

# QXmlEdit

## User Manual

*Editing and exploring XML data  
with QXmlEdit.*

**Version 0.9.15**

# Contents

<b>2 Contents</b>	
<b>4 Welcome to QXmIEdit</b>	
4 View and Navigate XML Data in Unusual Modes	
4 Information About QXmIEdit	
4 Before You Begin	
4 What You Will Learn	
5 Finding Out More	
5 What's New in This Version	
<b>6 Overview of QXmIEdit User Interface</b>	
<b>7 Operation with XML</b>	
<b>8 Startup</b>	
<b>10 Main Functionality</b>	
<b>10 Working with XML Files</b>	
10 Opening Files	
10 Opening Files in New or Same Window	
11 Using only One Window or One Window per File	
11 Saving files	
11 Creating a File From Clipboard Contents	
11 Creating specialized files	
12 Reloading last saved version	
12 Exporting the XML	
12 Avoiding modifications	
12 Loading the Last Edited Files	
12 Working With Preferred Directories	
13 Viewing Elements and Attributes	
16 Filtering Attributes	
<b>18 Working with name spaces and prefixes</b>	
<b>18 Prefix operations</b>	
18 <i>Assigning a prefix</i>	
18 <i>Removing a prefix</i>	
19 <i>Replacing a prefix</i>	
<b>19 Namespace operations</b>	
20 <i>Assigning a namespace</i>	
20 <i>Removing a namespace</i>	
21 <i>Replacing a namespace</i>	
21 <i>Normalizing a namespace</i>	
<b>23 Choosing an Editing Mode</b>	
25 Using XSLT Edit Mode	
25 <i>Specific display style</i>	
25 <i>Specific edit panels</i>	
26 <i>A specific context menu</i>	
27 <i>Navigate templates and functions</i>	
28 Using SCXML Edit Mode	
28 <i>Specific display style</i>	
28 <i>Specific edit panels</i>	
28 <i>Navigate states</i>	
<b>30 XML Schema management</b>	
31 Inserting XML Schema References	
31 Using the XSD Edit Mode	
31 <i>Editing or inserting a type, attribute or element</i>	
33 Edit an annotation	
33 <i>Single annotation panel</i>	
35 <i>Annotation list edit panel</i>	
36 Editing Enumeration or Facets	
36 <i>Edit facet panel</i>	
36 This panel edits a single facet item.	
37 Editing XML Schema Attributes	
37 <i>Inserting the Schema Instance "nil" Attribute</i>	
37 <i>Removing the Schema Instance "nil" Attribute</i>	
37 <i>Inserting the Schema Instance "type" Attribute</i>	
37 <i>Removing the Schema Instance "type" Attribute</i>	
<b>38 Help and information</b>	
38 Show keyboard shortcuts	
<b>39 Tasks</b>	
39 Adjust Display Detail	
<b>40 Preferences</b>	
41 <i>General</i>	
42 <i>Style</i>	
43 <i>Editor colors</i>	
44 <i>Validation</i>	
45 <i>Sessions</i>	
46 <i>Edit Modes</i>	
47 <i>XML</i>	
<b>49 Customization</b>	
49 Choosing a Display Style	
50 Editing Elements	
50 Navigating Elements	
51 Editing Elements and Attributes	
53 <i>Searching text</i>	
53 Deleting siblings	
53 Copying Selected Text Into the Clipboard	
54 Namespaces	
56 Using the Bookmarks	
58 Viewing and editing info and DTD	
59 Working with Base64 Coded Content	
59 Working With Inner XML Content	
59 Inserting an XML prolog on document creation	
59 Formatting a document	
60 <i>Formatting Presets</i>	
60 Validating an XML Document	
60 Counting Children Elements and Measuring Their Size	
61 Finding Text	
62 <i>Search results</i>	
63 XPath syntax	

- [64 Search results](#)
- [64 Browsing the search results](#)
- [65 Replacing text](#)
  - [65 Replace panel](#)
- [65 Comparing Files](#)
- [68 Anonymizing data](#)
  - [68 Data and business rules](#)
  - [69 Options](#)
  - [71 Configuring the anonymization](#)
  - [71 Format of the CSV for importing and exporting the exceptions](#)
- [73 Anonymizing a file](#)
- [74 Filling data](#)
- [75 Cloning Elements](#)
- [75 This picture shows the result of the clone operation with the insert of an index using the "id" attribute:](#)
- [76 Working with Snippets](#)
- [78 Executing XSL style sheet](#)**
  - [78 XSLT Execution panel](#)
- [80 Metadata](#)**
  - [80 Introduction](#)**
  - [80 Metadata types](#)**
    - [80 Updatable Metadata](#)
    - [80 Static metadata](#)
  - [80 Metadata implementation](#)**
  - [80 Operation](#)**
    - [80 Creating a document](#)
    - [80 Reading a document](#)
    - [81 Saving a document](#)
    - [81 Editing metadata](#)
  - [81 Metadata summary](#)**
    - [81 Updatable Metadata](#)
    - [81 Static Metadata](#)
  - [81 Editing metadata](#)**
- [83 Working With XML Schema Files \(XSD\)](#)**
  - [84 Visualization Types](#)
  - [84 Changing the background color and pattern.](#)**
  - [85 Exporting XSD Graphical view](#)**
  - [86 Working With Big XML Files](#)**
  - [86 Searching in Files and counting elements](#)**
  - [88 Split a XML file](#)**
    - [88 Decide if you want to extract information or browse it](#)
    - [88 Choose the file and how to fragment it](#)
    - [89 Limit the extracted fragments number](#)
    - [89 Decide the output folder and the naming](#)
    - [90 Advanced Options](#)

- [90 Go and examine data](#)
- [92 Flex Code Generation From Balsamiq Source](#)**
- [93 Life With Sessions](#)**
  - [94 The session user interface](#)
  - [94 Sessions Management Dialog](#)
  - [95 Session Properties](#)
  - [96 Managing sessions data](#)
  - [97 Sessions configuration](#)
- [97 Viewing data](#)**
  - [98 User Interface](#)
  - [99 3D Visualization](#)
  - [99 Measurement types](#)
    - [99 Size](#)
    - [99 Payload size](#)
    - [100 Attributes count](#)
    - [100 Children Elements count](#)
    - [100 Structure](#)
  - [100 Commands available in the contextual menu of the map](#)
- [100 Visualize Relationship](#)**
  - [102 Attributes summary](#)
- [102 Testing XML Files](#)**
- [102 Base 64 Explorer](#)**
- [103 Encoding tools](#)**
  - [103 Explore how a string will be coded in a specific encoding](#)
  - [104 Explore how binary values are recognized by different encoding](#)
  - [105 Explore how a string will be translated when written in some encoding and read in another one](#)
- [106 Get information about a character](#)**
- [107 Code Pages explorer](#)**
- [108 Visually Comparing XML Schema files](#)**
- [109 Binary files viewer](#)**
- [110 Command Line arguments](#)**
- [111 Appendix](#)**
  - [111 Note on "Single Application" Mode](#)
  - [111 Style file format](#)
    - [111 Root Tag:](#)
    - [111 Element "styles"](#)
    - [111 Element "default"](#)
    - [111 Element "style"](#)
    - [112 Element "keywords"](#)
    - [112 Element "keyword"](#)
    - [112 Element "ruleSet"](#)
    - [112 Element "rule" \(inner\)](#)
    - [113 Element "ids"](#)
    - [113 Element "id"](#)
    - [114 Installation of new styles](#)

## Welcome to QXmlEdit

### View and Navigate XML Data in Unusual Modes

QXmlEdit offers you the possibility to view, navigate and edit XML data as few other editors. It is like a Swiss army knife. It is multi platform and, more, it is *Libre Software*; that means that it gives to the user the freedom to adapt it to its needs and it comes with complete source code.

### Information About QXmlEdit

QXmlEdit is an XML editor and its main features are:

- Hierarchical customizable view of XML elements.
- Fast XML hierarchy navigation.
- Handling of base 64 coded text and XML inserted as data.
- Powerful search.
- XML snippets.
- XML compare utility.
- XML Schema viewer.
- Visual compare of XML schema.
- Flex code generation from Balsamiq source.
- Session handling.
- XSL specialized mode.
- Split of big XML files.
- Data anonymizer.

### Before You Begin

QXmlEdit is an XML editor. To use it with proficiency you need to be accustomed with XML. It is a technical tool oriented toward software developers.

### What You Will Learn

Following this manual you will learn to:

- Examine XML data in details.
- Applying different views to XML data to examining them under different points of view.
- Customize QXmlEdit to ease your work.
- Navigate and inspect XML data.

## Finding Out More

On the QXmlEdit Internet site you will find the up to date documentation and tutorials. The site address is: <http://qxmledit.org>. The link for the source code is: <https://github.com/lbellonda/qxmledit>.

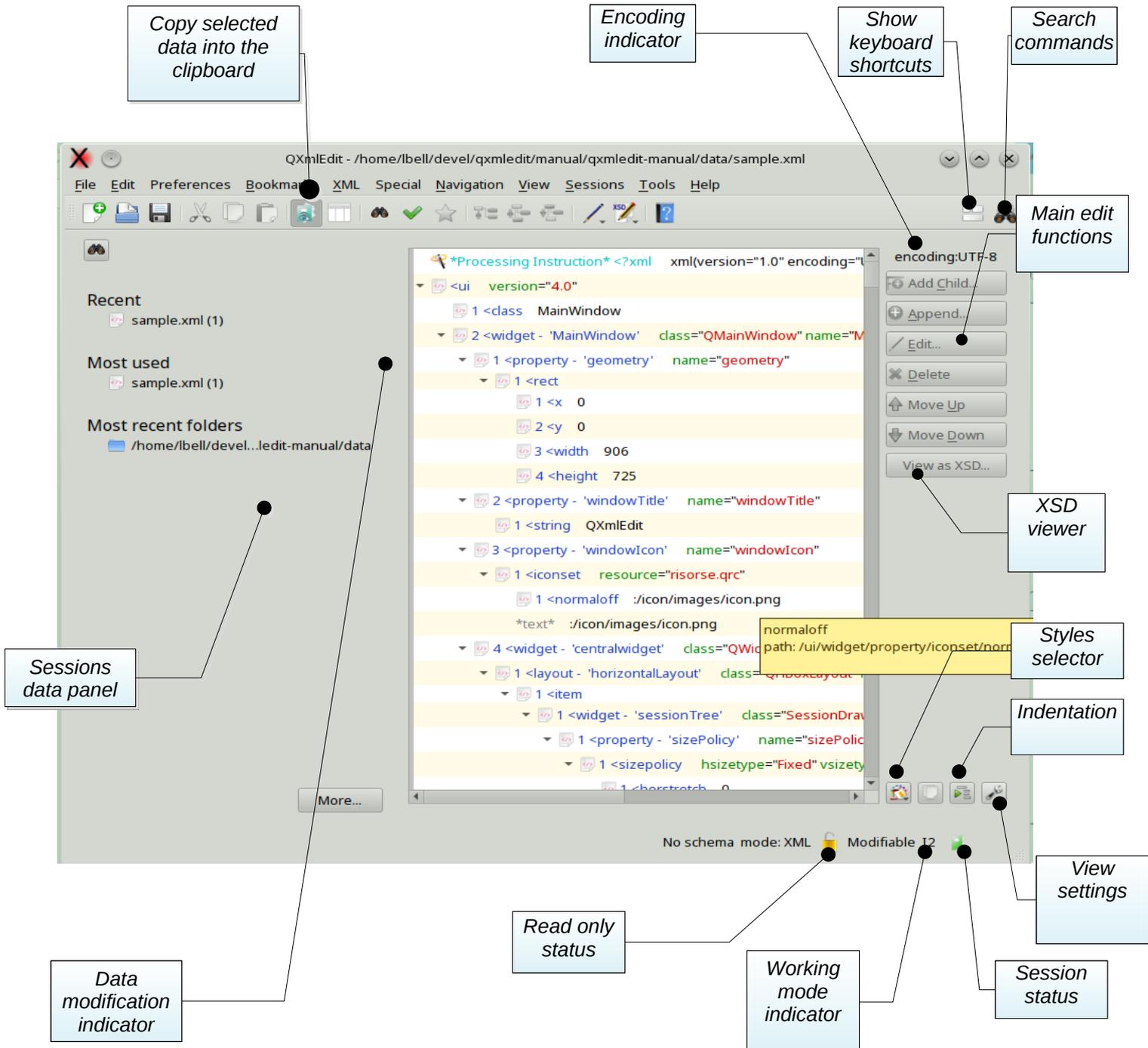
To get help while using the program you can use the menu **Help > Help on QXmlEdit**.

## What's New in This Version

Find in text panel and the possibility of using the text panel to edit attributes, plus some fixes.

## Overview of QXmlEdit User Interface

The user interface is very simple and consists of a main panel where all the functions of the program can be controlled.



## Operation with XML

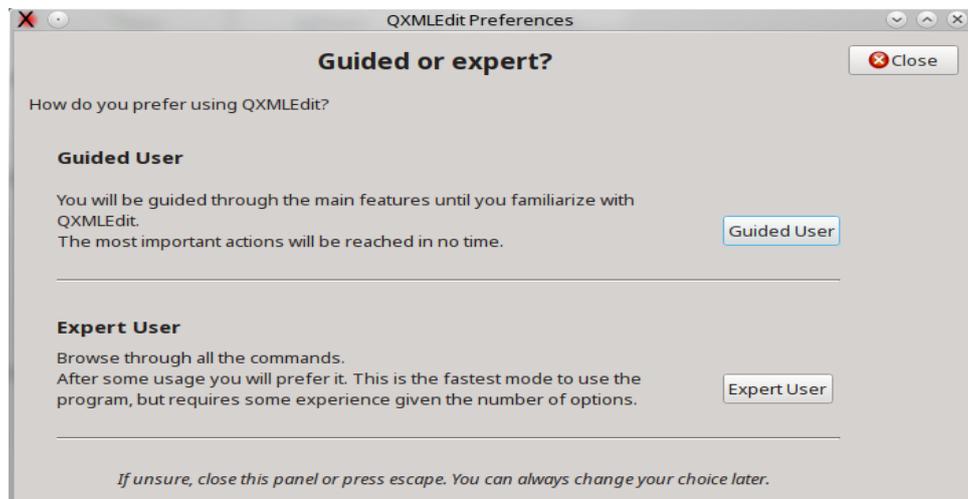
This is a small abstract of the common operations grouped by purpose:

- Navigation:
  - Go to parent element (menu Navigation, shortcut F12).
  - Go to next sibling (Navigation, F10).
  - Go to previous sibling (Navigation, F9).
- Data modification:
  - Insert or append an element.
  - Insert or append a comment.
  - Insert or append a processing instruction.
  - Change the position of a node.
  - Paste a node as child or sibling.
  - Copy or paste a set of attributes.
  - Operation on containers:
    - Insert a parent
    - Insert a child container that enclose all the current children
    - Remove the parent of a node.
  - Transform a node into a comment and vice versa.
  - Insert prolog and namespace declarations.
  - Edit base 64 coded text blocks.
  - Edit base 64 coded attributes.
  - Transform files into base 64 text blocks.
- View Data:
  - Hide all the siblings of a node.
  - Hide all the children of all the sibling of a node.
  - Filter the visible attributes.
  - Hide all the elements that have no children.
  - Show the children count and the size of a node.
  - Close all the siblings of a node.

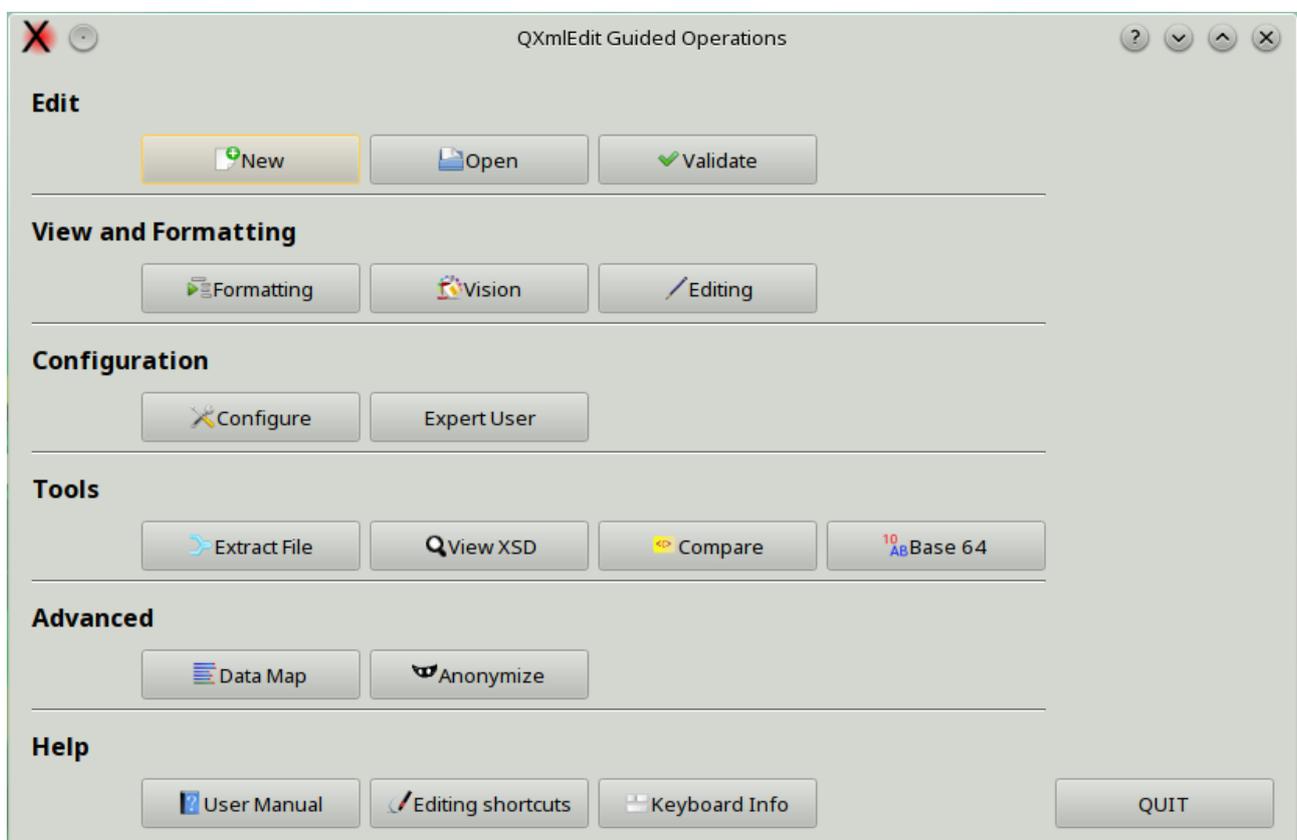
For a complete description of each operation go to the specific section of this document. Here you will find a small reminder of the commands.

## Startup

The first time QXMLEdit is started, it asks the user if a simplified user interface should be activated. Through a panel it is possible to access the main features but there are no restrictions about what QXMLEdit can perform. At any time, using the **Preferences** menu, the choice can be reverted.



If the choice is "guided user", the following panel is open:



From above panel the following operations can be started:

<b>Item</b>	<b>Description</b>
<b>New</b>	Open a new editor.
<b>Open</b>	Load an XML file.
<b>Validate</b>	Validate XML using XSD.
<b>Formatting</b>	Formatting and indentation options.
<b>Vision</b>	View Options.
<b>Editing</b>	Editing options.
<b>Configure</b>	Open the main preferences panel.
<b>Expert User</b>	Switch to "Expert User" mode.
<b>Extract File</b>	Extract fragments from an XML.
<b>View XSD</b>	View an XSD file as a diagram.
<b>Compare</b>	Compare XML files.
<b>Base 64</b>	Open the Base64 operations panel.
<b>View Data Map</b>	View XML file statistics.
<b>Anonymize</b>	Anonymize a file using previously created profiles.
<b>User Manual</b>	Open the user manual.
<b>Editing Shortcuts</b>	Show the editing shortcuts.
<b>Keyboard Info</b>	Show the panel that list keyboard shortcuts.

## Main Functionality

Menus are divided in main functional groups; the most used commands are kept visible in main dialog as a button on the right.

The menus are described here:

Menu	Functionality
File	XML document creation and saving
Edit	clipboard handling; find command and configuration options
Tasks	common tasks, grouping other menu commands
Bookmarks	all about bookmarks
XML	XML tree manipulation (insertion, deletion, etc)
Special	special XML operations
Navigation	XML tree navigation
View	view commands and options
Sessions	Session related commands
Tools	plugins and tools
Help	help

## Working with XML Files

The QXmlEdit goal is to view and edit small XML files. It is not intended to handle data of hundred of megabytes, since it works completely in memory.

## Opening Files

1. Open files using the **File > Open menu**.
2. In the **Open Files** dialog choose the file type you are looking for.

## Opening Files in New or Same Window

This feature lets you override the settings by opening a file in a new window, if open new files in same window is selected as a global option. If the configured behavior is to open files using new windows, the option lets you reuse the current editor window.

- Use the menu **File > Open in New Window** if behavior is to open files reusing same window.
- Use the menu **File > Open in Same Window** if behavior is to open files using new windows.

## Using only One Window or One Window per File

This setting can be configured in the preferences dialog.

- Menu **Edit > Configure**
- Select **General** tab
- Check “**Open in new window**” option.

## Saving files

You can save the file using this menu commands:

To save the file with the same name:

- Save the file using the **File > Save** menu.

To save the file with another name

- Save the file using the **File > Save As...** menu.

To save a copy of the file with another name, but continue to work in the original file:

- Open files using the **File > Save a Copy As...** menu.

**Note:** using this last option you make a copy of the file as it is in its current state. The original file is not modified and the editor retains the current state.

## Creating a File From Clipboard Contents

Sometimes you can create a new file by pasting an XML text from the clipboard. For example you can cut a block from a database client or a PDF or HTML page.

Use the menu **File > New From Clipboard**.

## Creating specialized files

Using the **File > New Specialized** item, you can create new documents of a particular kind.

- XSL-FO
- XML Schema
- XSLT
- Maven POM.xml
- SCXML

The documents are created with the predefined namespaces and declarations.

Creating Documents from Snippets

Using the **File > New Specialized > New from Snippet** item, you can create new documents from a previously saved snippet.

## Reloading last saved version

Using the menu **File > Reload the last saved version**. The modifications to the current file will be lost.

## Exporting the XML

The following export methods are available from the **File > Export** menu:

- Export Element to File
- Copy all to the clipboard
- Export as Java/JavaScript
- Export as C/C++

## Avoiding modifications

The document can be put in read only mode using the menu **Edit > Read Only**

## Loading the Last Edited Files

Using the menu **File > Recent Files** you can load the last edited files.

