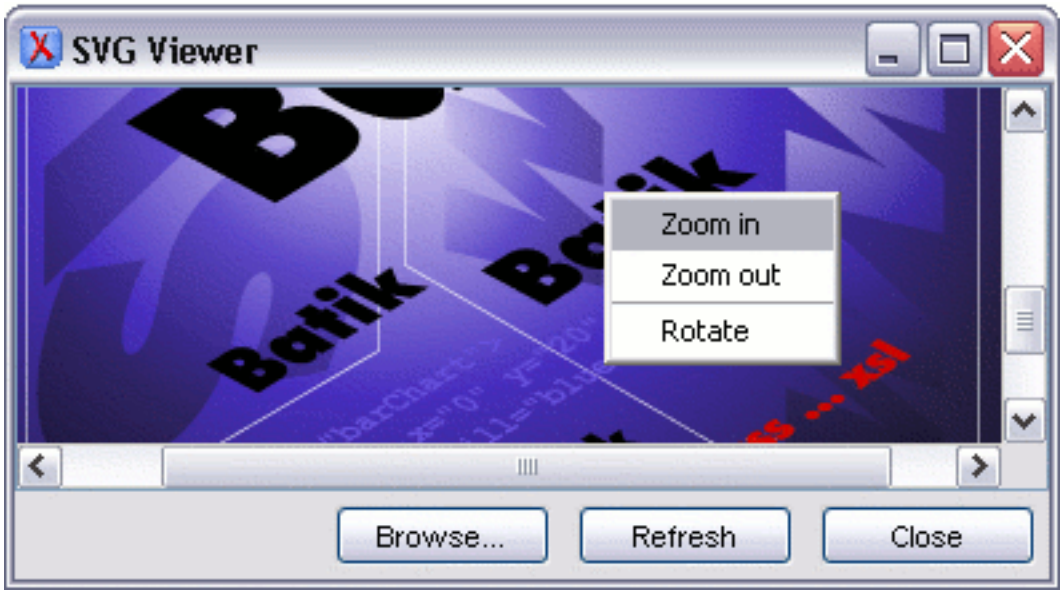
<oXygen/> XML Editor adds SVG support by using the Batik [http://xml.apache.org/batik/] package, an open source project developed by the Apache Software foundation. The SVG DTD is solved by the default XML catalog.

<oXygen/> can render SVG by two means:

The Standalone SVG Viewer.

You may use it to browse and open any SVG file.

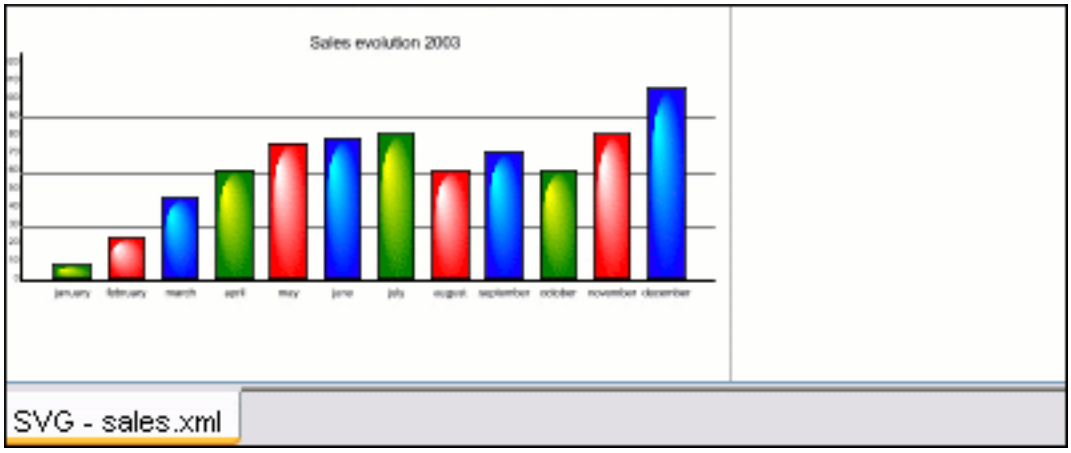
Figure 9.2. SVG Viewer



The Preview Result Pane.

This panel can render the result of an XSL transformation that generates SVG documents.

Figure 9.3. Integrated SVG Viewer



The basic use-case of <oXygen/> consists in the development of the XSL stylesheets capable of producing rich SVG graphics. For example when you have an XML document describing the evolution of a parameter over time and you need to create a graphic from it. You can start with a static SVG, written directly in <oXygen/> or exported from a graphics tool like the Adobe suite. Extract then the parts that are dependent of the data from the XML document and create the XSL templates.

Appendix A. Appendix

Accelerator Shortcut Keys

The Main Editor

File-> New (**Ctrl+N**) : Displays the New dialog from which to select the document file type.

File-> Open (**Ctrl+O**) : Displays the Open dialog used to discover, select and open one or more files.

File-> Save (**Ctrl+S**) : Saves the current document. If the document does not have a file, displays the "Save As" dialog.