## Working With Preferred Directories

Usually, working with files, there are some directories that are accessed often. QXmlEdit gives you the possibility to remember these places and to quick recall them when opening files.

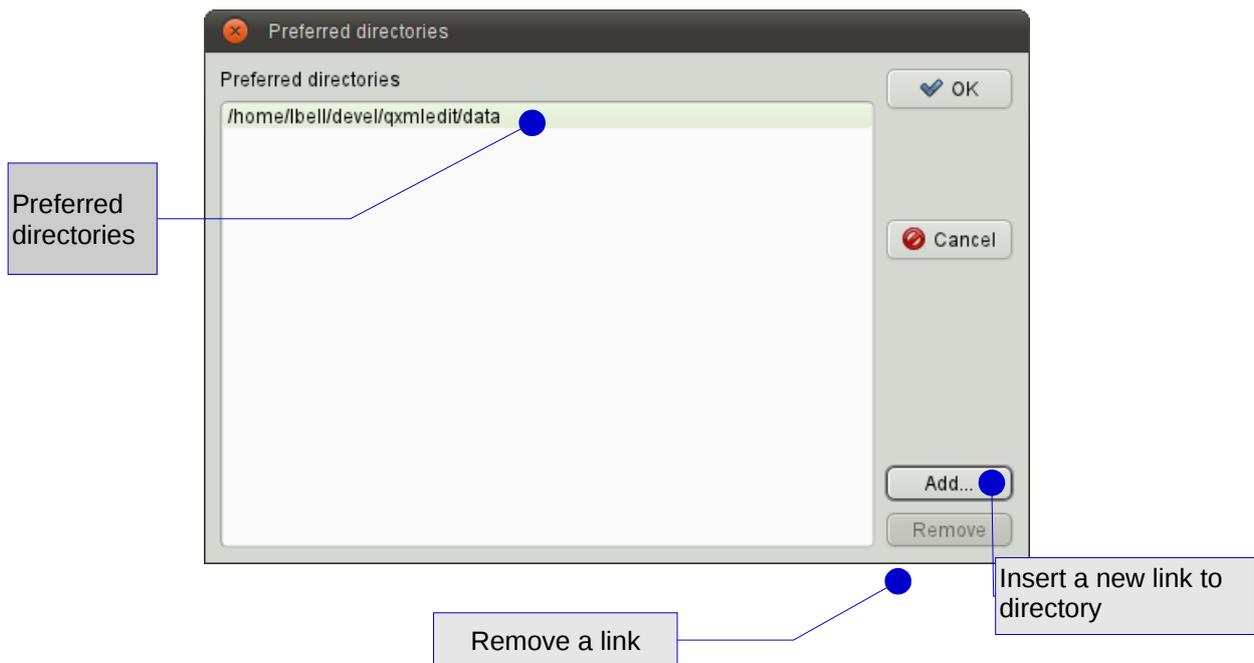
To add the directory of the current open file to the preferred ones:

- Use the menu **File > Preferred Directories > Add Current Directory to Preferred Ones**

To edit the directory list in a separate window:

- Use the menu **File > Preferred Directories > Edit Preferred Directories...**

A dialog will open



To add a new link to a preferred directory

- Push the **Add...** button.

To remove a link to a directory

- Push the **Remove** button.

To confirm the directory list press **OK**.

To open a file into a preferred directory:

Use the menu **File > Preferred Directories**, then select the directory.

## Viewing Elements and Attributes

After having loaded XML data, there are many modes to view elements and attributes. The main view uses a tree representation, mapped on XML data, however you can highlight some data aspect:

To view one element attribute on a separate line (very readable if you are looking for attribute values):

- Select the menu **View > Show One Attribute per Line**

item	
layout - 'verticalLayout'	class="QVBoxLayout" name="verticalLayout"
item	
layout - 'formLayout'	class="QFormLayout" name="formLayout"
property - 'fieldGrowthPolicy'	name="fieldGrowthPolicy"
enum	
item	row="0" column="0"
widget - 'label'	class="QLabel" name="label"
property - 'text'	name="text"
string	
item	row="0" column="1"
widget - 'editTag'	class="QLineEdit" name="editTag"
item	row="1" column="0"
widget - 'label_2'	class="QLabel" name="label_2"
property - 'text'	name="text"
string	

To show the ordinal position of each child relative to their parent:

1 - layout - 'verticalLayout'	class="QVBoxLayout" name="verticalLayout"
1 - item	
1 - layout - 'formLayout'	class="QFormLayout" name="formLayout"
1 - property - 'fieldGrowthPolicy'	name="fieldGrowthPolicy"
1 - enum	
2 - item	row="0" column="0"
1 - widget - 'label'	class="QLabel" name="label"
1 - property - 'text'	name="text"
1 - string	
3 - item	row="0" column="1"
1 - widget - 'editTag'	class="QLineEdit" name="editTag"
4 - item	row="1" column="0"
1 - widget - 'label_2'	class="QLabel" name="label_2"
1 - property - 'text'	name="text"
1 - string	
	row="1"

- Select the menu **View > Show Child Index**

To show a more compact view:

- select the menu **View > Compact View**



To view only the structural components of the XML (the elements that have other elements as children):

- Select the menu **View > Hide All the Leaf Children**

To undo the effect of the previous command:

- Select the menu **View > Show All the Leaf Children**.

To show or hide on the basis of individual node, first select the node, then, to hide:

- Select the menu **View > Hide Leaf Children**.

To show:

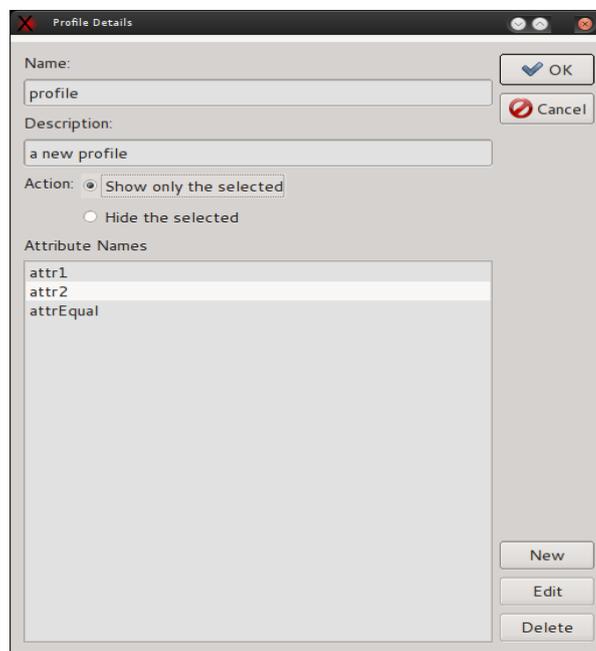
- Select the menu **View > Show Leaf Children**.

## Filtering Attributes

It is possible to filter the display of the attributes. A profile can show only a set of attributes or hide a given list of attributes. When an element has some hidden attributes, a filter icon appears.

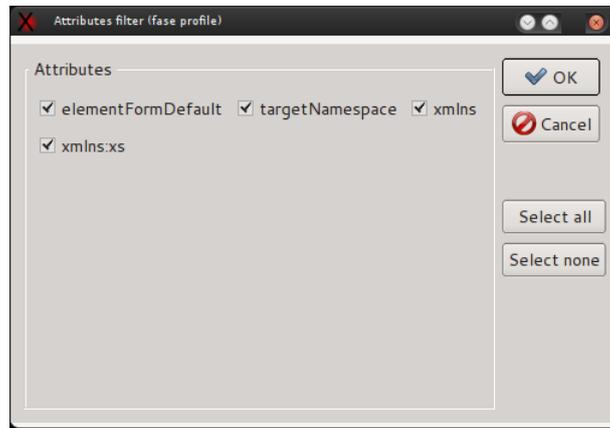
It is possible to manage the attributes that can be shown or hidden in each element using the menu **View > Filter attributes > Manage Profiles...**

In the management section it is possible to create delete or apply profiles.



To apply a temporary profile, based on the current element:

1. Select an element.
2. Use the menu **View > Filter attributes > Apply a Temporary Profile**.



It is possible to transform the temporary profile into a regular one using the menu Use the menu **View > Filter attributes > Save the current Profile.**

## Working with name spaces and prefixes

Namespaces and prefixes commands can be accessed from the XML > Namespaces menu item. Both offer prefix manipulation, but only namespaces commands are aware of namespace concepts. Prefix related items do not know about namespaces at all, they simply manipulate prefixes.

### Prefix operations

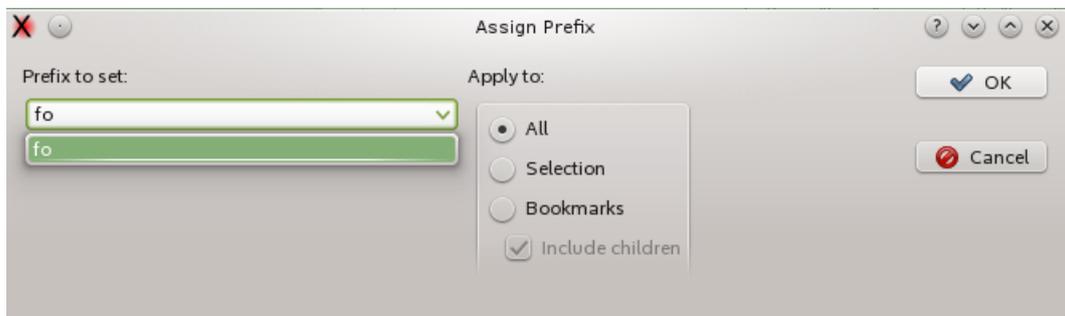
The operations that can be performed on prefixes are:

- Assign a prefix.
- Remove a prefix.
- Replace a prefix.

### Assigning a prefix

This operation sets the same prefix to all the elements on which it is applied. No considerations about namespace declarations are made.

Panel:

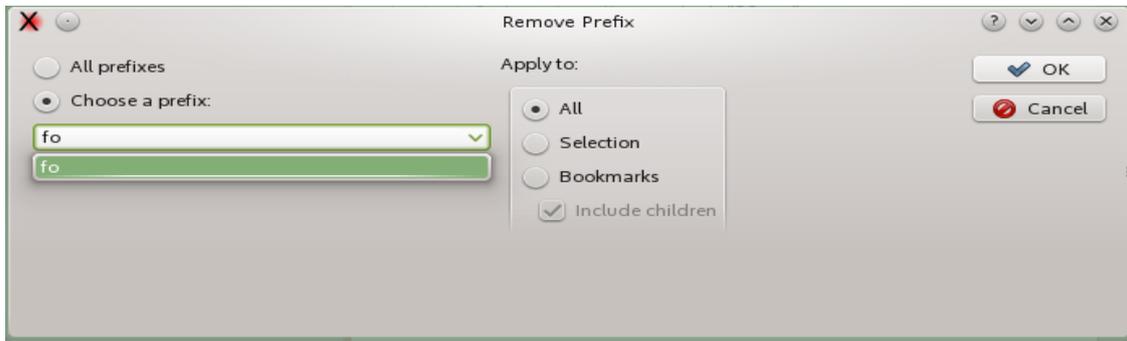


Items:

Item	Description
Prefix to set	The prefix to apply
Apply to	Elements involved.

### Removing a prefix

The remove option removes the selected prefix from all the items, only from the selected item or from the selected item and all its descendants, from tags and attributes. If there is a selection, the prefixes found recursively in the selection are shown in the prefixes combo. No attention is paid to the namespaces in this operation.

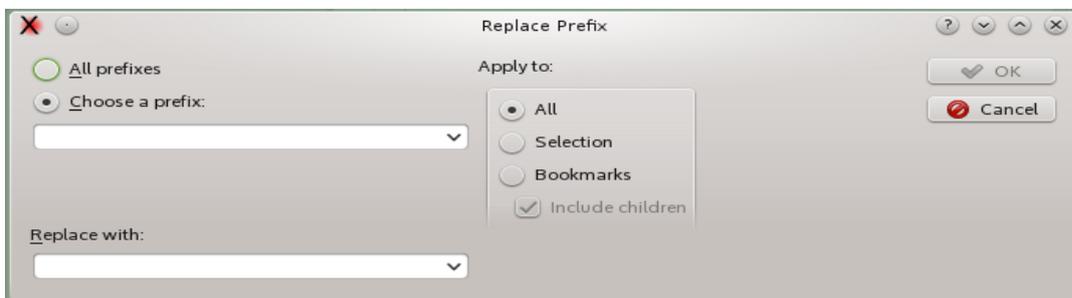


Items:

Item	Description
All prefixes	Choose to remove all the prefixes.
Choose a prefix	Removes a single prefix.
Apply to	Elements involved.

### Replacing a prefix

This operation replaces one or all the prefixes with another one.



Items:

Item	Description
All prefixes	Choose to replace all the prefixes.
Choose a prefix	Replace a single prefix.
Replace with	The prefix that will be inserted.
Apply to	Elements involved.

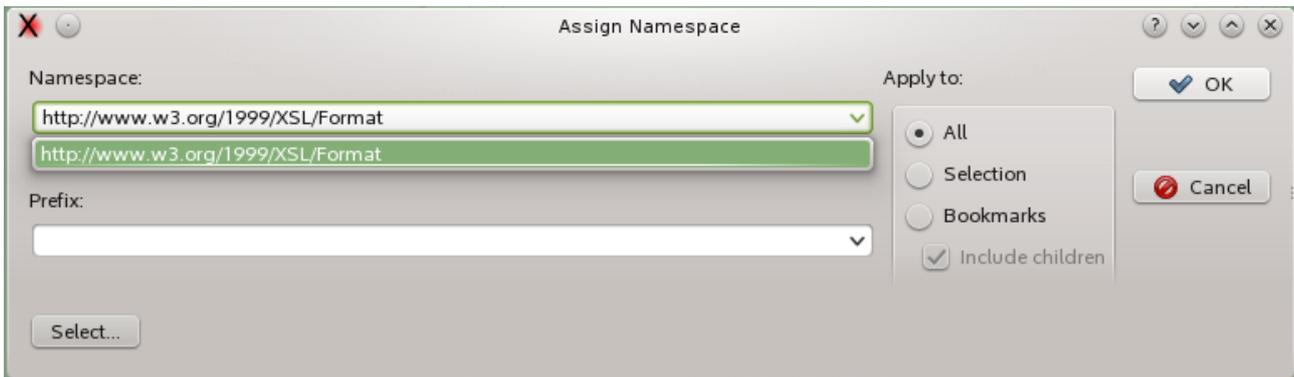
### Namespace operations

The operations that can be performed on namespaces are:

- Assign a namespace.
- Remove a namespace.
- Replace a namespace.
- Normalize a namespace.

## Assigning a namespace

This operation assigns a namespace to a tree or to all the document. The selected namespace will be declared if not existent.



Items:

Item	Description
Namespace	The namespace to set.
Prefix	The prefix to set.
Apply to	Elements involved.

## Removing a namespace

Remove one or all the namespaces. The declarations of the removed namespace can be optionally removed.

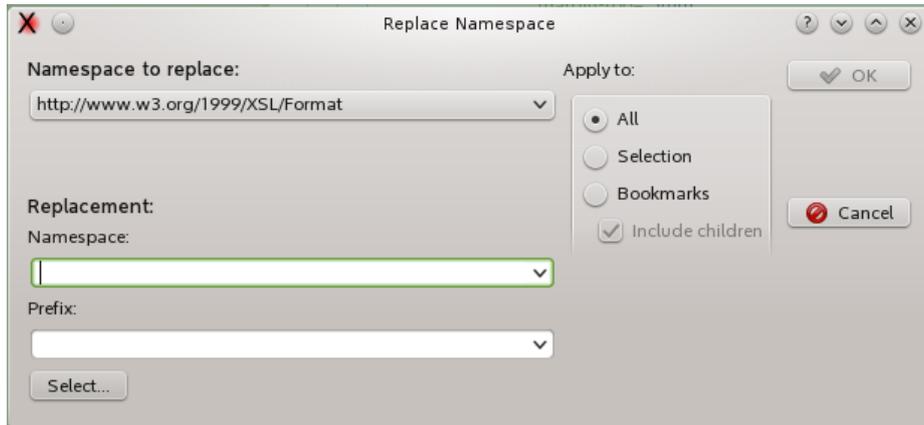


Items:

Item	Description
All namespaces	If chosen, all the namespaces will be removed.
Choose a namespace	Select only one namespace to remove.
Remove declarations	With this option even the namespace declaration will be removed.
Apply to	Elements involved.

## Replacing a namespace

This operation replaces a selected namespace with another one. All the namespace elements will be translated, even if declared with different prefixes.



Items:

Item	Description
Namespace to replace	The namespace to be replaced.
Namespace	The namespace that will be assigned.
Prefix	Prefix of the new namespace.
Select	Recalls the namespace selection panel.

## Normalizing a namespace

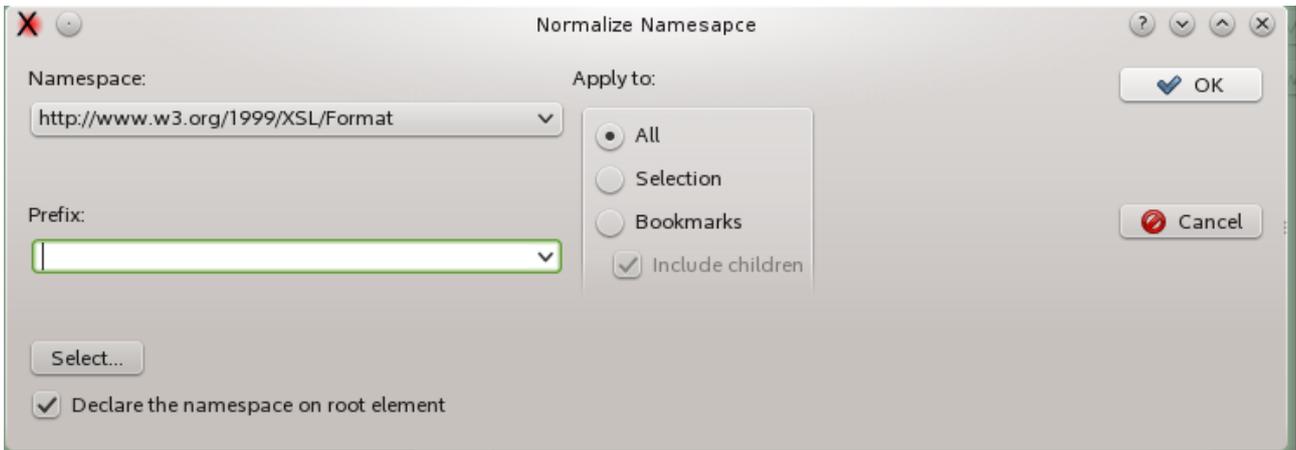
This operation sets the same prefix on all the elements and attributes belonging to a namespace. The prefixes of the namespace other than the selected one are lost and their declarations removed.

The normalization operation assumes that the XML is well formed and valid, so the chosen namespace must be legally declared. This said, it will not be declared anew on the children if the option “Declare namespace on root” is selected.

This option removes all the declaration from the inner of the selection tree and forces a declaration on root of the selection tree. If the apply zone is only a part of the tree, only this part will be checked. If there is no namespace declared, a declaration on root will be created if the option is enabled.

Using this option it is possible to manage a namespace in only a branch of the XML tree.

Panel:

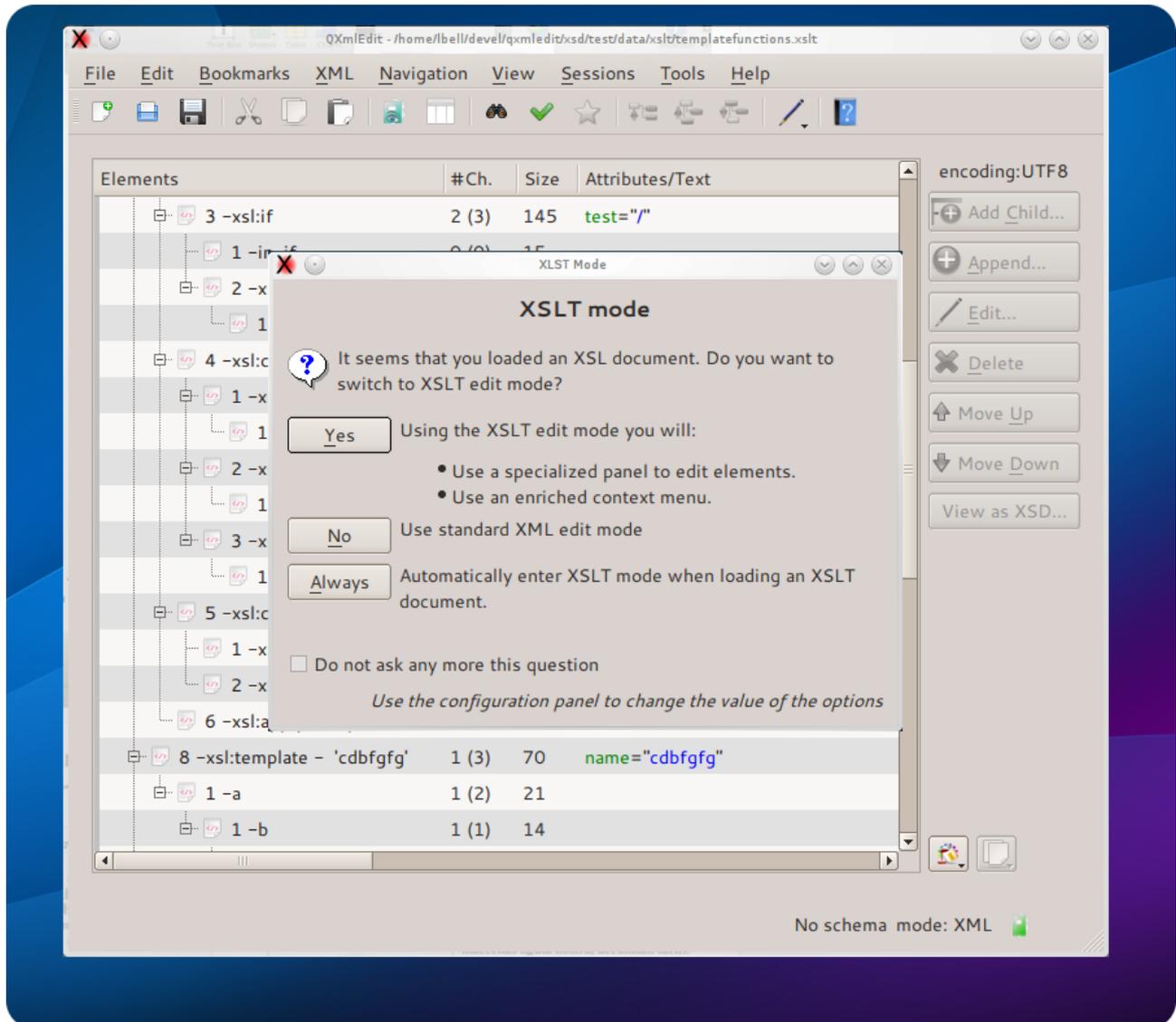


Items:

Item	Description
<b>Namespace</b>	The namespace to normalize.
<b>Prefix</b>	The prefix to apply to normalized namespace.
<b>Select</b>	Opens a namespace selection panel.
<b>Apply to</b>	Elements involved.
<b>Declare namespace on root</b>	The declaration of the namespace is moved to the root element only, any other inner declaration will be removed.

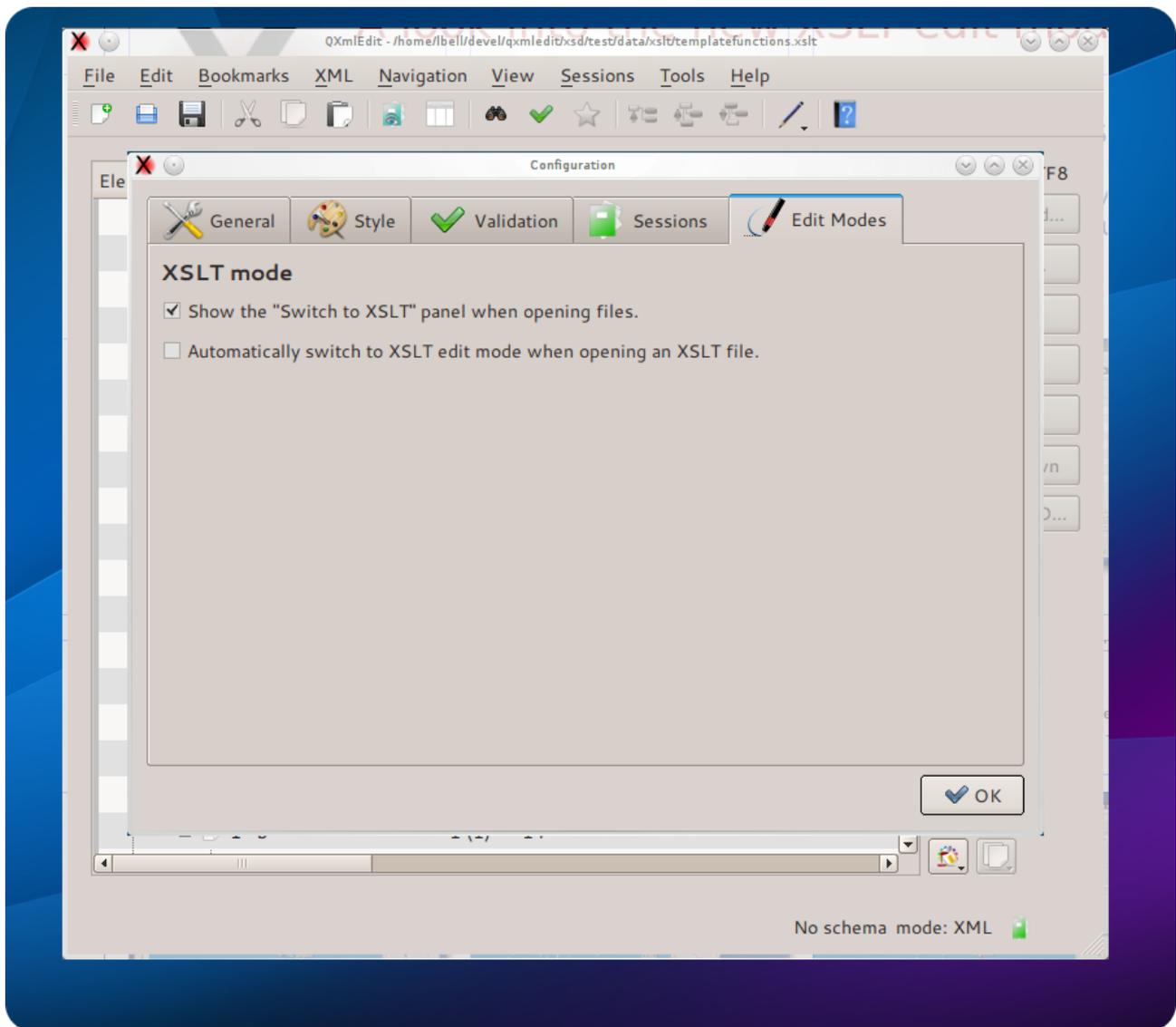
## Choosing an Editing Mode

Using the **XML > XSLT Edit Mode** menu item you can switch to XSLT edit mode. When loading a file with "xsl" extension or containing a reference to XSLT namespace a panel will open.



1. To activate the XSLT mode simply answer "Yes" or use the return key.
2. To have the mode always activated without answering the panel any more, select "Auto".

The settings can be also changed using the preferences.

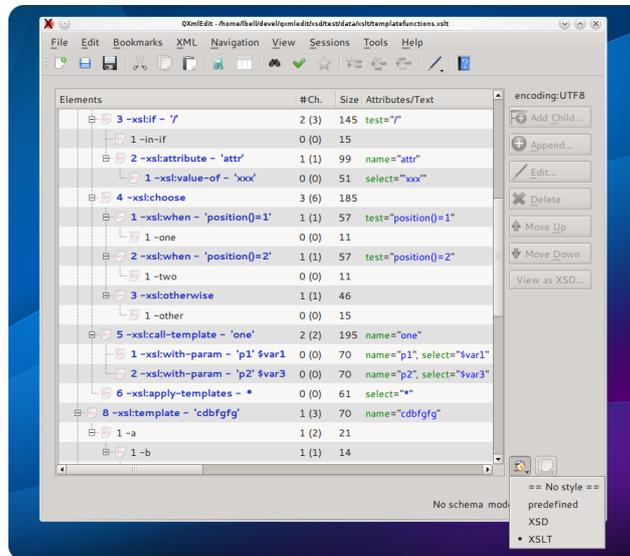


1. From the Edit menu choose "Configure", then "Edit modes".
2. Set if the XSLT edit mode should automatically engage.
3. You can hide the panel if you do not care to enter this mode.

## Using XSLT Edit Mode

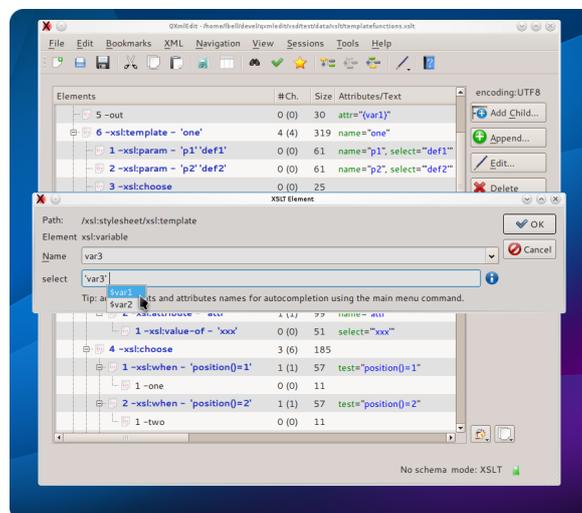
### Specific display style

A specific display style will be applied only to XSLT elements. The intermixed non XSLT elements will not be touched, to reveal the XSL structure at a glance.



### Specific edit panels

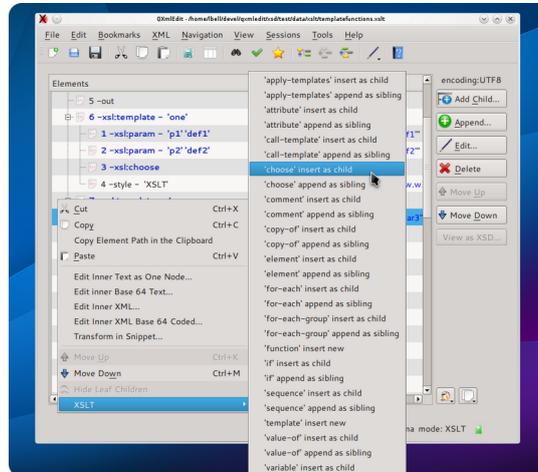
- 1- Double click an XSL element in the editor.
- 2- A special panel will open.
- 3- When you edit the "select" attribute, the auto-completion function will list the relevant variables.
- 4- When invoking templates or parameters, you will see their names in the combo.
- 5- To return to the usual property panel use CTRL-double click.



## A specific context menu

A new context menu appears on the editor elements. The menu propose to you the most used XSL elements giving you the choice to append them to the current element as sibling, or to insert them as a child.

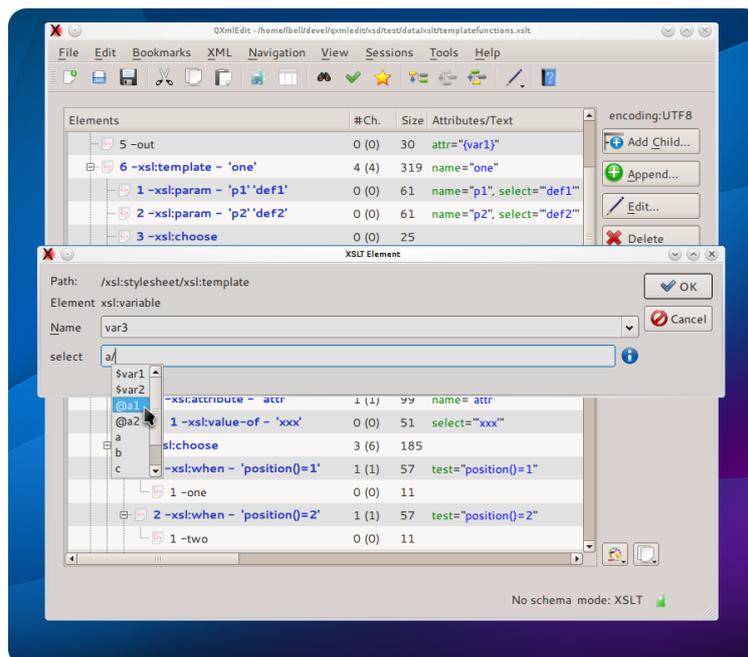
The proposed elements are sync with the context, that is in a "call-template", the only "with-parameter" element is proposed as a child.



Using names for auto-completion

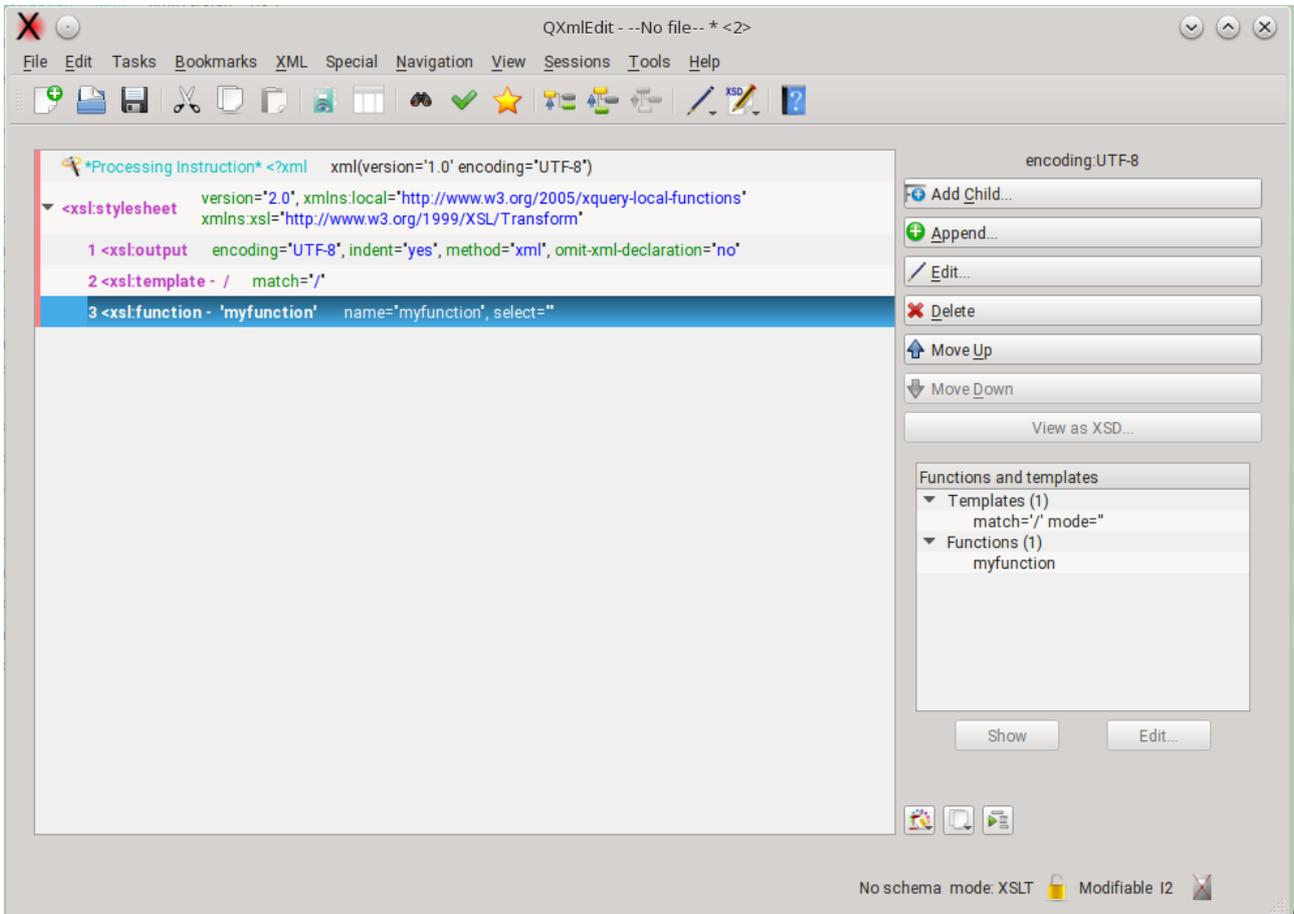
When XSTL mode is acting, you can scan the target XML file to extract names to be used with the auto-completion feature of the edit panel.

Use the menu **XML > Target XML**



## Navigate templates and functions

A XSL navigator for functions and templates will open automatically in the editor when choosing XSL edit mode. The navigator can be shown or hidden using the menu **Navigation > Show XSL Navigator**. A double click will lead you to the template or function definition in the editor.



## Using SCXML Edit Mode

### *Specific display style*

A specific display style will be applied only to SCXML elements. The intermixed non SCXML elements will not be touched, to reveal the SCXML structure at a glance.

### *Specific edit panels*

With SCXML edit mode enabled:

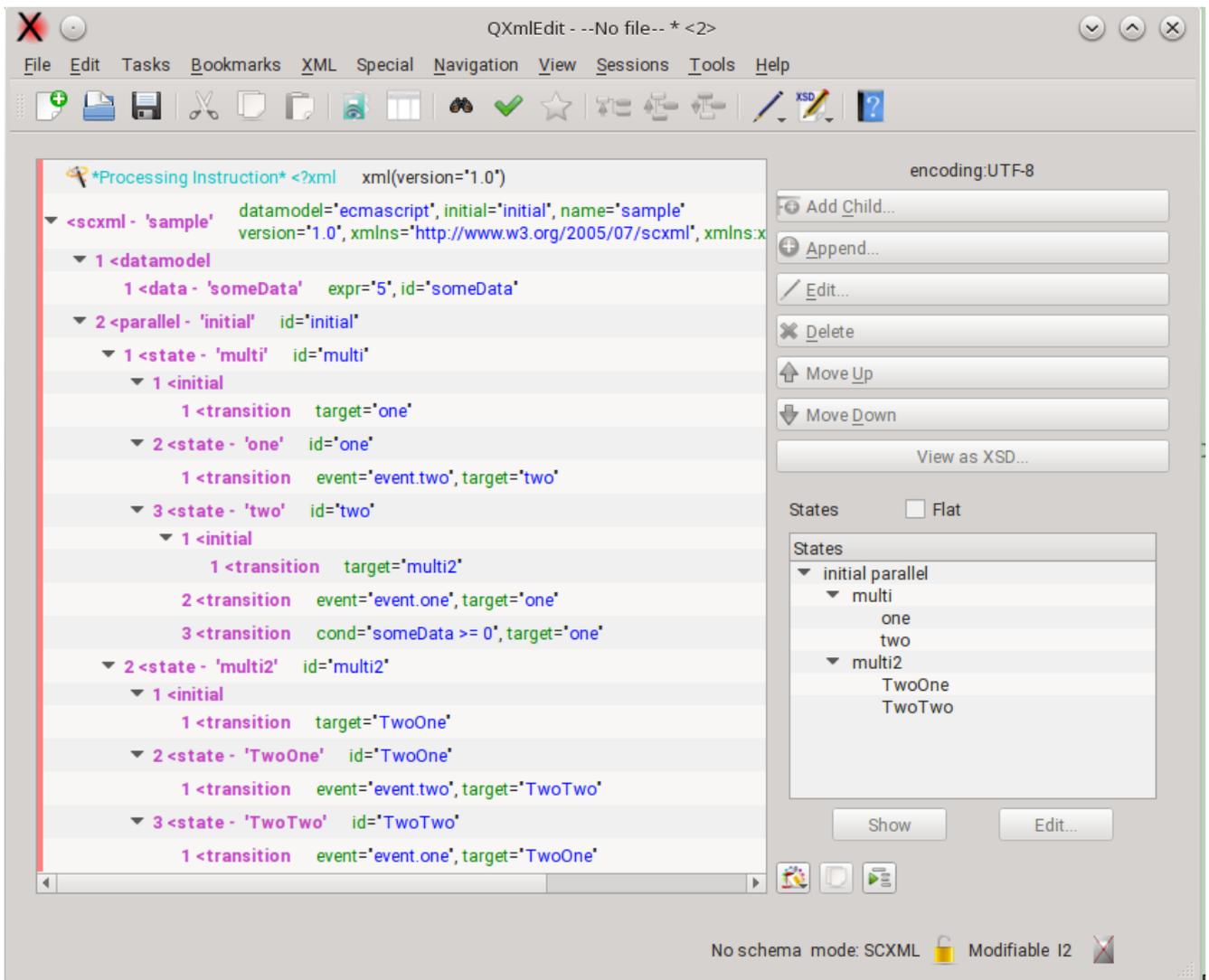
- 1- Double click an SCXML element in the editor.
- 2- A special panel will open.
- 3- When selecting a state name, a combo will reveal the list of states defined in the file..
- 5- To use to the usual element edit panel use CTRL or ALT while double clicking.

If the SCXML edit mode is not enabled, the double click modifiers are reversed.

### *Navigate states*

A SCXML navigator for states will open automatically in the editor when choosing SCXML edit mode. The navigator can be shown or hidden using the menu **Navigation > Show SCXML Navigator**. A click will lead you to the state definition in the editor, a double click will open the edit properties window.

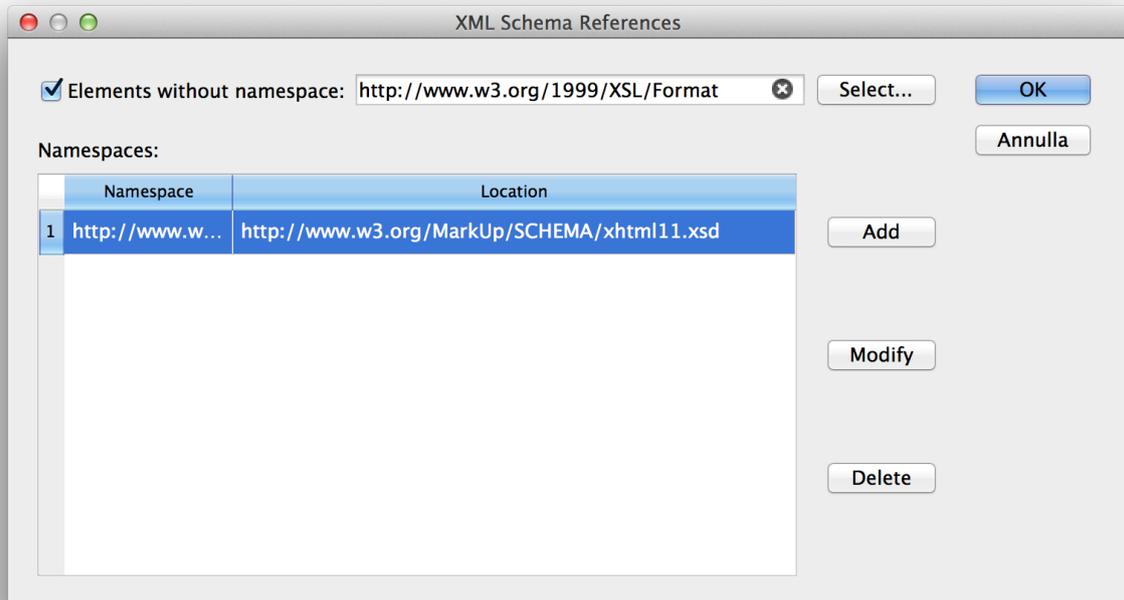
The list of states can be flat or hierarchical.



# XML Schema management

## Inserting XML Schema References

It is possible insert references to an XML Schema set of files. Using the menu **XML > Schema Instance > XML Schema References** the “Insert Xml Schema References” dialog will appear.



In this dialog you can insert

- a schema reference for elements without namespace
- for each namespace, a schema reference

The references will be inserted in the root element using the xsi namespace defined as `http://www.w3.org/2001/XMLSchema-instance` namespace. If the namespace is already declared in the root element, it will be used, else the namespace declaration will be inserted. If the prefix `xsi` is already used, a new prefix will be created.