File->Save Results (**Ctrl+R**) : Displays the Save Results dialog, used to save the result-list of the, currently in focus, message tab.

File->Open Project (**Ctrl+F2**) : Displays the Open Project dialog used to discover, select and open a project file.

File->Save Project (**Ctrl+F3 (Cmd+G on Mac)**) : Saves the current project. If the project does not have a file, displays the "Save Project As" dialog.

File->Print (**Ctrl+P**) : Displays the Page Setup dialog used to define the page size and orientation properties for printing.

File-> Close (**Ctrl+W**) : Closes only the selected tab. All other tab instances remain.

File->Exit (**Ctrl+Q**) : Terminates the <oXygen/> XML Editor. Session information such as the current Project, open Documents and Option settings is made persistent. When the <oXygen/> editor is reopened, the persistence information returns to the last saved state.

Edit->Undo (**Ctrl+Z**) : Reverses, a maximum of 100, editing actions to return to the preceding state.

Edit->Redo (**Ctrl+Shift+Z**) : Recreates, a maximum of 100, editing actions that where undone by the "Undo" function.

Edit->Cut (**Ctrl+X**) : Removes the current selected node from the document and places it in the clipboard.

Edit->Copy (**Ctrl+C**) : Places a duplicate copy of the current selection in the clipboard.

Edit->Paste (**Ctrl+V**) : Places the current clipboard content into the document at the cursor position.

Edit->Select All (**Ctrl+A**) : Selects the entire body of the current document, including whitespace preceding the first and following the last character.

Edit->Check Spelling (**F4**) : Checks the spelling in your document.

Find->Find/Replace... (**Ctrl+F**) : Displays the Find/Replace dialog, used to define "search for" or "search for and replace" operations on the current document. The replace operation can bind Perl 5-like regexp group variables ($1, $2, etc.) from the find match.

Find->Go to Line (**Ctrl+G (Cmd+L on Mac)**) : Displays the Go to Line dialog used to move the cursor directly to the line number specified.

Find->Search again (**F3**) : Performs another search using the last search configuration.

Tree Editor->Show... (**Ctrl+T**) : Opens the window for editing a document displayed as a structured tree.

Help->Help (**F1**) : Opens the <oXygen/> XML Editor Online Help System.

Document->Validate document (**Ctrl+Shift+V**) : Executes the Validation operation on the current document using a validating parser. Returns an error result-list in the Message panel. Mark-up of current document is checked to conform with the specified DTD rules.

Document->Check document form (**Ctrl+Shift+W**) : Executes the XML Form check operation on the current document using a non-validating parser. Returns an error result-list in the Message panel.

Document->Apply transformation scenario (**Ctrl+Shift+T**) : Executes the transformation process using the configuration properties defined in the Configure Transformation dialog.

Document->Configure transformation scenario (**Ctrl+Shift+C**) : Displays the Configure Transformation dialog, used to define properties for conversion of documents to multiple output targets. Also enables saving of "Scenarios". Each scenario, can store a unique configuration ready to be used in the future.

Document->Format and Indent (**Ctrl+Shift+P**) : Also referred to as "Pretty Print", "Format and Indent" performs layout functions to make mark-up easier to read on screen and in print output.

Document->Learn Structure (**Ctrl+Shift+L**) : Reads the mark-up structure of the current document so that it can be saved as a template using the Save Structure option.

Document->Save Structure (**Ctrl+Shift+S**) : Displays the Save Structure dialog, used to name and create DTD documents learnt by the "Learn Structure" function.

Document-> Find All (**Ctrl+Shift+F**) : Finds all occurencies of selected word in current file.

The Tree View Editor

File-> New (**Ctrl+N**) : Creates a new empty document and displays it in the Tree View Editor.

File-> Open (**Ctrl+O**) : Displays the Open dialog used to discover, select and open a file to be edited.

File-> Save (**Ctrl+S**) : Saves the current document. If the document does not have a file, displays the "Save As" dialog.

File-> Close (**Ctrl+W**) : Closes the Tree View Editor.

Edit->Copy (**Ctrl+C**) : Places a duplicate copy of the current node in the clipboard.

Edit->Cut (**Ctrl+X**) : Removes the current selected node from the document and places it in the clipboard.

Edit->Paste (**Ctrl+V**) : Places the node from clipboard as a child of the selected node.

Edit->Delete (**Delete**) : Delete the selected node from the document.

Edit->Start Editing (**F5**) : Starts editing the selected node from the document.

Edit->End Editing (**F6**) : Ends editing the selected node.

Edit->Undo (**Ctrl+Z**) : Reverses, a maximum of 100, editing actions to return to the preceding state.

Edit->Redo (**Ctrl+Shift+Z**) : Recreates, a maximum of 100, editing actions that where undone by the "Undo" function.

Insert-> Insert (**F9**) : Insert a new node of the same type like the selected one as its sibling.

Move->Move Up (**Ctrl+Up**) : Move up the selected node with one position.

Move->Move Down (**Ctrl+Down**) : Move down the selected node with one position.