## Using the XSD Edit Mode

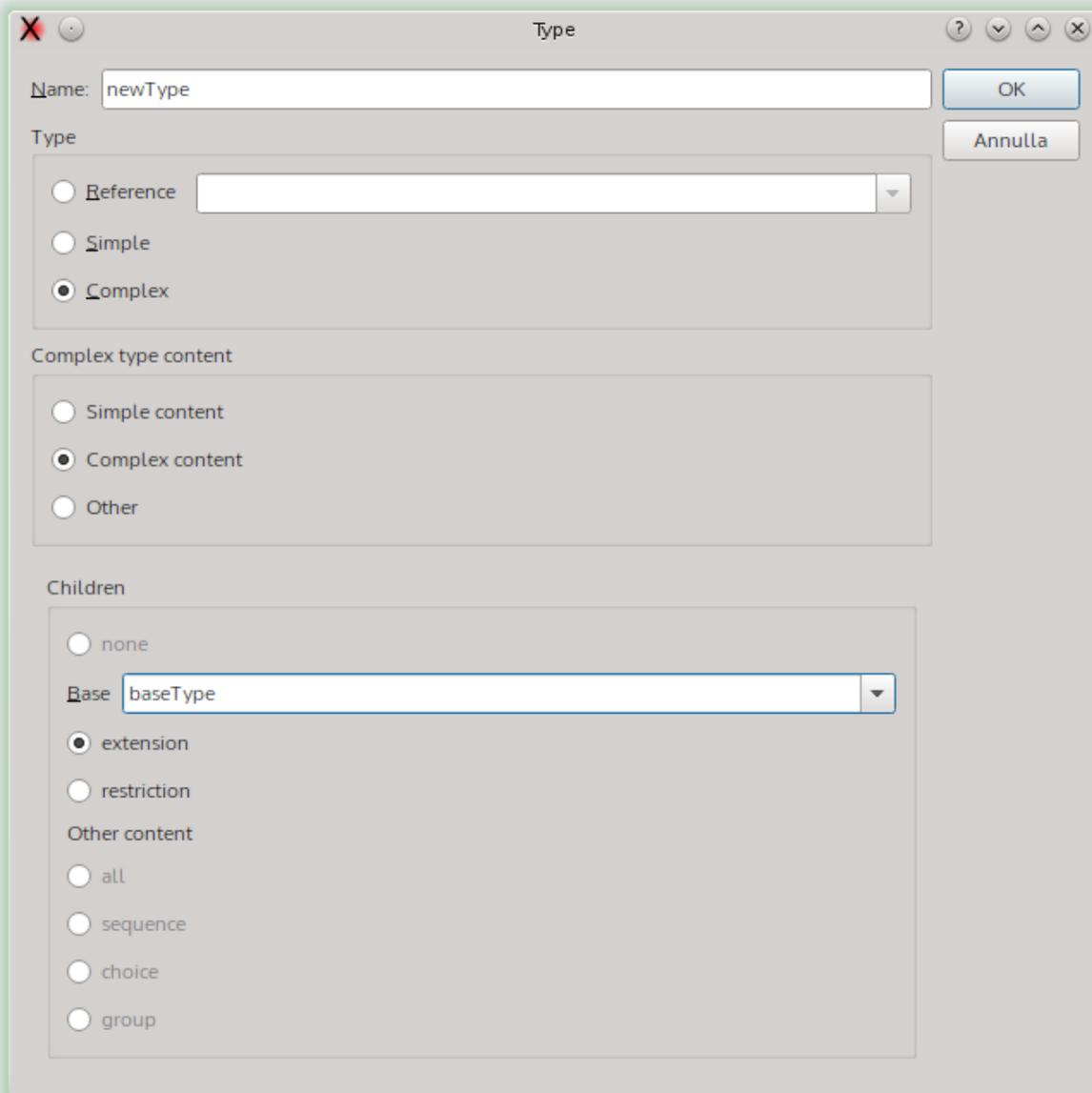
There is a facility for edit XML Schema documents in the editor a set of menu commands:

**XML Schema > Insert No Namespace Schema Reference Attributes** : insert a no namespace attribute for the target schema to complete.

**XML Schema > Insert Schema Reference Attributes** : insert a namespace attribute for the target schema to complete.

## Editing or inserting a type, attribute or element

Using the **XML Schema** menu items it is possible to edit or insert XSD types.



Operations in the dialog:

Choosing the type

- Reference: will set a reference to a type.
- Simple: inner simple type.
- Complex: inner complex type.

When choosing a complex type, it is possible to choose the content type.

If it is an update of existing data, the existent children that are not compatible with the selected type will be deleted, but the children that are compatible will be preserved.

## Edit an annotation

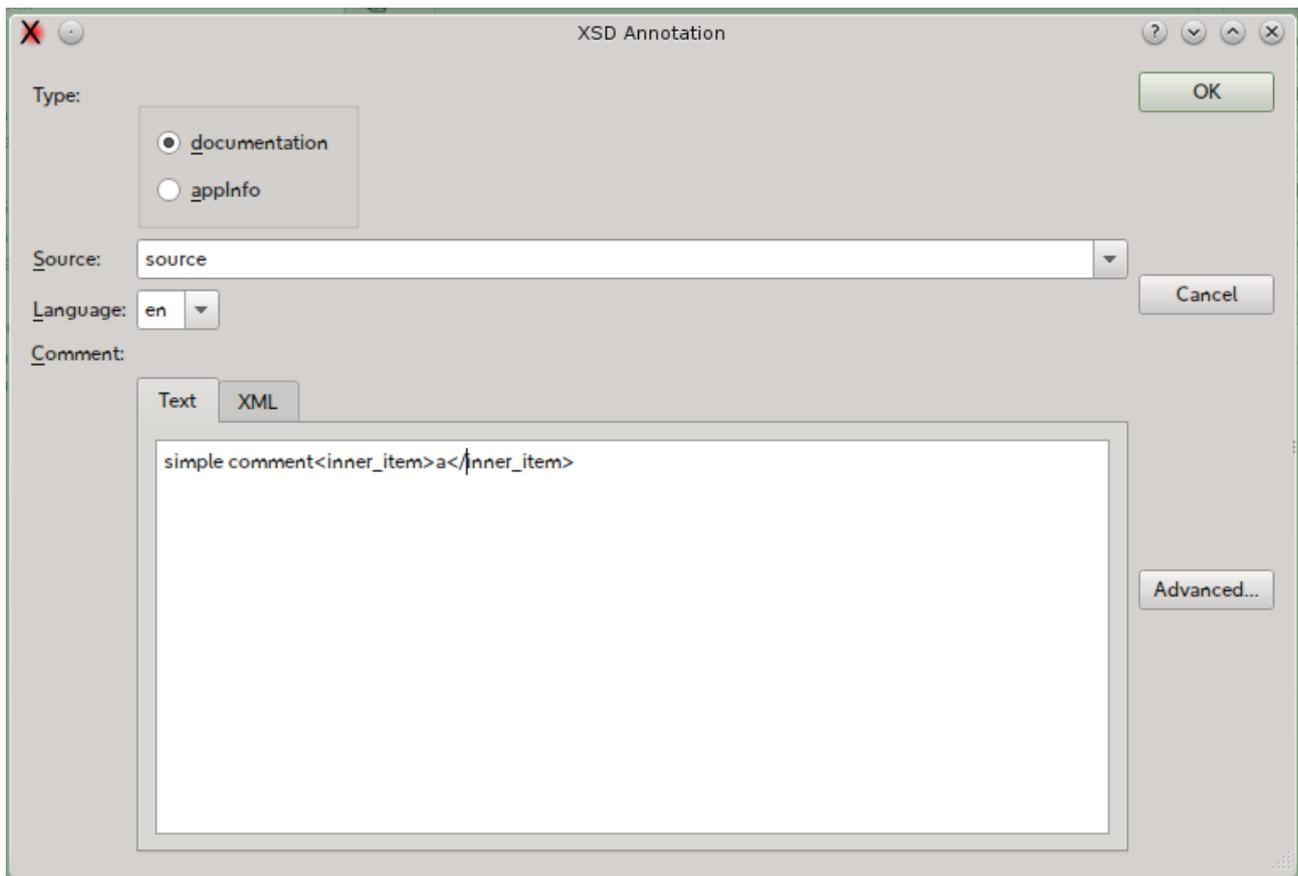
The annotation of XSD objects can be edited using the menu:

### **XML > XML Schema > Edit XSD Annotation**

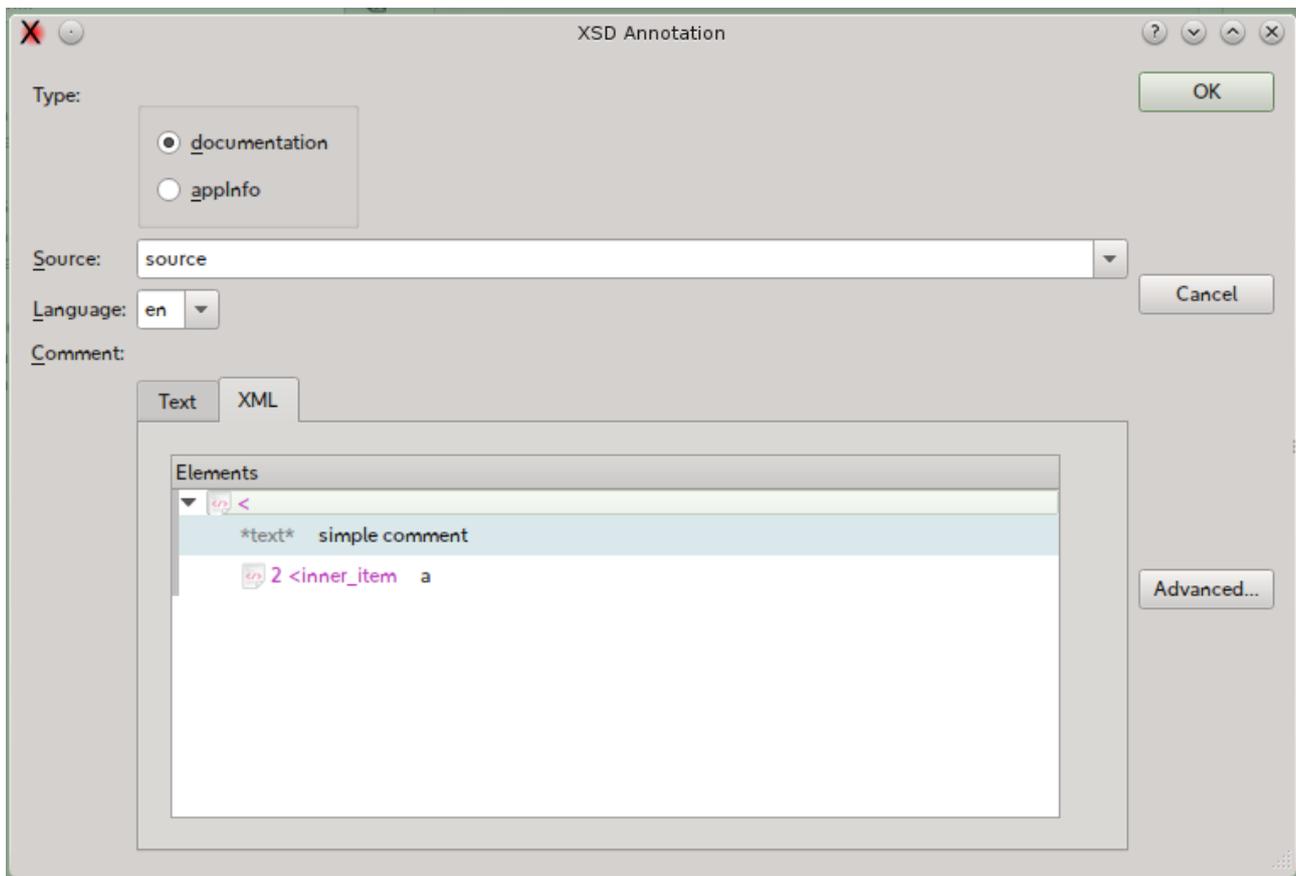
If the element has no annotation one annotation is automatically created and the edit panel is open. From this panel it is possible to open the complete edit annotations panel where more than one annotation can be inserted for the given element. If the element has already more than one annotation item, the full annotation panel is open.

### **Single annotation panel**

This panel configures a single annotation item.



The panel can show the inner contents of the annotation when containing XML mixed to text.

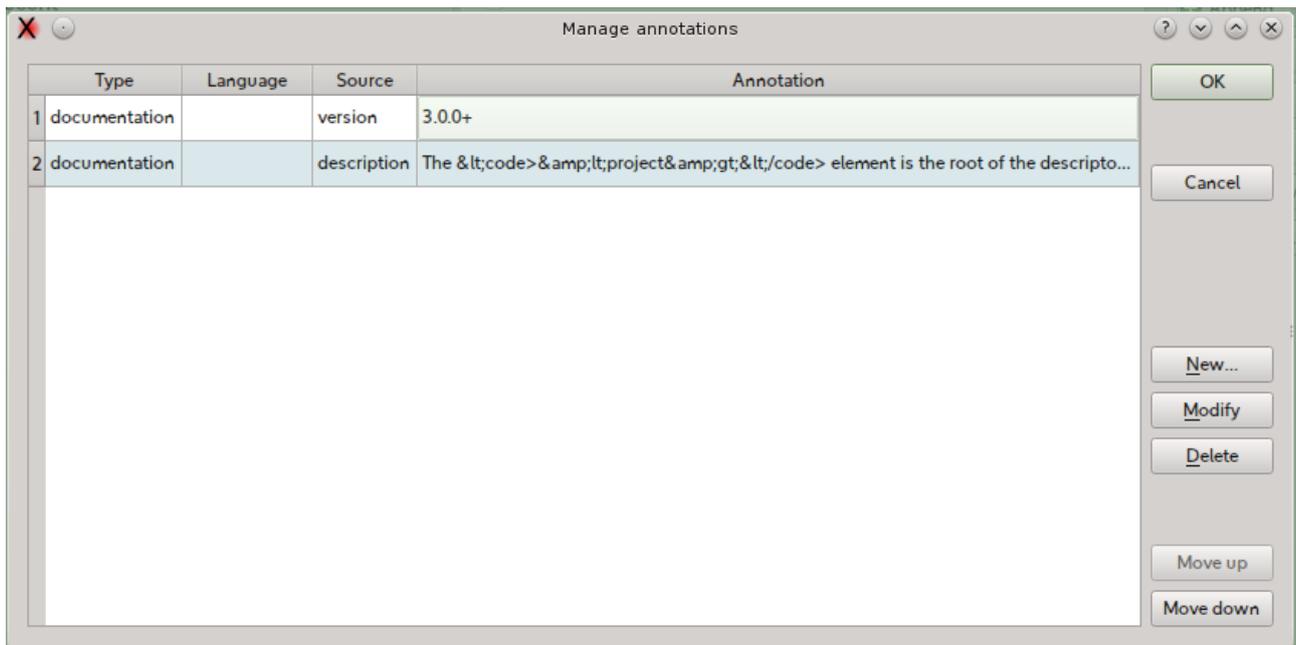


Available options:

Command	Functionality
Type	Switch the type of the annotation item between documentation and appinfo.
Source	Set the source of the item.
Language	Valid only for documentation type.
Comment/text	Edit field for the annotation item
Comment/XML	XML view of the annotation item. This view is read only.
Advanced	When the panel opens and the parent element has no items or only one, this button opens the advanced edit panel.

## Annotation list edit panel

This panel shows a list of annotation items. From here the items can be created, edited and deleted.



Available options:

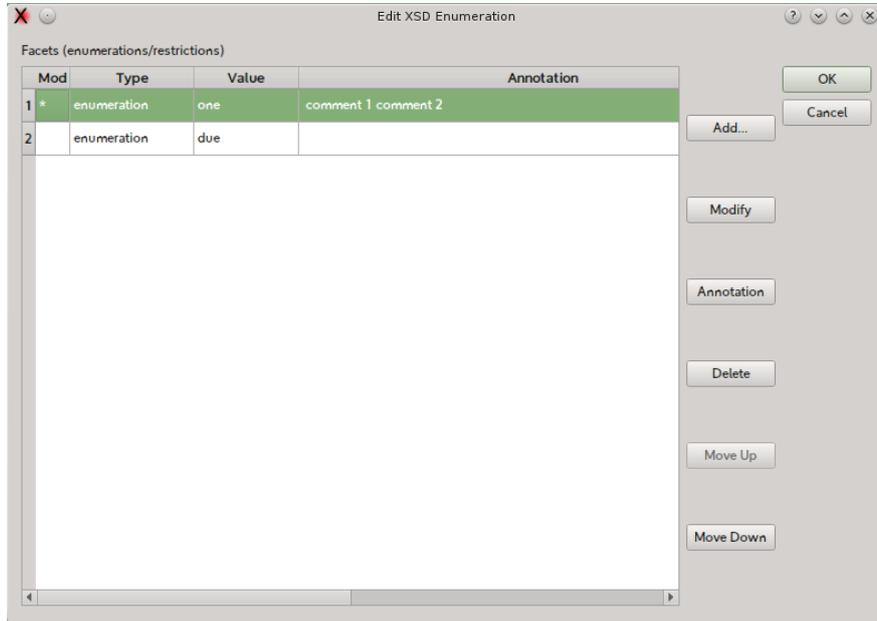
Command	Functionality
<b>New</b>	Create a new item.
<b>Modify</b>	Edit the selected item. This option can be triggered also by a double click on the list.
<b>Delete</b>	Delete the selected item.
<b>Move Up</b>	Move the item one position up in the list.
<b>Move Down</b>	Move the item one position down in the list.

## Editing Enumeration or Facets

The facets can be edited selecting the restriction element and using the menu item:

**XML > XML Schema > Edit Enumeration or Facet**

The panel lists all the facets. The items can be reordered, edited and annotated. Not legal elements are discarded.



## Edit facet panel

This panel edits a single facet item.



## Editing XML Schema Attributes

### *Inserting the Schema Instance “nil” Attribute*

Select an element and use the menu **“XML>Schema Instance>Insert “nil” Attribute”** to insert the Schema Instance (XSI) “nil” attribute. If the namespace “xsi” is not present, it will be inserted in the element, changing the prefix if the same prefix is already used; after that the attribute is inserted.

### *Removing the Schema Instance “nil” Attribute*

Select an element containing the Schema Instance “nil” attribute and use the menu **“XML>Schema Instance>Remove “nil” Attribute”** to remove the XSI “nil” attribute. If the XSI namespace is declared in the elements and it is not used in the same element or its children, the namespace is removed too.

### *Inserting the Schema Instance “type” Attribute*

Select an element and use the menu **“XML>Schema Instance>Insert “type” Attribute”** to insert the Schema Instance (XSI) “type” attribute. If the namespace “xsi” does not exist, it will be inserted in the element, changing the prefix if the same prefix is already used; after that the attribute is inserted. The type can be chosen by using a dialog.

### *Removing the Schema Instance “type” Attribute*

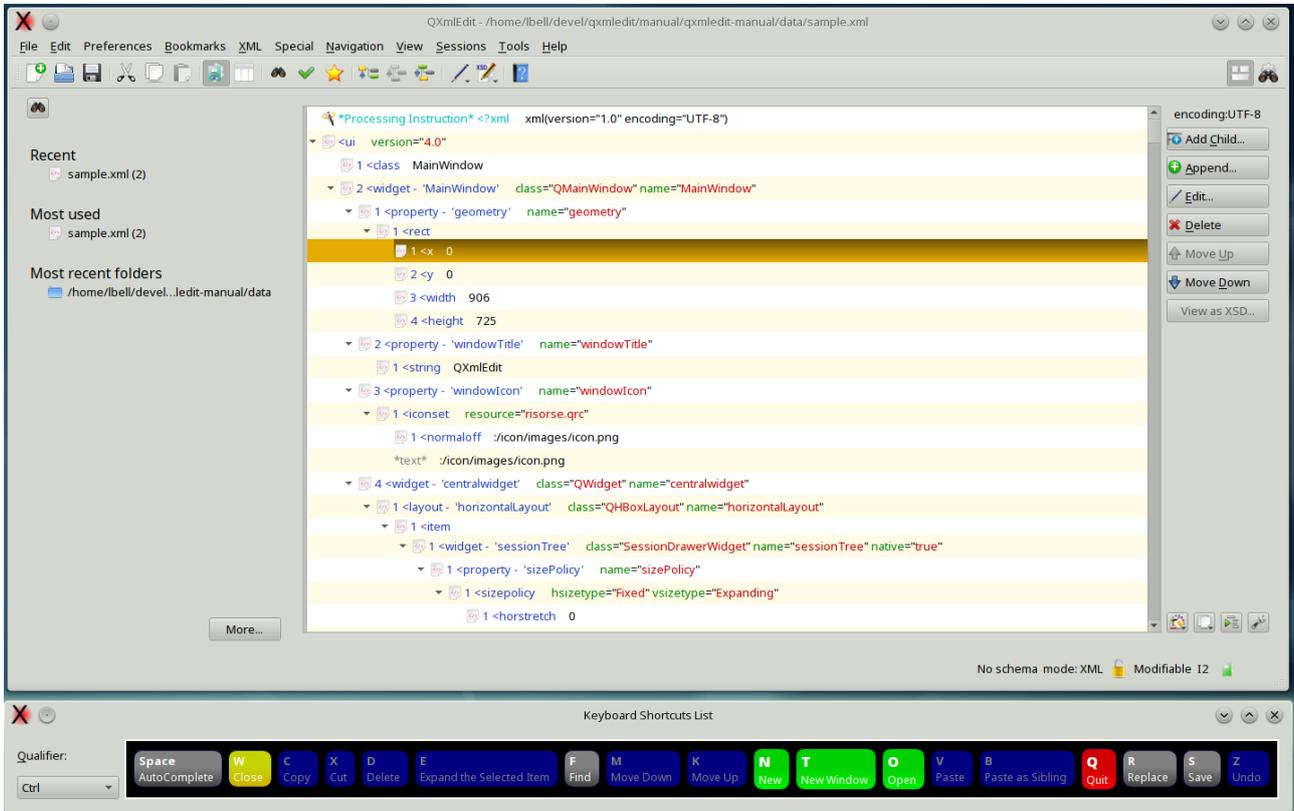
Select an element containing the Schema Instance “nil” attribute and use the menu **“XML>Schema Instance>Remove “type” Attribute”** to remove the XSI “type” attribute. If the XSI namespace is declared in the elements and it is not used in the same element or its children, the namespace is removed too.

## Help and information

There are some facilities to discover and search commands

## Show keyboard shortcuts

With the toolbar button or the menu Help>Show Keyboard Shortcuts the shortcuts bound to the current editor window can be visible listed by ket qualifier.



Changing the qualifier key in the combo the shortcuts introduced by that qualified are reported. The buttons can be pressed using the mouse.

## Tasks

Tasks are operations than can be accomplished using different commands. They are a sort of macro, The following tasks are predefined:

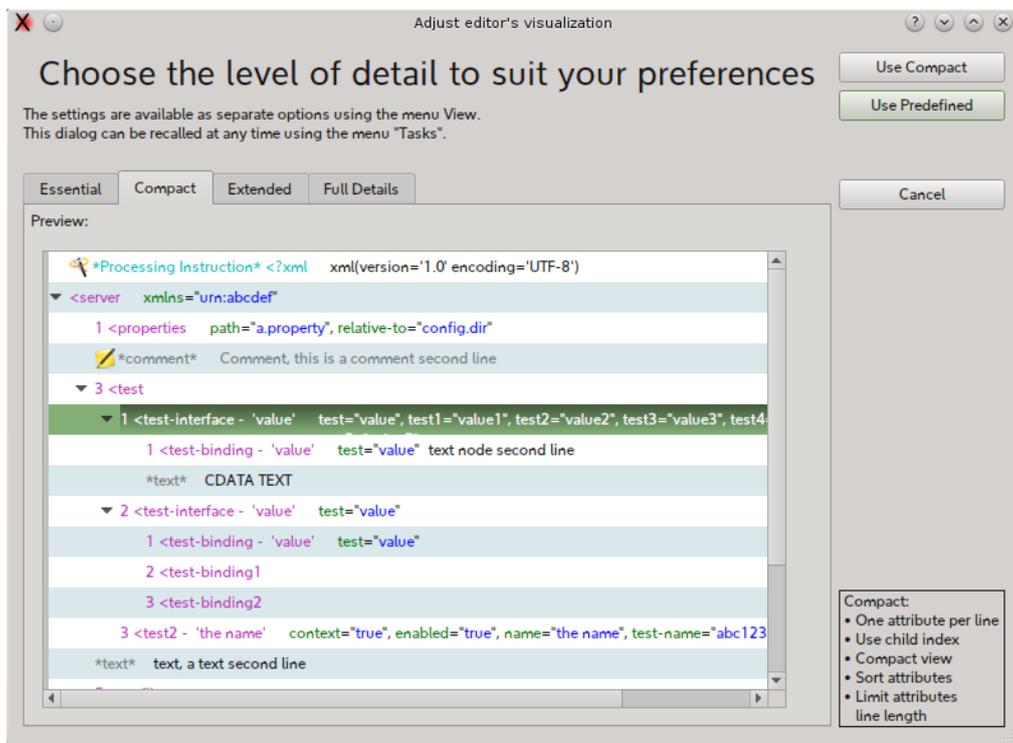
Task	Description
<b>Adjust Display Detail</b>	Set the editor view options.

## Adjust Display Detail

With this task the level of detail of the editor is set. You can choose between some predefined settings and have a preview of each one.

Options:

Command	Description
<b>Use Predefined</b>	Use the predefined settings.
<b>Use (value)</b>	Use the style shown in the selected tab.



## Preferences

The preferences panel can be opened using the menu **Edit > Configuration**.

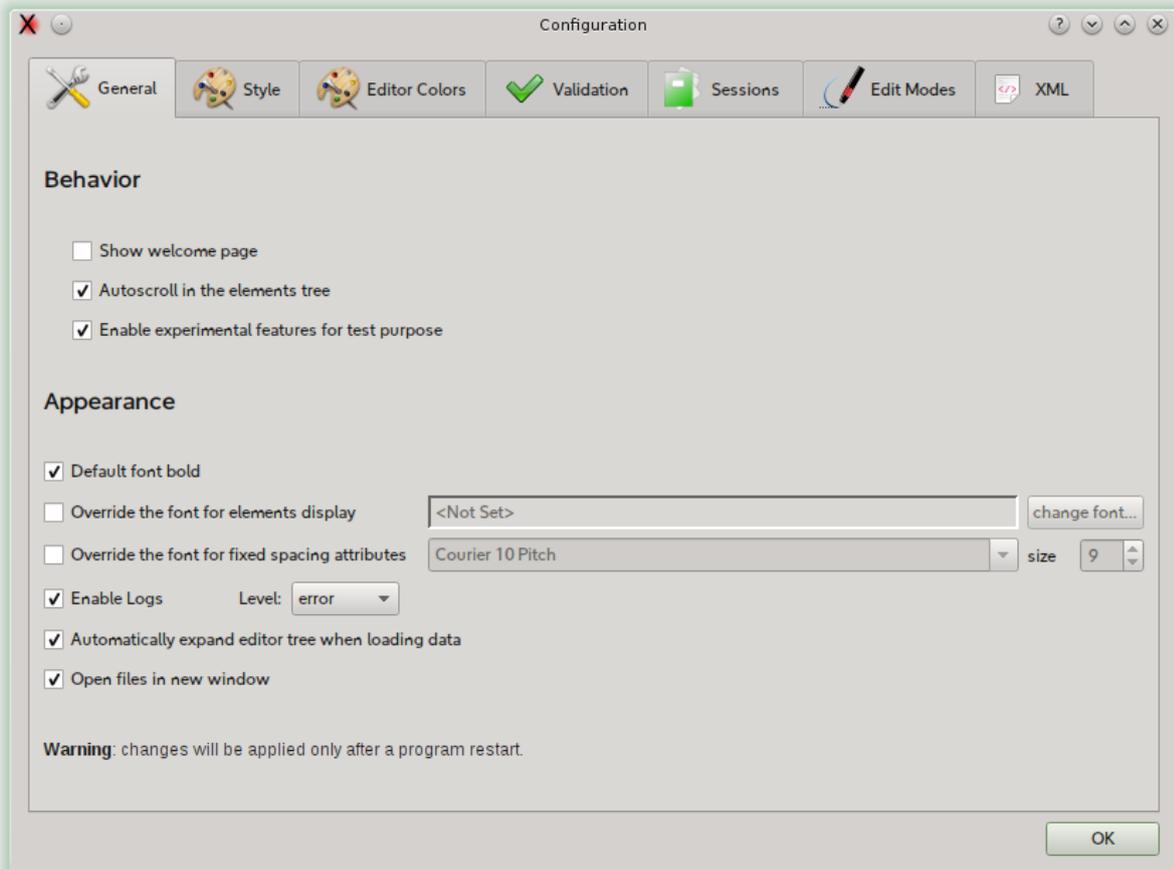
The panel allows the configuration of the following categories:

Information	Description
<b>General</b>	General options.
<b>Style</b>	Management of display style in the editor.
<b>Editor Colors</b>	The name says all.
<b>Validation</b>	Option related to XML and XML Schema validation.
<b>Sessions</b>	Management of sessions data.
<b>Edit Modes</b>	Activation of special modes in the editor (e.g. XSLT).
<b>XML</b>	XML management.

Here a detailed explanation of the options:

## General

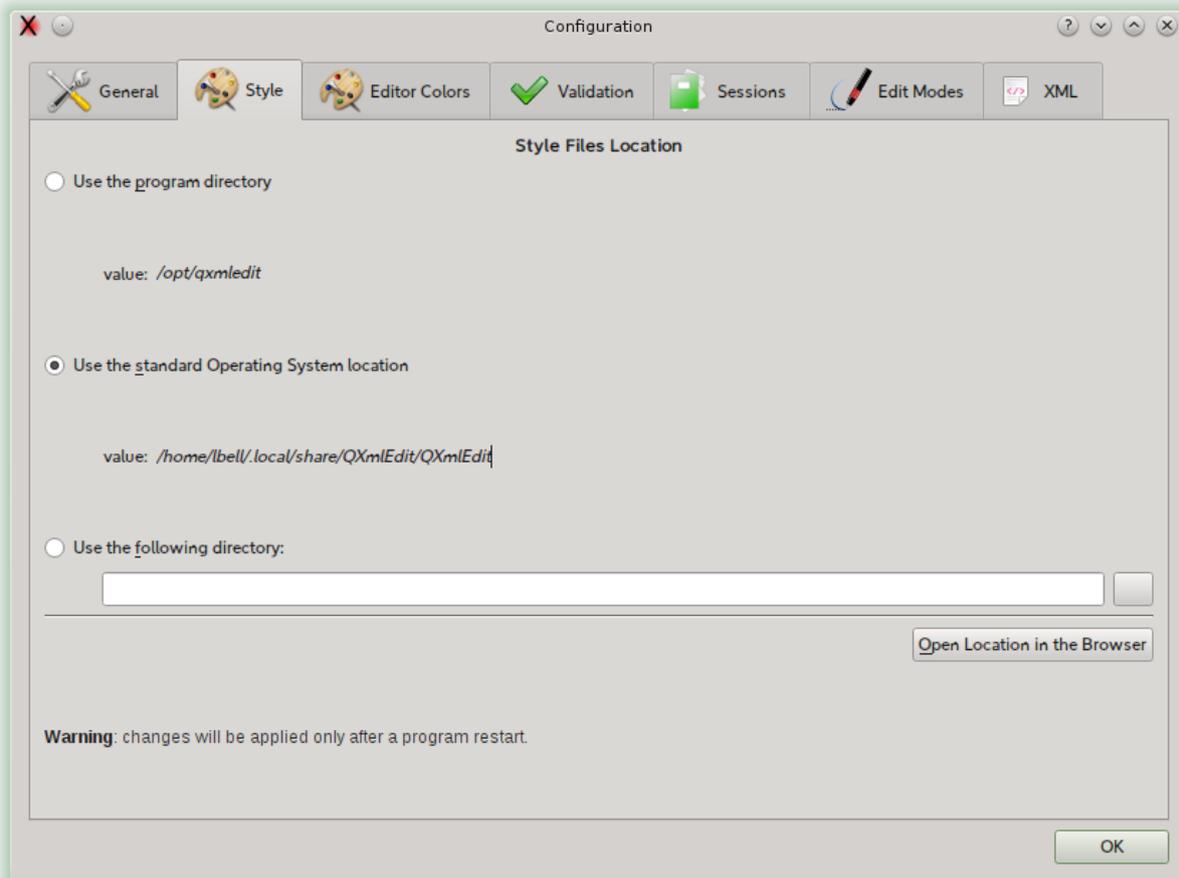
Option	Description
<b>Show welcome page</b>	Shows the welcome page at program start up.
<b>Enable experimental features for test purpose</b>	Enables experimental feature in development editions.
<b>Default font bold</b>	Use bold font weight for elements.
<b>Override the font for elements display</b>	Use a user chosen family in the editor.
<b>Override the font for fixed spacing attributes</b>	Use a custom font when rendering attributes with mono spaced font.
<b>Enable Logs</b>	Enable logging.
<b>Automatically expand editor tree when loading data</b>	Expands (open) the whole tree after loading a new file.
<b>Autoscroll in the elements tree</b>	Enable autoscroll when dragging items
<b>Open files in new window</b>	A new window is used each time a file is loaded.
<b>Limit attributes length</b>	The maximum length of the attributes lines in XML visualization.



## Style

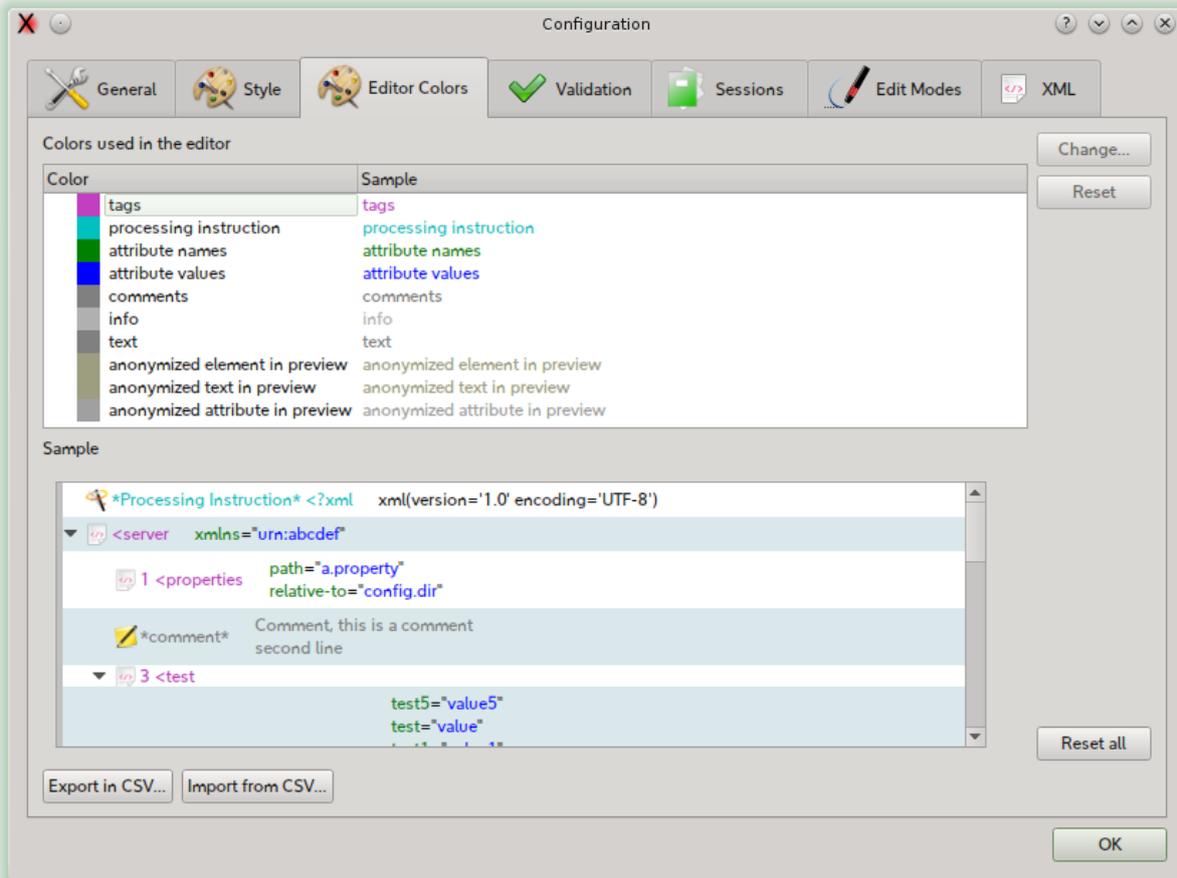
This panel lets you choose the location for user defined styles files.

Option	Description
<b>Use the program directory</b>	Use the installation folder
<b>Use the standard Operating System location</b>	Use the default location for data.
<b>Use the following directory</b>	Use a user defined folder.



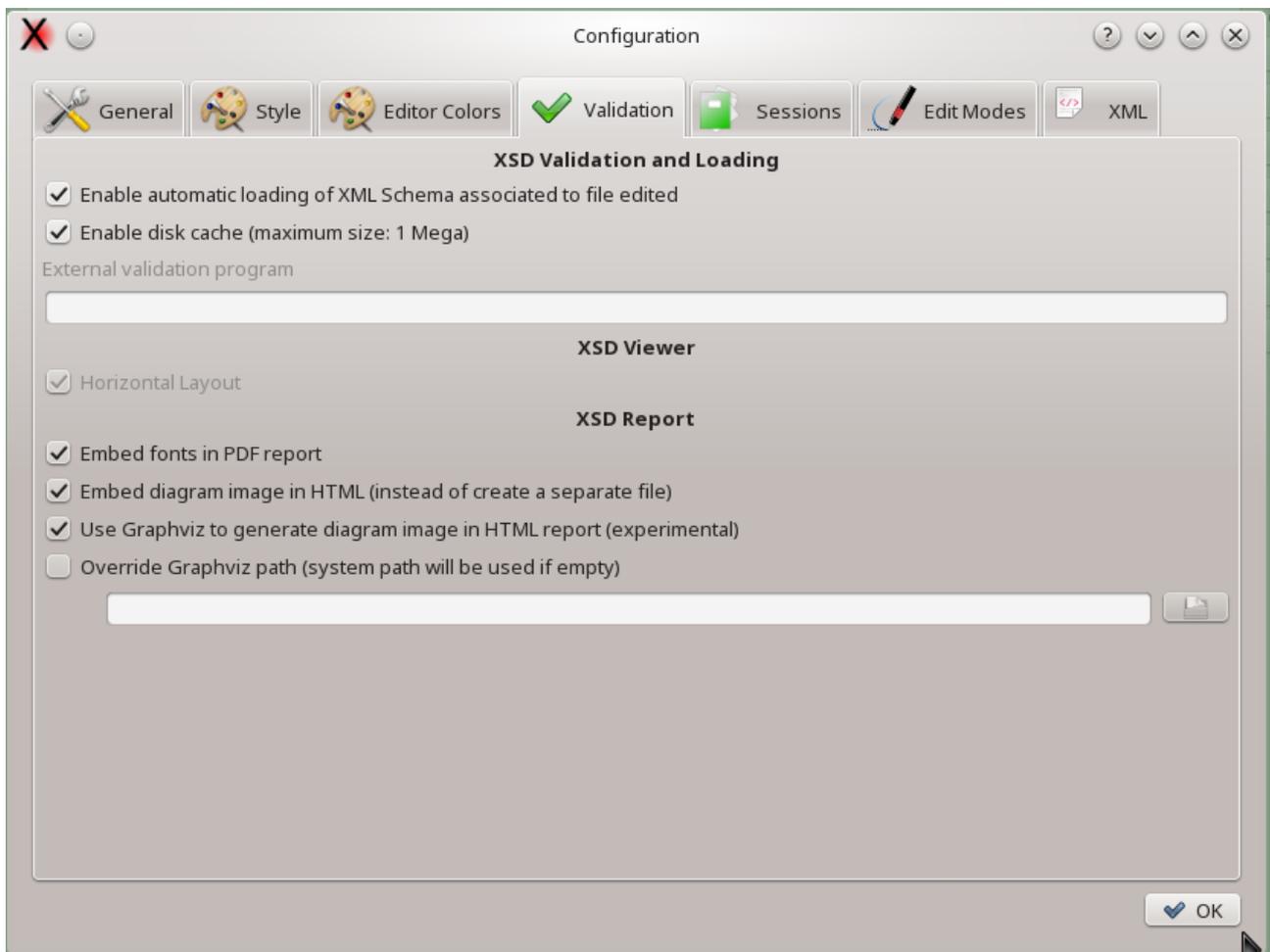
## Editor colors

This dialog is described in details in the Style section. You can customize the colors of the editor.



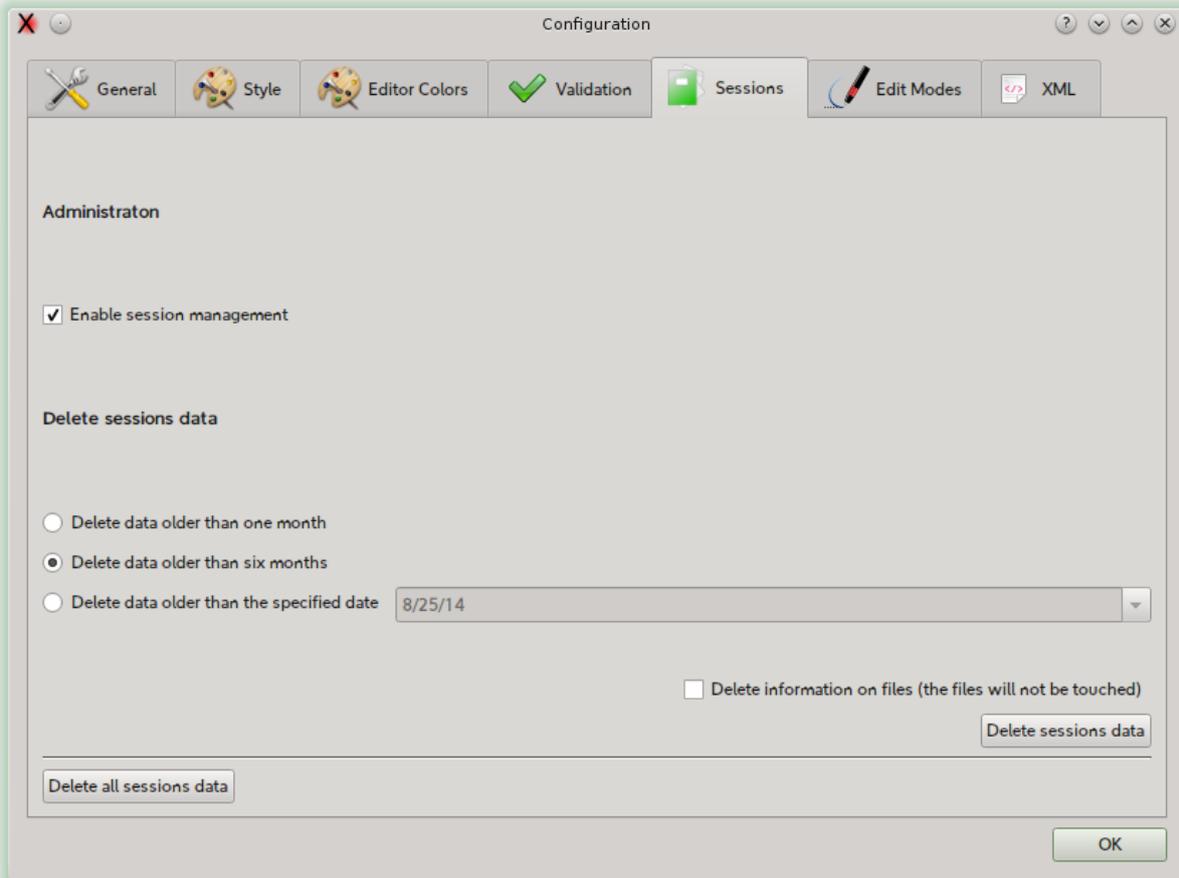
## Validation

Option	Description
<b>Enable automatic loading of XML Schema associated to file edited</b>	Automatically loads referenced XML schema files in the background at the XML file loading.
<b>Enable disk cache (maximum size: 1 Mega)</b>	Enable cache for XML schema files to avoid unnecessary network traffic.
<b>Embed fonts in PDF report</b>	Option to embed fonts in PDF report
<b>Embed diagram image in HTML</b>	If the diagram image in HTML report should be inlined or generated as external file.
<b>Use Graphviz to generate diagrams in HTML</b>	Use GraphViz (not included in QXmlEdit) to generate the diagram image. This option is experimental
<b>Override GraphViz path</b>	Custom path to dot executable.