Unicode Character Encoding

The table below provides a matrix from which to match Unicode names with the names shown by the Java Encoder when it cannot identify encoding.

Table A.1. Unicode to Java Name Matrix

Common Name	Name in XML files	Name Type	Java Encoder Name
8 bit Unicode	UTF-8	IANA	UTF8
16 bit Unicode	UTF-16	IANA	Unicode
16 bit Unicode little endian	UTF-16LE	IANA	UnicodeLittle
16 bit Unicode big endian	UTF-16BE	IANA	UnicodeBig
ISO Latin 1	ISO-8859-1	MIME	ISO-8859-1
ISO Latin 2	ISO-8859-2	MIME	ISO-8859-2
ISO Latin 3	ISO-8859-3	MIME	ISO-8859-3
ISO Latin 4	ISO-8859-4	MIME	ISO-8859-4
ISO Latin Cyrillic	ISO-8859-5	MIME	ISO-8859-5
ISO Latin Arabic	ISO-8859-6	MIME	ISO-8859-6
ISO Latin Greek	ISO-8859-7	MIME	ISO-8859-7
ISO Latin Hebrew	ISO-8859-8	MIME	ISO-8859-8
ISO Latin 5	ISO-8859-9	MIME	ISO-8859-9
EBCDIC: US	ebcdic-cp-us	IANA	cp037
EBCDIC: Canada	ebcdic-cp-ca	IANA	cp037
EBCDIC: Netherlands	ebcdic-cp-nl	IANA	cp037
EBCDIC: Denmark	ebcdic-cp-dk	IANA	cp277
EBCDIC: Norway	ebcdic-cp-no	IANA	cp277
EBCDIC: Finland	ebcdic-cp-fi	IANA	cp278
EBCDIC: Sweden	ebcdic-cp-se	IANA	cp278
EBCDIC: Italy	ebcdic-cp-it	IANA	cp280
EBCDIC: Spain, Latin America	ebcdic-cp-es	IANA	cp284
EBCDIC: Great Britain	ebcdic-cp-gb	IANA	cp285
EBCDIC: France	ebcdic-cp-fr	IANA	cp297
EBCDIC: Arabic	ebcdic-cp-ar1	IANA	cp420
EBCDIC: Hebrew	ebcdic-cp-he	IANA	cp424
EBCDIC: Switzerland	ebcdic-cp-ch	IANA	cp500
EBCDIC: Roece	ebcdic-cp-roece	IANA	cp870
EBCDIC: Yugoslavia	ebcdic-cp-yu	IANA	cp870

Common Name	Name in XML files	Name Type	Java Encoder Name
EBCDIC: Iceland	ebcdic-cp-is	IANA	cp871
EBCDIC: Urdu	ebcdic-cp-ar2	IANA	cp918
Chinese for PRC, mixed 1/2 byte	gb2312	MIME	GB2312
Extended Unix Code, packed for Japanese	euc-jp	MIME	eucjis
Japanese: iso-2022-jp	iso-2020-jp	MIME	JIS
Japanese: Shift JIS	Shift_JIS	MIME	SJIS
Chinese: Big5	Big5	MIME	Big5
Extended Unix Code, packed for Korean	euc-kr	MIME	iso2022kr
Cyrillic	koi8-r	MIME	koi8-r

References

Organization for the Advancement of Structured Information Standards (OASIS) [http://www.oasis.org/]

OASIS is a not-for-profit, global consortium that drives the development, convergence and adoption of e-business standards. Members themselves set the OASIS technical agenda, using a lightweight, open process expressly designed to promote industry consensus and unite disparate efforts. OASIS produces worldwide standards for security, Web services, XML conformance, business transactions, electronic publishing, topic maps and interoperability within and between marketplaces.

World Wide Web Consortium (W3C) XML Specifications [http://www.w3.org/]

The World Wide Web Consortium (W3C) develops inter operable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. W3C is a forum for information, commerce, communication, and collective understanding.

DocBook [http://www.docbook.org/]

DocBook is an XML/SGML vocabulary particularly well suited to books and papers about computer hardware and software (though it is by no means limited to these applications). DocBook is officially available as a Document Type Definition (DTD) for both XML and SGML and enjoys the support of a broad user base throughout 100's of organizations around the world.

IBM Developer Works XML Zone [http://www-106.ibm.com/developerworks/xml/]

A gateway to all things XML and home of the Darwin Information Typing Architecture (DITA) [http://www-106.ibm.com/developerworks/xml/library/x-dita1/] is an XML-based, end-to-end architecture for authoring, producing, and delivering technical information. This architecture consists of a set of design principles for creating "information-typed" modules at a topic level and for using that content in delivery modes such as online help and product support portals on the Web.

The Unicode Consortium [http://www.unicode.org]

The Unicode Consortium is responsible for defining the behavior and relationships between Unicode characters, and providing technical information to implementers. The Consortium cooperates with ISO in refining the specification and expanding the character set. It has liaison status "C" with ISO/IEC/JTC 1/SC2/WG2, which is responsible for ISO/IEC 10646.