## Sessions

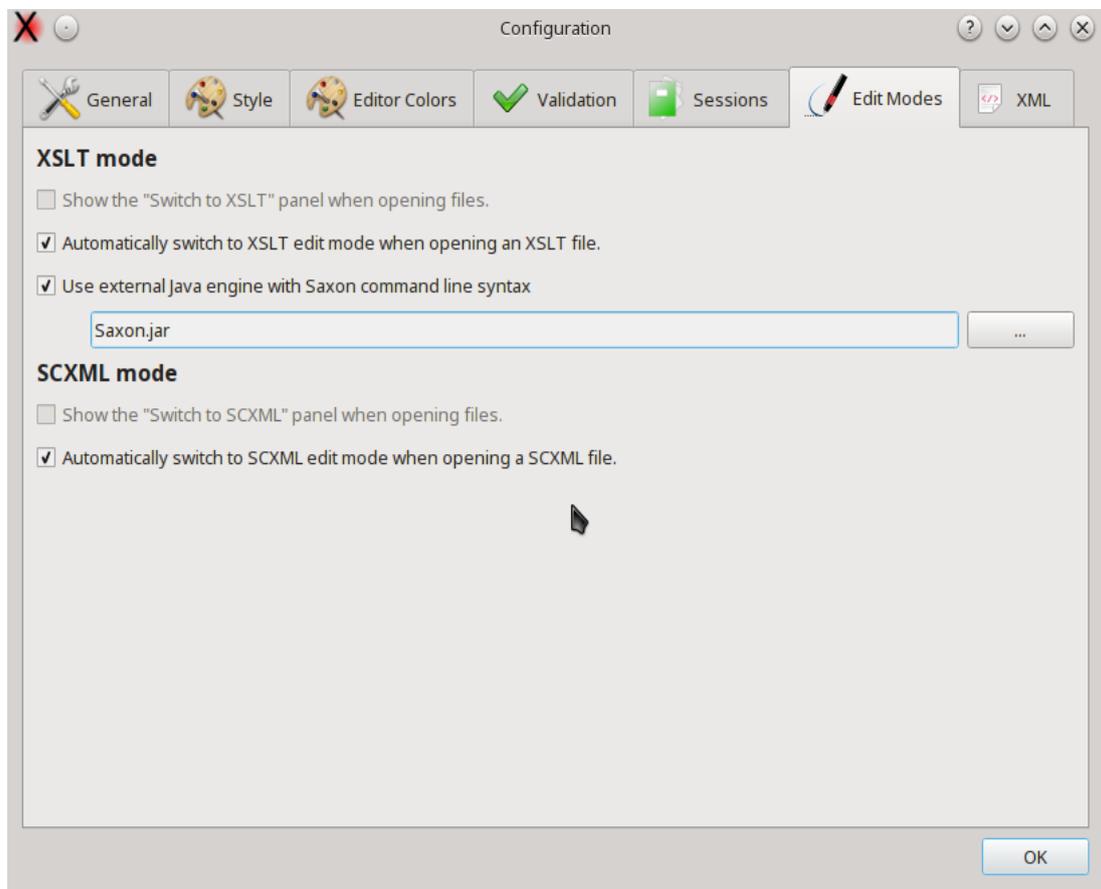
Session options described in the Session section.



## Edit Modes

This panel for the options related to specialized edit modes.

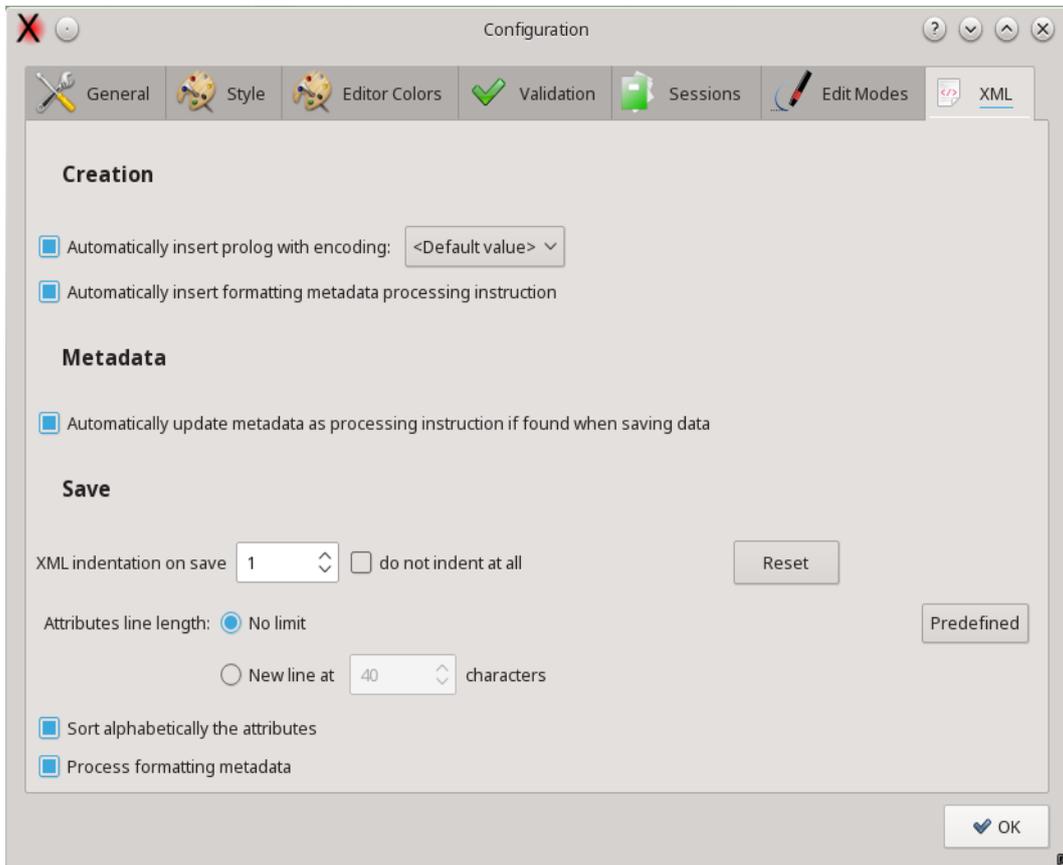
Option	Description
<b>Automatically switch to XSLT edit mode when opening an XSLT file.</b>	When loading a new file, if it is recognized as a XSLT documents, XSLT edit mode is automatically enabled.
<b>Show the "Switch to XSLT" panel when opening files.</b>	When loading a new file, if it is recognized as a XSLT documents, open the XSLT edit mode panel.
<b>Use external Java engine with Saxon command line syntax</b>	If you want to use an external XSLT Java engine having the same command line interface as Saxon, insert here the path of the jar.
<b>Automatically switch to SCXML edit mode when opening a SCXML file.</b>	When loading a new file, if it is recognized as a SCXML documents, SCXML edit mode is automatically enabled.
<b>Show the "Switch to SCXML" panel when opening files.</b>	When loading a new file, if it is recognized as a SCXML documents, open the SCXML edit mode panel.



## XML

### XML management options

Option	Description
<b>Automatically insert prolog with encoding</b>	When creating new files, insert XML prolog with optional encoding specification.
<b>Automatically update metadata as processing instruction if found when saving data</b>	Enable the update of QXmlEdit specific metadata in the file when saving.
<b>Automatically insert formatting metadata process instruction</b>	When creating a new XML document insert a processing instruction that holds formatting settings (indentation, ...)
<b>XML indentation on save</b>	Enable or disable XML indentation when saving. A value of 0 still saves on element per row. You have to disable it explicitly if you want only one output text line.
<b>Save using stream instead of DOM</b>	Enabled by default. Allow a finer control on input and output. DTD declarations are not handled if this option is unchecked.
<b>Sort alphabetically the attributes</b>	On saving the attributes will be sorted alphabetically on their name.
<b>Attributes line length</b>	It limits the start column of the attributes. If a limit is set, if the attribute start at a column that is greater than the distance from the tag end than the limit, a new line is started.
<b>Limit attribute length to X columns</b>	Limits the length of the attribute zone when saving the file.
<b>Process formatting metadata</b>	When reading a document, if formatting metadata are found, they will be applied.



## Customization

### Choosing a Display Style

QXmlEdit can use a different display style, the font used, its dimensions and the color, to represent different XML structures. The style description can be created by the user and applied on demand. QXmlEdit comes with some predefined styles.

The style can display the value of some attribute near to the element tag to highlight the element type.

To apply a style:

- In the main window, click the styles popup  ( ), then choose the desired style.

To remove a style:

- In the main window, click the styles popup (  ), then choose the "No Style" item

The style files location can be configured using the menu item **Edit > Configure**.

In the appendix there is the complete style reference.

An example of style applied to an XML Schema file is represented in the following screenshot:



The style can be used to highlight the elements.

▼  4 -event	2 (2)	266	logger="logger1", time
 1 -message	0 (0)	28	<html><body bgcolor...
 2 -locationInfo	0 (0)	122	file="Test.java", line='
▼  5 -event	2 (2)	265	logger="logger1", time

The Style can be based on rules, in the appendix there is the full detail.

## Editing Elements

Basically you can use the tree view to insert child nodes or append nodes as brothers of the current one. There is a separate input panel for each node type you can manipulate: elements, comments, processing instructions.

To insert an element as a child of some other node:

- Select the ***XML > Add a Child Element*** menu.

or

- Click the button Add Child... on the main window.

To append a node as brother of the current selection:

- Select the ***XML > Append a Brother Element*** menu.

or

- Click the button ***Append...*** on the main window.

To work with attributes:

- open the element edit panel and start editing

To insert a processing instruction as a child of some other node:

- select the ***XML > Add Processing Instruction as a Child*** menu

To append a processing instruction as a brother of some other node:

- select the ***XML > Add Processing Instruction as Brother*** menu

To remove a node of any type (element, attribute, comment, processing instruction):

- click the button ***Delete*** on the main view

## Navigating Elements

There are some shortcuts to move from one element to another one using the relationship between the two. These shortcut are handful especially when the data tree is very tall, and avoid you the necessity to continually scroll the window while examining data.

To rise from an element to its parent:

- select the element, then press ***F11***

To jump to the next element at the same level of the current one:

- select the element, then press ***F10***

To jump to the previous element at the same level of the current one:

- select the element, then press **F9**

To close all the children of an element and all the children of their brothers (it gives you a compact view of the tree level in one operation):

- select the element, then press **F12**

## Editing Elements and Attributes

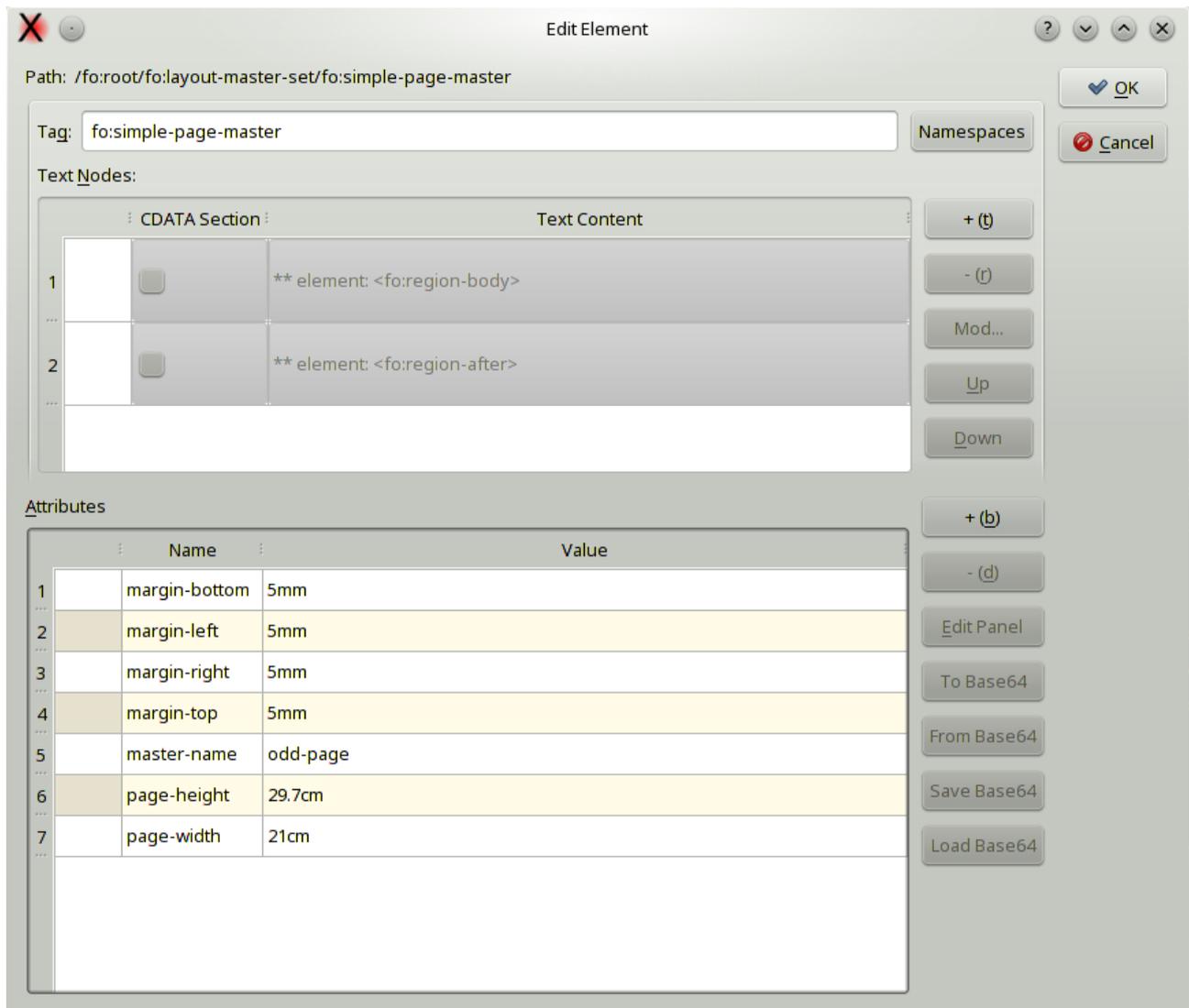
To edit the selected object:

- double click on the selected element

or

- press the **Edit...** button in the main window

When editing elements a panel will open. In the panel the current element path is shown; it is possible to manage namespaces by recalling a library of user defined data. It is possible to manage base 64 coded attributes.



In the panel you can insert the following information:

Name	Information
Tag	The tag of the element
Text Nodes	You can insert or remove text nodes as element children
Attributes	You can edit the element attributes

To insert a text node:

- press the **+** button in the text nodes section

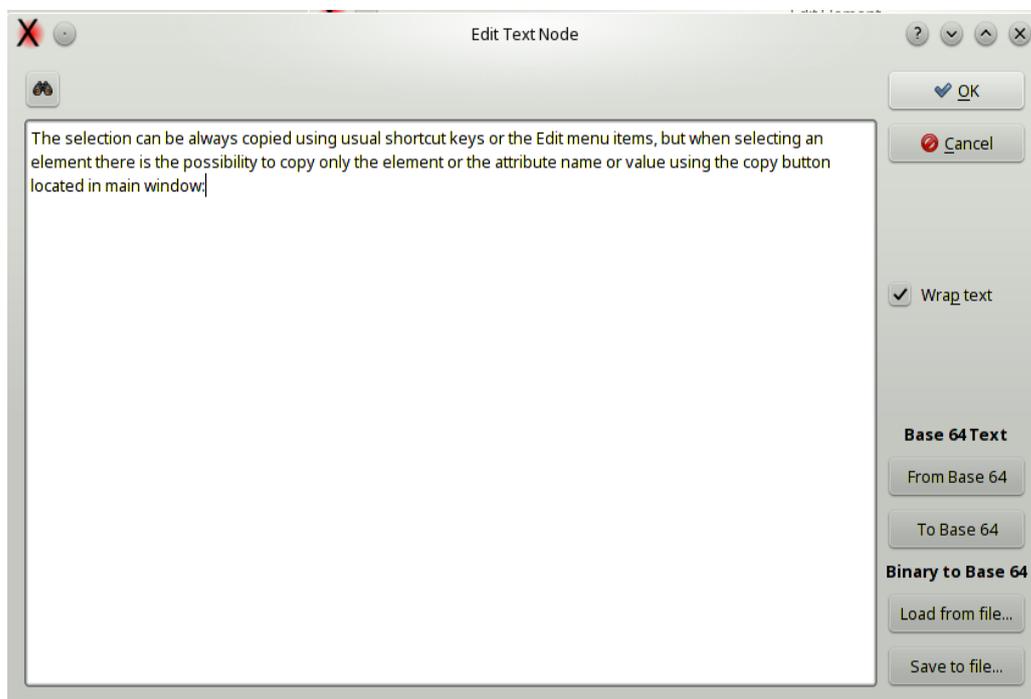
To insert a CDATA text node:

- press the **+** button in the text nodes section then activate the **CDATA** check box on the text nodes table

To modify text node content:

- Select the text node, then press the **Mod...** button. A panel will open.

**Note:** you cannot edit child element nodes in this panel, only text nodes.



To insert a base 64 coded text:

- Insert the text, then press **To Base 64** button, then save the text.

To modify a base 64 coded text:

- Press the **From Base 64** button, the text will be decoded, modify the text, then press **To Base 64** button, then save the text.

To view the text on more lines even it is a single line:

- Activate the **Wrap text** check box.

To load a binary file as base 64 coded text:

- Press the **Load from file...** button, then browse to the file in the file dialog.

To save current base 64 coded text into a binary file, decoding the values:

- Press the **Save to file...** button.

## Searching text

In text panel there is a tool button at the top just over the text box; when pressed the search panel appears.

The option for text search are:

- Search only whole words.
- Search with case sensitivity.

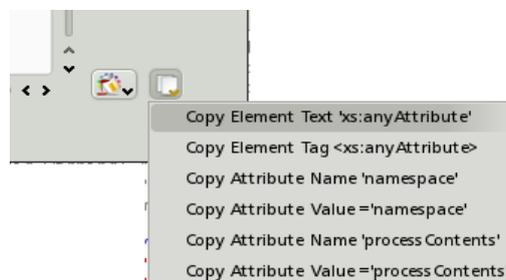
## Deleting siblings

The menu **XML>Siblings** lets you

- Delete the siblings before the selected element.
- Delete the siblings after the selected element.
- Delete all the siblings of the selected element.

## Copying Selected Text Into the Clipboard

The selection can be always copied using usual shortcut keys or the Edit menu items, but when selecting an element there is the possibility to copy only the element or the attribute name or value using the copy button located in main window:



to copy the selected element tag:

- click the **Copy Element Tag** item

to copy the selected element content:

- click the **Copy Element Text** item

For each attribute of the selected element:

to copy the attribute name:

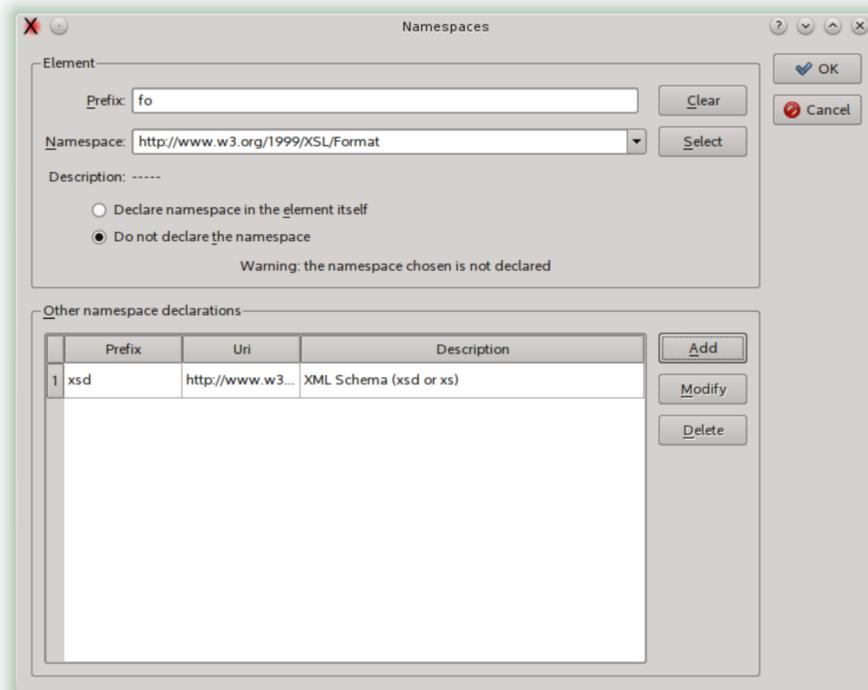
- click the **Copy Attribute Name** item

to copy the selected attribute value:

- click the **Copy Attribute Value** item

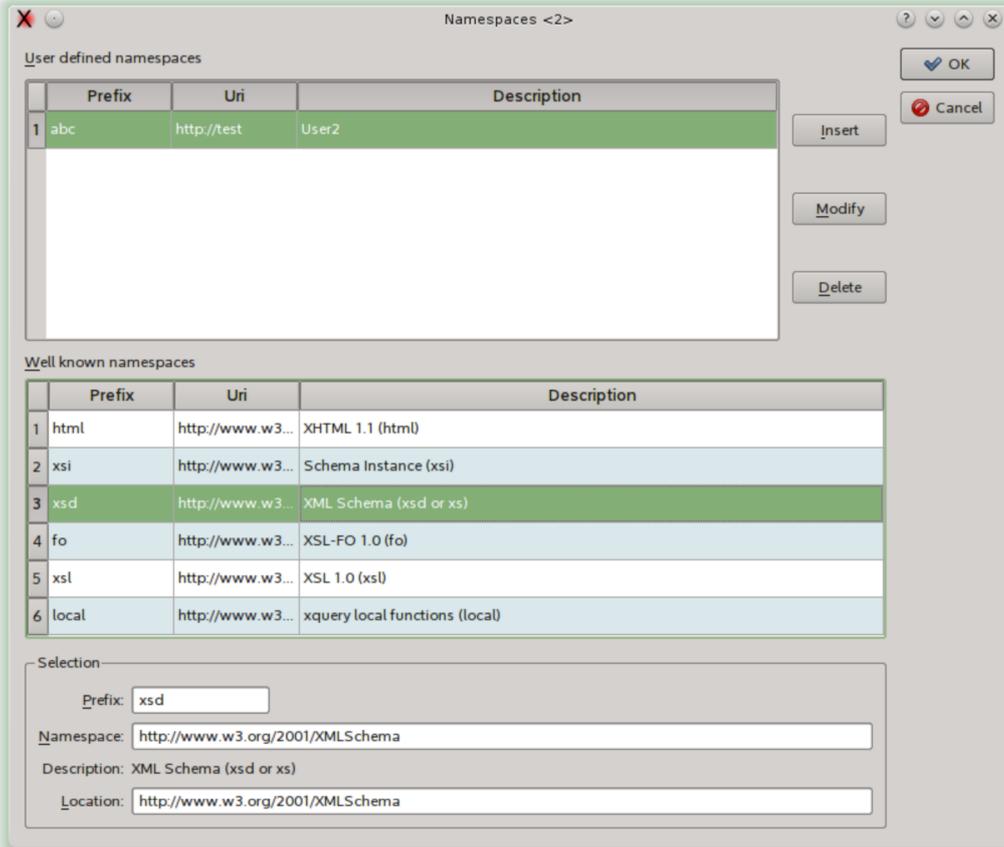
## Namespaces

In the edit element window it is possible to recall the namespaces management dialog.



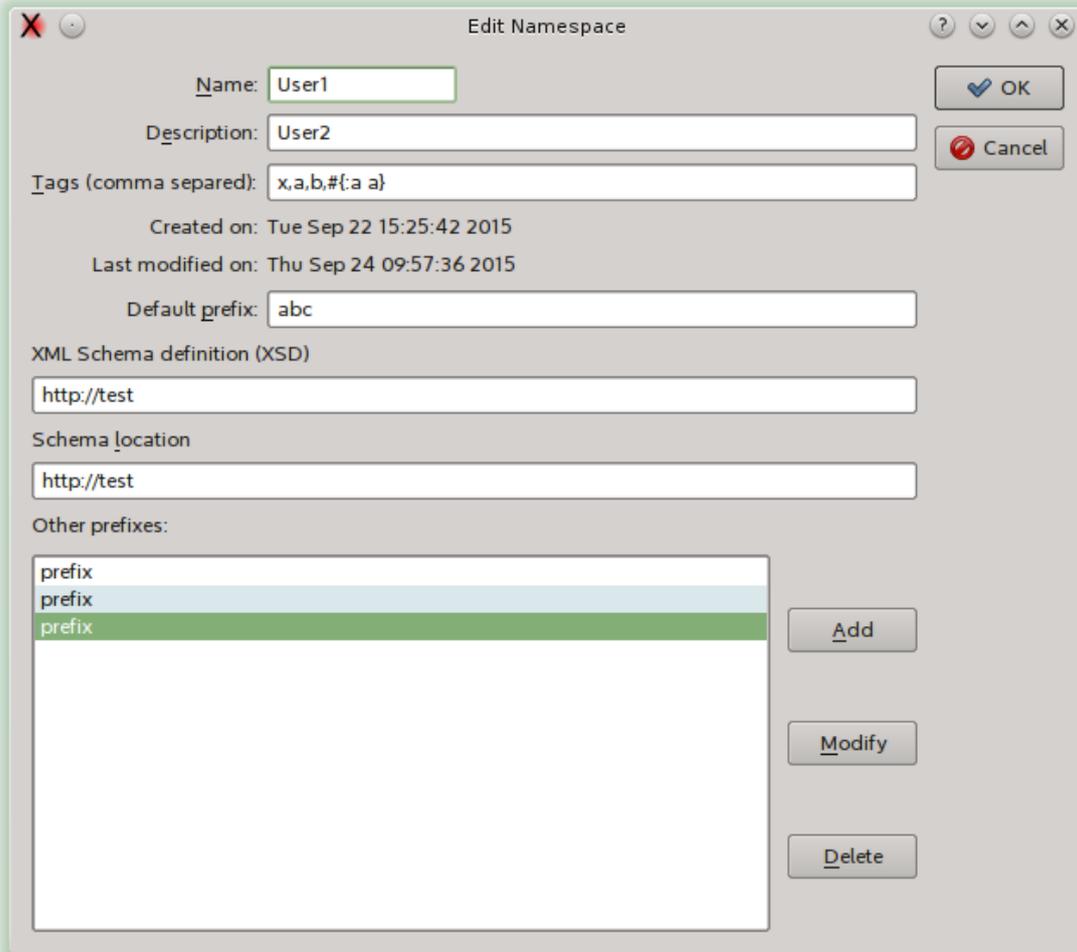
### *Namespace management dialog*

In this dialog it is possible insert namespace references recalling the namespace selection dialog.



### *Namespace list dialog*

In this panel it is possible to select a system defined namespace or choose a user defined namespace. The user defined namespaces are managed in a separate dialog, stored permanently and are available to successive sessions of QXMLEdit.



## Using the Bookmarks

You can mark the elements with bookmarks for easy retrieval and navigation.

To set a bookmark:

- select an element, then use the menu **Bookmarks > Toggle Bookmark** (CTRL-F2)

To remove a bookmark:

- navigate to the bookmark, then use the menu **Bookmarks > Toggle Bookmark** (CTRL-F2)

To remove all the bookmarks:

- use the menu **Bookmarks > Remove All Bookmarks**

To move to the next bookmark:

- Press F2

or

- use the menu **Bookmarks > Go to Next Bookmark**

To move to the previous bookmark:

- Press Shift-F2

or

- use the menu **Bookmarks > Go to Previous Bookmark**

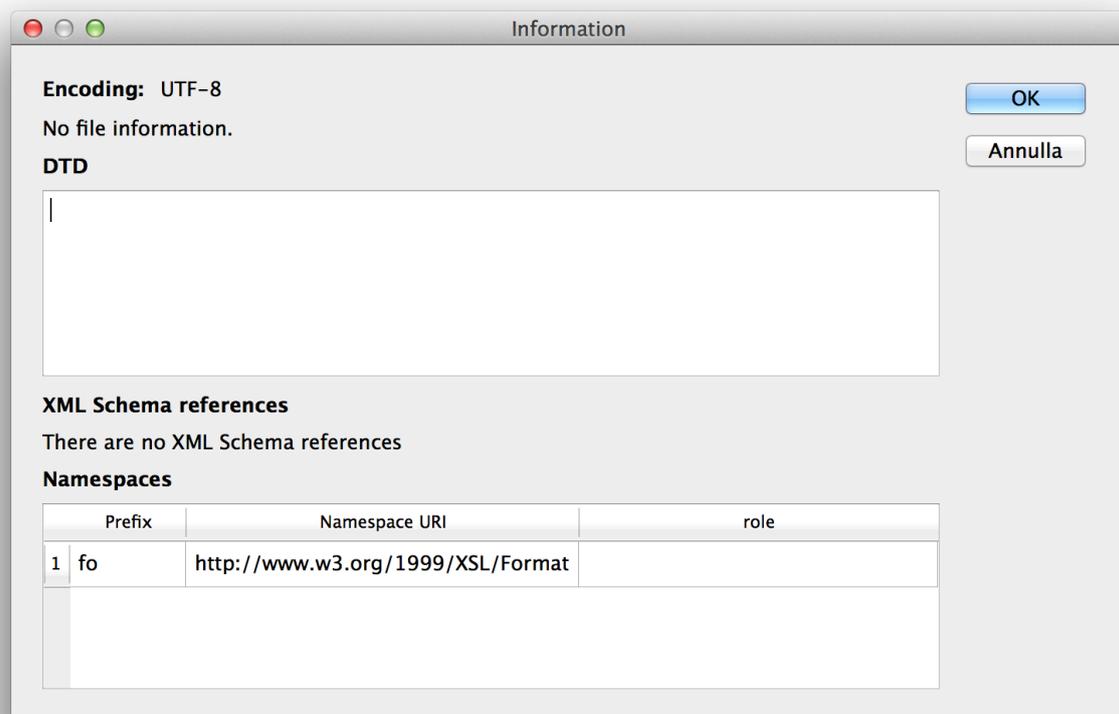
## Viewing and editing info and DTD

Basic information about the file can be shown using the menu **Edit > Info**

The information is relative to:

- Encoding.
- XML Schema references embedded into the file.
- Declared name spaces.
- DTD.
- File information.

Here an “Info” panel:



Please note that some namespace prefix can be used in more than one place with different references.

The “role” column will list well known namespaces.

The DTD of the document can be edited in this panel if the file is not read only. The manually edited DTD is not checked against syntax errors.

## Working with Base64 Coded Content

Sometimes the data of an element are text nodes or binary data base 64 coded. QXmlEdit can display decoded base 64 coded text near to the literal content in the main window, and permit you to edit directly the content.

To edit directly the base 64 coded text of an element:

1. select the element, then right click to make the context menu appear and select **Edit Inner base 64 Text**.
2. Write directly the text into the edit box of the panel that appears.

The text will be automatically encoded and decoded as needed.

## Working With Inner XML Content

Sometimes the data of an element is an XML payload registered as text, or a base 64 coded XML. QXmlEdit can edit directly both the types of contents.

To edit directly the XML data stored as payload of an element:

1. select the element, then right click to make the context menu appear and select **Edit Inner XML**.
2. Another QXmlEdit Window will appear; you can use it to edit the inner XML.

The XML will be automatically encoded and decoded as needed.

To edit directly a base 64 coded XML used as element payload:

1. select the element, then right click to make the context menu appear and select **Edit Inner XML Base 64 Coded**.
2. Another QXmlEdit Window will appear; you can use it to edit the inner XML.

The XML will be automatically encoded and decoded as needed.

## Inserting an XML prolog on document creation

You can insert an XML prolog XML using the menu XML > Insert prolog. You can insert automatically a prolog on document creation using an option in the configuration panel. In the panel you can also select the encoding to be used.

## Formatting a document

QXmlEdit can use a processing instruction to store and retrieve formatting settings for indentation, attributes display and so on.

If the processing of the instruction is enabled, the settings are read and applied at the next saving. If some formatting options is changed and the instruction exists in data, the instruction is automatically updated before saving.

The instruction processing can be enabled in preferences, XML section.

To insert a formatting processing instruction use the menu item **Edit>Formatting>Add**

**Formatting Info.** To remove it us the same menu, and the **Remove Formatting Info** item.

To enable or disable the management for a single document, use the formatting panel with the toolbar or the menu item **Formatting>Formatting...**

The preferences panel has an option to automatically insert the instruction at each document creation.

The processing instruction is transparent to XML management, however should you want to avoid it, you can disable it in preferences panel.

### Formatting Presets

In the menu **Edit>Formatting** there are some common (hopefully) presets. The name of the items gives a clue about its settings.

## Validating an XML Document

An XML Schema validation option is available, but it uses the Qt XML validation facility, that is rather limited at the moment. The validation is made by default using schema references contained in the XML document, if any. You can use an external XML Schema file; after this assignment the validation is performed using the assigned file until an explicit command to use the document references is given.

If an external schema file is used, its path is visible In the tool tip of the toolbar “Validate” command.

To validate using the last assigned schema:

- select the menu **XML > Validate**

To validate a document using self referenced XSD, if any:

- select the menu **XML > Validate Using Document References**

To use a different an XML Schema file:

- select the menu **XML > Validate Using New Schema File**

**NB:** the validation option is available only if compiling at least with the Qt 4.4. It uses the Qt XML validation facility, that is rather limited at the moment.

## Counting Children Elements and Measuring Their Size

It is possible to display the sum of the number of elements children and the element total size, calculated on the canonical element form ( i. e. counting stating and ending tags ). The size calculated is only an approximation of the real value, given that there are many modes to write the same XML data.

To activate the option:

- select the menu **View > Show Element Size**

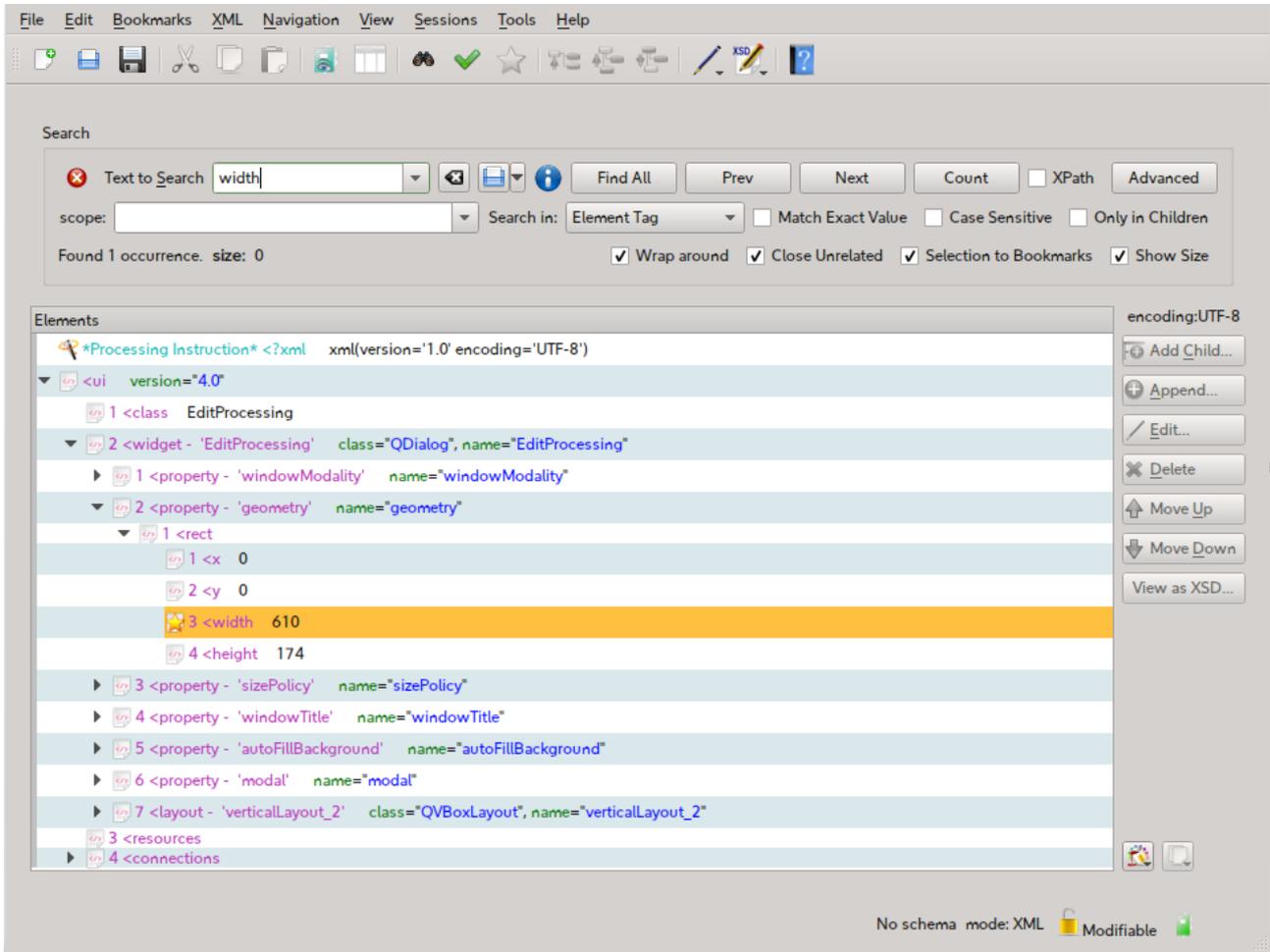
## Finding Text

It is possible to execute searches in the editor. To open the search panel:

- use the **Edit > Find** menu

or

- press Control+F



In the panel you can do the following operations:

To start a search:

- insert the text in the edit box and press Enter key or press the Find button

To limit the search to a particular kind of XML components, use the **Search In** option box:

- All the types of components
- Element tags only
- Attribute names only
- Attribute values only
- Text nodes
- Base 64 coded text

The elements that match the search rules will be highlighted, but you can act on the display in the

following manners:

Name of the Option	Information
close unrelated	close all the branches that does not contain any occurrence of the search text.
selection to bookmarks	adds all occurrences found to the bookmarks collection.
match exact value	executes the search of the exact value of text (opposed to regard the text as a substring)
case sensitive	the search will be case sensitive
find all	mark all the occurrences of the search pattern with a background pattern to ease the reading, elsewhere the search will stop to first match.
only in children	the search can be limited only to children of the selected item or to the whole tree.
scope	can limit the search to a set of elements or attributes. This field can contain a path in XPath like syntax. For example to search a value in the "id" elements children of "resource", the field must contains "resource/id". If the search has to be performed only on "name" attribute of the "window" element, write "window/@name". To include any elements between "window" and "widget" tags, simply omit it as in "window//widget"
XPath	Activates the XPath syntax (see below for details). If unchecked the text entered in the search box is used as pure text.

The number of matches is reported in a box at the bottom of the search panel.

It is possible to find the next occurrence starting the search from the selected item. You can use the command buttons in the panel or activate it using the following command:

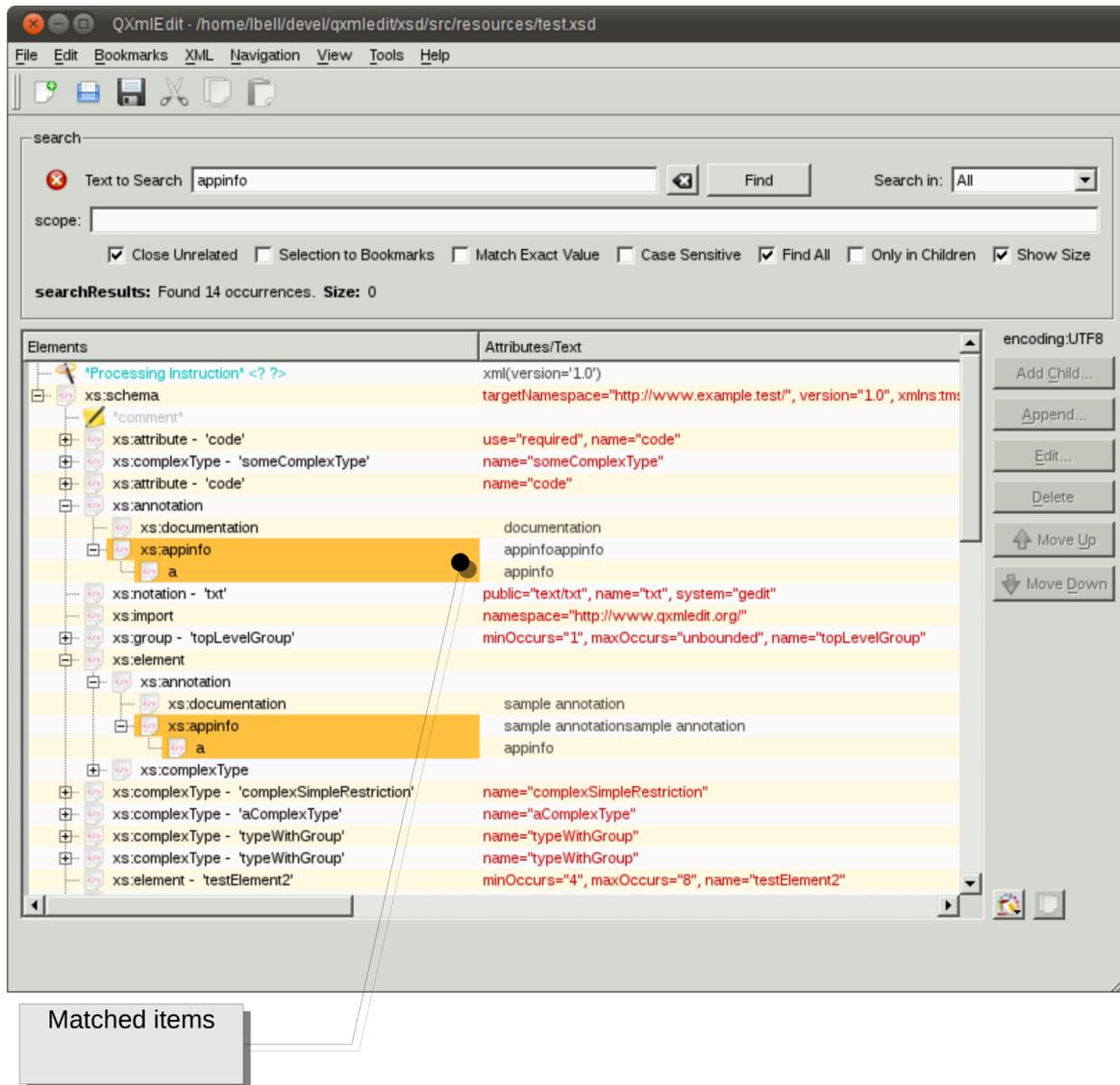
- use the **Edit > Find Next** menu

or

- Use the key F3

### Search results

The matching items are highlighted and, in combination with other options, non matching items are closed.



## XPath syntax

In the panel you can do the following operations:

To start a search:

- select the “**XPath**” check box
- insert the text in the edit box using XPath syntax and press Enter key or press the Find button

A small note will indicate the XPath mode activation.

If “**Only in Children**” check box is selected the root of the search is the current element, if not the search starts at the document root.

Some of the options available to the text search are hidden because their operation can be carried on using XQuery and XPath.

The elements that match the search rules will be highlighted, but you can act on the display in the following manners:

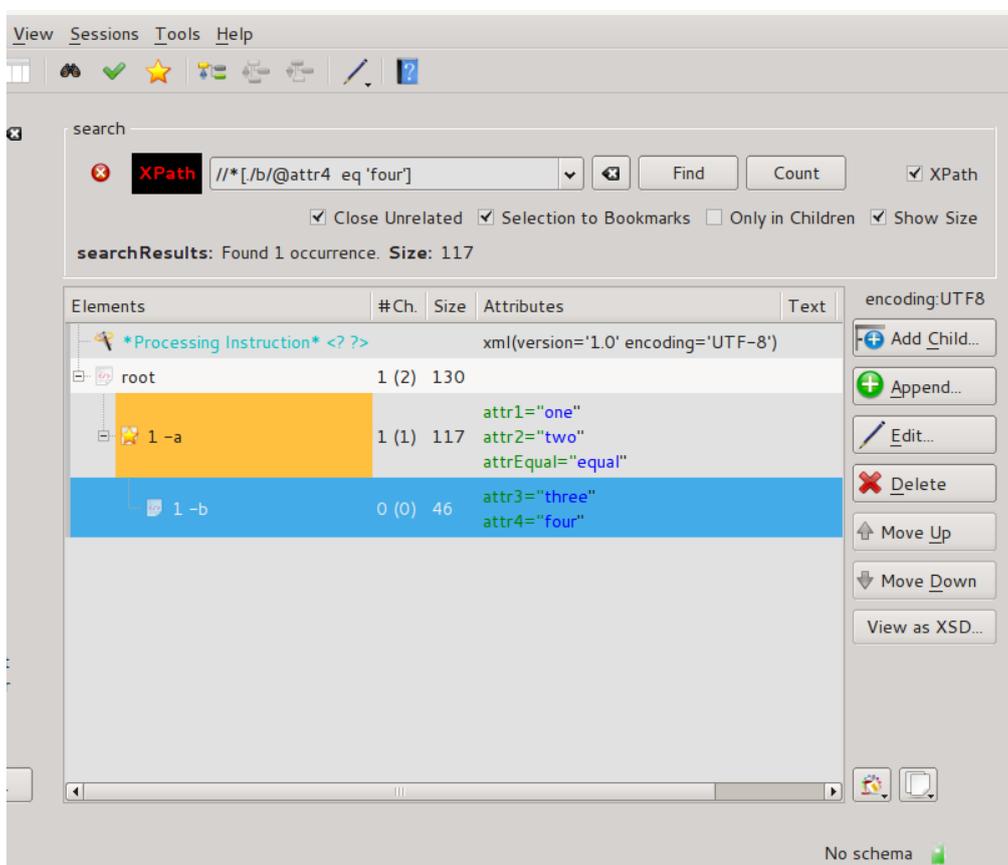
Name of the Option	Information
close unrelated	closes all the branches that does not contain any occurrence of the search text.
selection to bookmarks	adds all occurrences found to the bookmarks collection.
only in children	the search can be limited only to children of the selected item or to the whole tree.
scope	can limit the search to a set of elements or attributes. This field can contain a path in XPath like syntax. For example to search a value in the “id” elements children of “resource”, the field must contains “resource/id”. If the search has to be performed only on “name” attribute of the “window” element, write “window/@name”. To include any elements between “window” and “widget” tags, simply omit it as in “window//widget”

The number of matches is reported in a box at the bottom of the search panel.

### Search results

The matching items are highlighted and, in combination with other options, non matching items are closed.

The following screen shot shows a search using XPath:



### Browsing the search results

You can open a search results panel using the button “**Open panel**” in the find panel.

The new panel will show only the search results, giving the possibility to browse directly the XPath

queries elements or the text search results.

## Replacing text

Replace uses the search feature to find text that can match and replace it all at once or one at time. Replace is triggered by the menu **Edit > Replace**.

### Replace panel



The following commands are available:

Command	Behavior
Replace All	Replaces all the occurrences of the search pattern.
Replace and find next	If there is a selection, a replacement is applied if there is a match; then the next element that matches is selected. If no selection was existing, the next match is selected.
Replace and find previous	If there is a selection, a replacement is applied if there is a match; then the previous element that matches is selected. If no selection was existing, the previous match is selected.
Skip and find next	If there is a selection then the next element that matches is selected. If no selection was existing, the next match is selected.
Skip and find previous	If there is a selection then the next element that matches is selected. If no selection was existing, the previous match is selected.

**Note:** replace operation cannot use XQuery search.

## Comparing Files

The purpose of the comparison as implemented is to give a compact view of changes, comparing the current editor content with another file. The sense of the comparison is to show what transformations are needed to the chosen file to become the current loaded one.

To compare a file against the current:

1. select the menu **File > Compare with...**
2. choose a file

In the panel that opens you have the following information:

- The current file in XML format with elements and attributes that are different between the files with highlighted colors (Synthetic View).
- A structure comparison side by side of the two files (Analytical view).
- The list of the differences with the path of elements (Difference list).

The following options can be set:

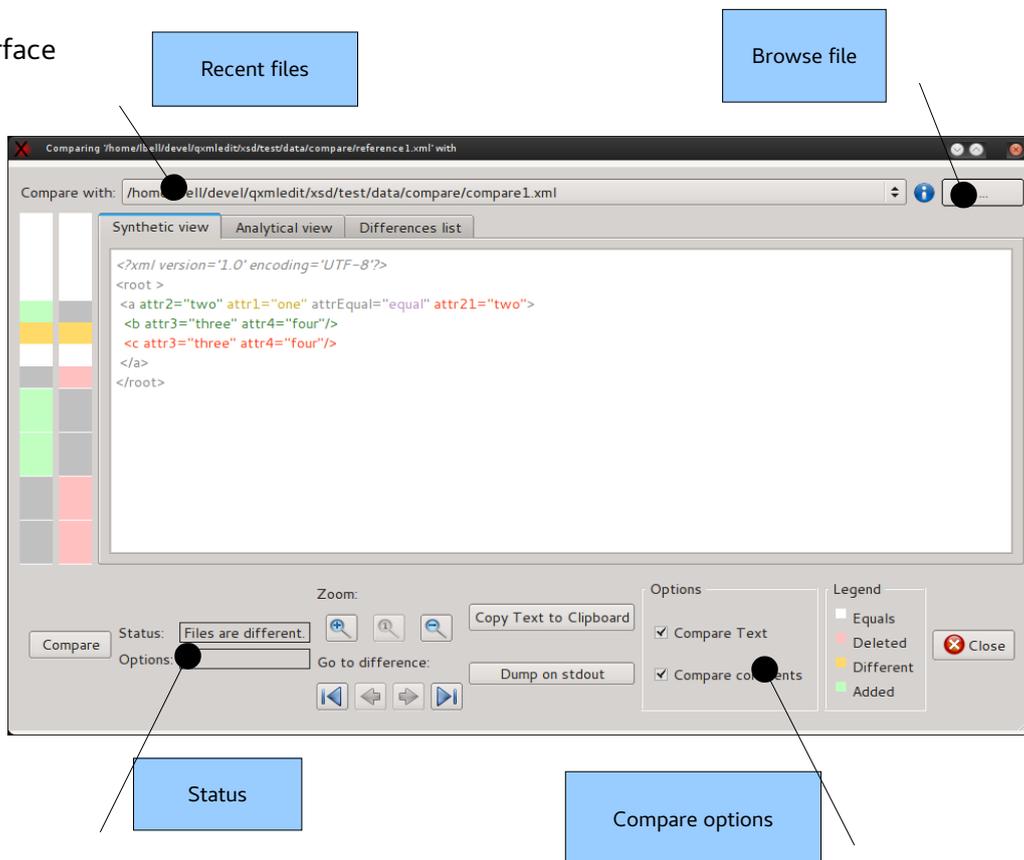
Option	Meaning
Compare text	Compare even the text nodes
Compare comments	Take into account the comments
Denormalize EOL	In CDATA nodes do not transform the CR LF sequences into LF as stated by specs.

In the panel that opens you have the following options:

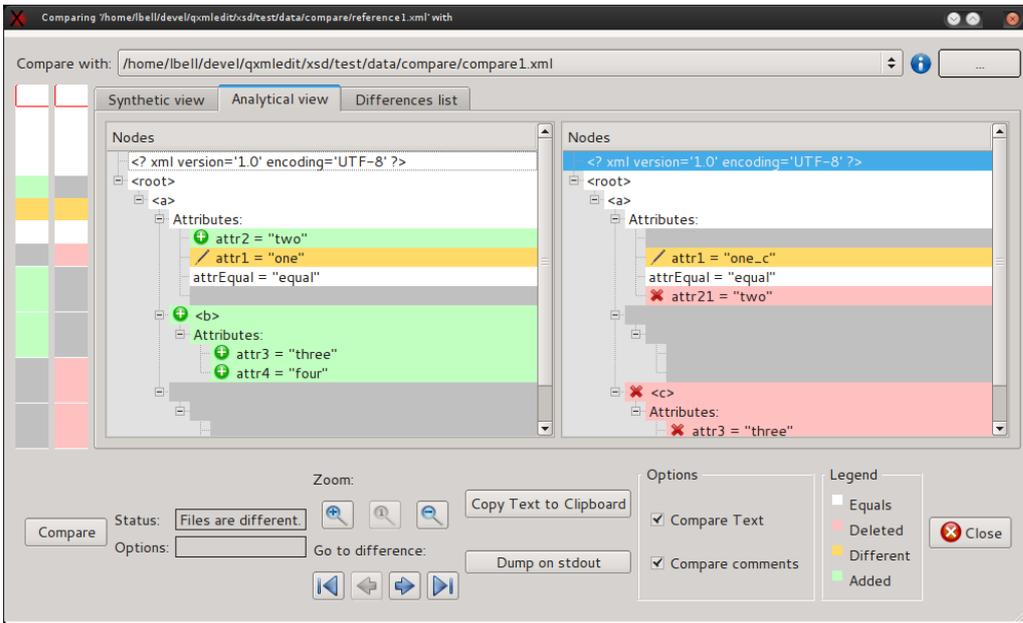
- Compare only elements or comments and mixed content too.
- Copy the comparison result to the clipboard.
- Retry the comparison, after changing some option.

To start a comparison, load a recent file from the relative combo box or select another one using the browse button.

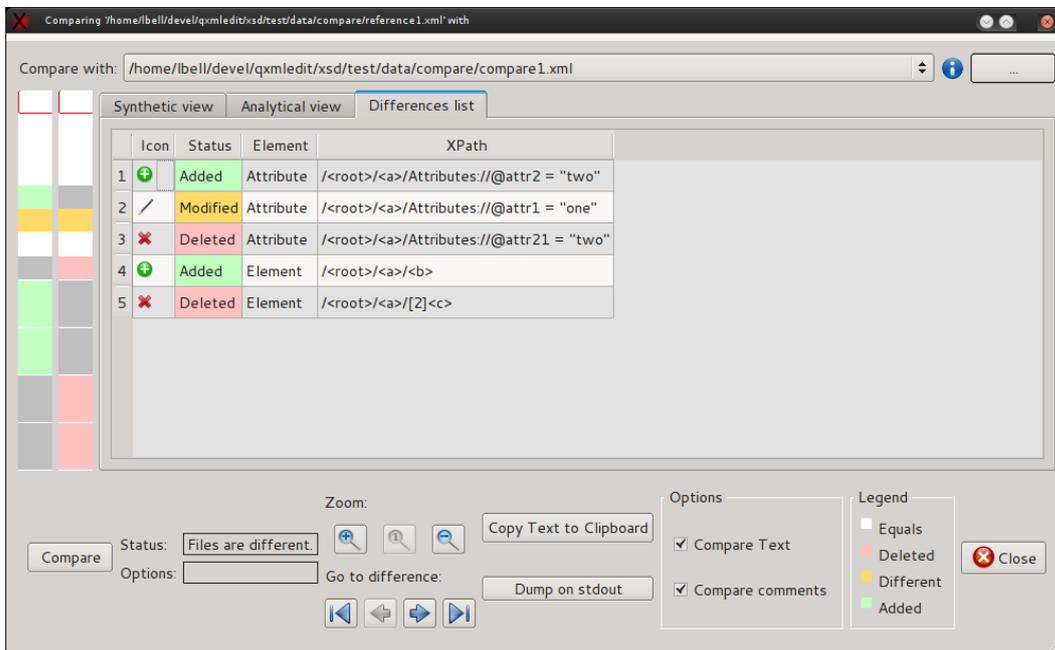
User interface



Analytical view



## Differences list



## Anonymizing data

The menu **Special > Anonymize** recalls a dialog to anonymize the data contained in attributes and text of the XML.

This feature can be useful if the XML file should be sent to technical staff for debugging purposes avoiding the disclosure of personal information.

Using this feature the text and the attributes can be altered using a fixed character or using a different letter for each input character by preserving the case of the original letters.

The original data structure is preserved replacing only letters and digits with altered text, preserving the case of the letters and the reserved XML attributes. The tags and attribute names are not changed.

## Data and business rules

Often the files contain not only personal information, but also data for business rules. Modifying all the contained text or attributes will end in a useless file transporting no information at all.

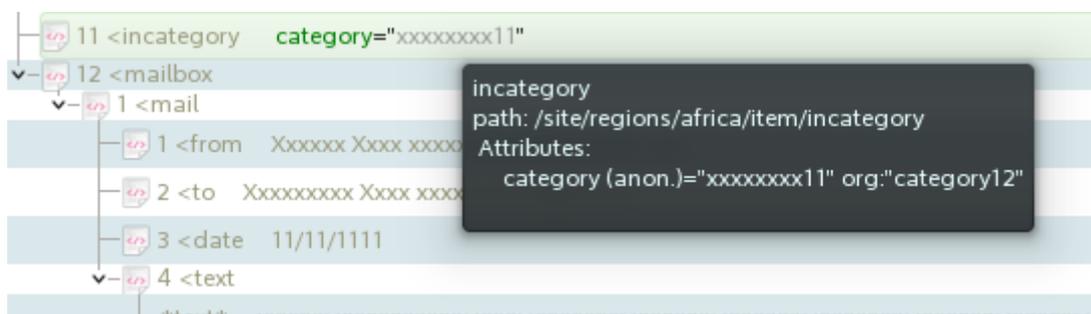
To avoid that result, manipulating the parameters that belong to the business rules, two features can be used:

- A filter to avoid the anonymization of values consisting of less than 4 characters (the length of many codes).
- The use of exceptions.

The exceptions can be bound to any element, attribute or text node and force the subject to be anonymized or not; if anonymized, a user defined value can be used to replace the original one.

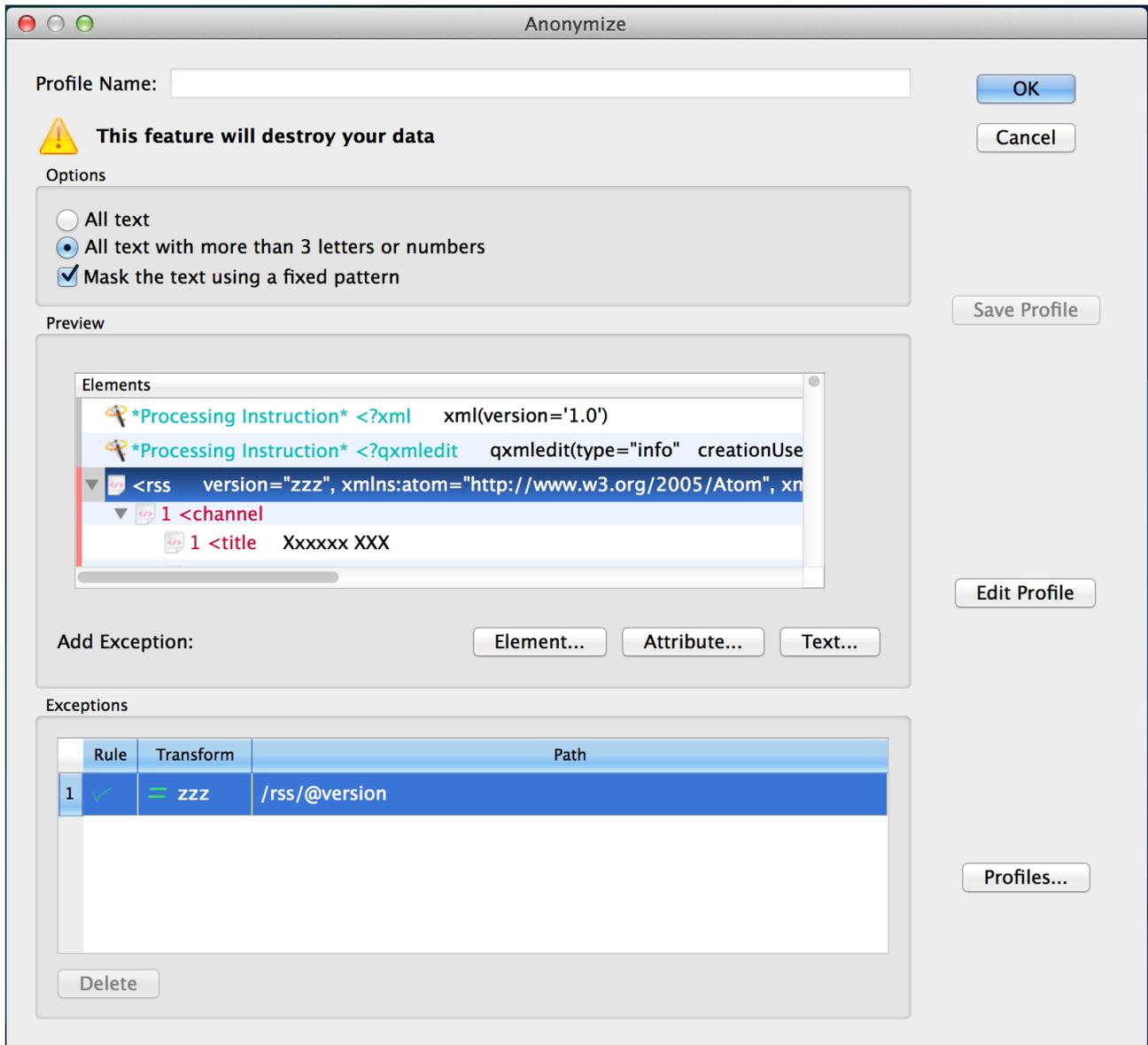
The exceptions can be hierarchical and extended to the children of the subject. It is possible to isolate data from business rules from personal data. When applying more expressions, the more specific wins; for example if an element has been forced to be anonymized with children, and one of its children has another exception of reverse meaning, the child exception is applied.

Each time an exception is modified, the anonymization operation results are shown in a preview pane. The original values can be seen in the tooltips of the elements of the preview.



Since creating rules and exceptions is a time expensive process, it is possible to save them using a profile and recall them later in other anonymization processes.

## Options



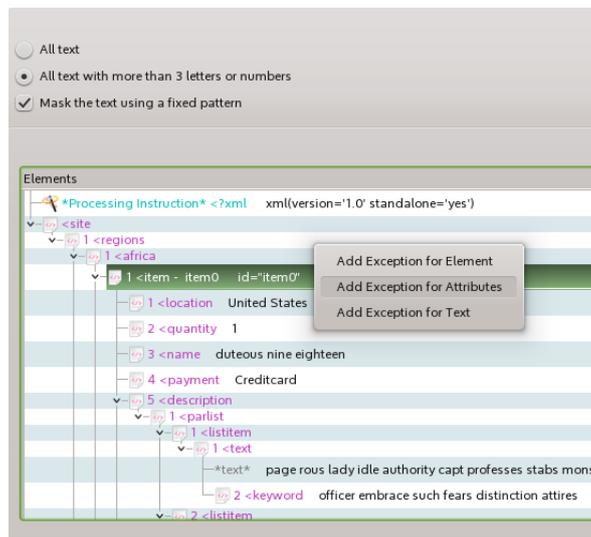
The options are:

- All text: all the text will be changed.
- Text with more than 3 letters or digit: only the text that is composed by more than 3 letters or digits will be changed. This to skip codes that usually have a limited length.

The check box “fixed pattern” determines how the text is transformed.

- If checked: letters are replaced by 'x' and numbers by '1'.
- If unchecked: the letters and the numbers are replaced by a succession of letters or numbers in alphabetical order.

The exceptions can be entered selecting the element and using the contextual menu.



Information in the panel:

Name	Information
Profile Name	Name of the profile (read only)
Options	Options to anonymize data: <ul style="list-style-type: none"> <li>anonymize all the text</li> <li>anonymize only the text with more than 3 letters or numbers</li> </ul> As an option it is possible to use a fixed pattern to replace the text.
Preview	It shows the final result of the operation on the current data
Add Exception	You can add exceptions when an element is selected in the preview.
Exceptions	The list of the exceptions
Save Profile	Saves the current profile data. If the profile has not been created, a creation panel will appear.
Edit Profile	Edits the profile data.
Profiles	Manages the profiles.
Import	Import a list of exception in CSV format (see below for details)
Export	Export the exception list to the clipboard in CSV format

To insert an exception:

1. Select an element in the preview pane
2. Use one of the buttons in the “**Add exception**” row

To remove an exception:

1. Select an exception in the exception list
2. Press “**Delete**” in the exception group.

To save the exception list in a profile:

- Press “**Save profile**”

If the profile has not be created, a profile creation panel will appear.

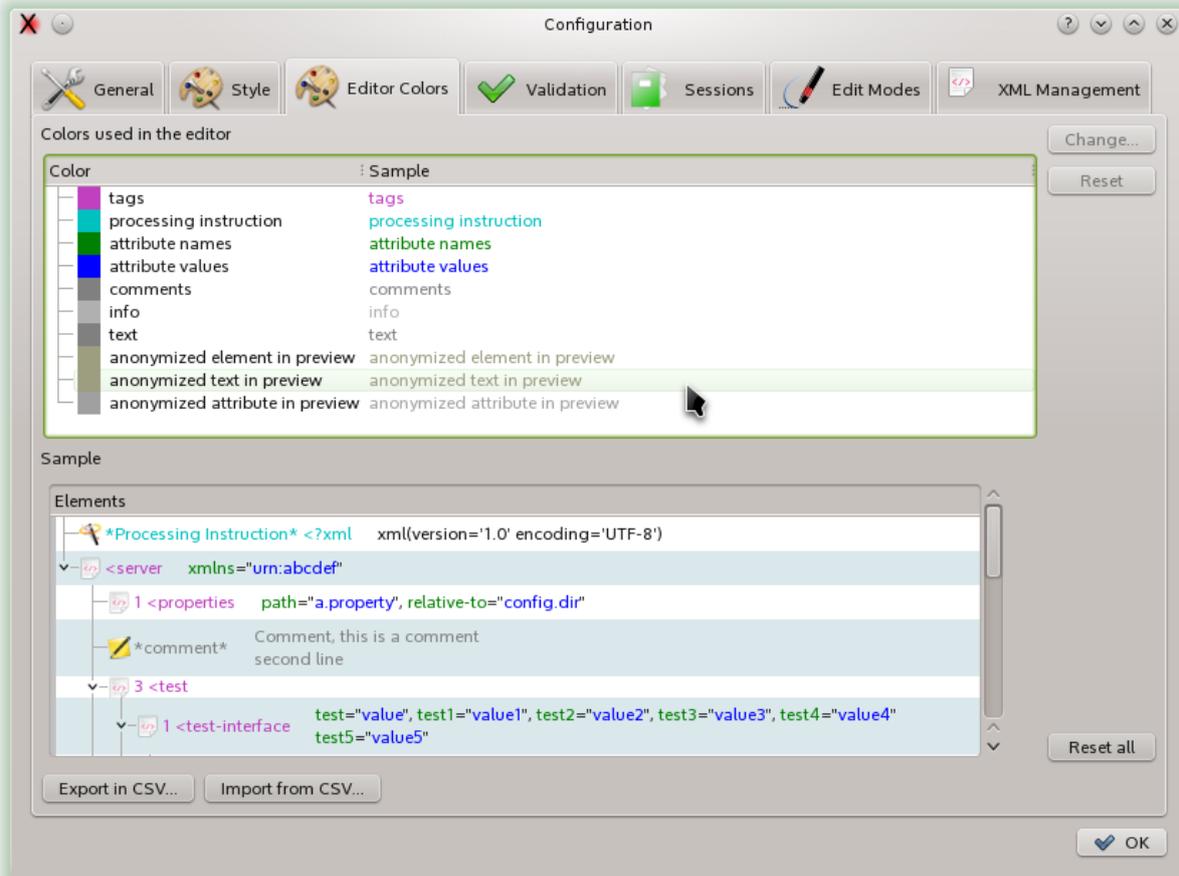
To manage profiles:

- press “**Profiles...**”

A list of profiles, grouped by tags will appear. Selecting a profile and exiting with “**OK**” will load the profile in the caller dialog. It is possible also creating and deleting profiles.

### Configuring the anonymization

The colors of anonymized items can be configured using the “Editor Colors” panel.



### Format of the CSV for importing and exporting the exceptions

The CSV fields used to exchange the exception list are:

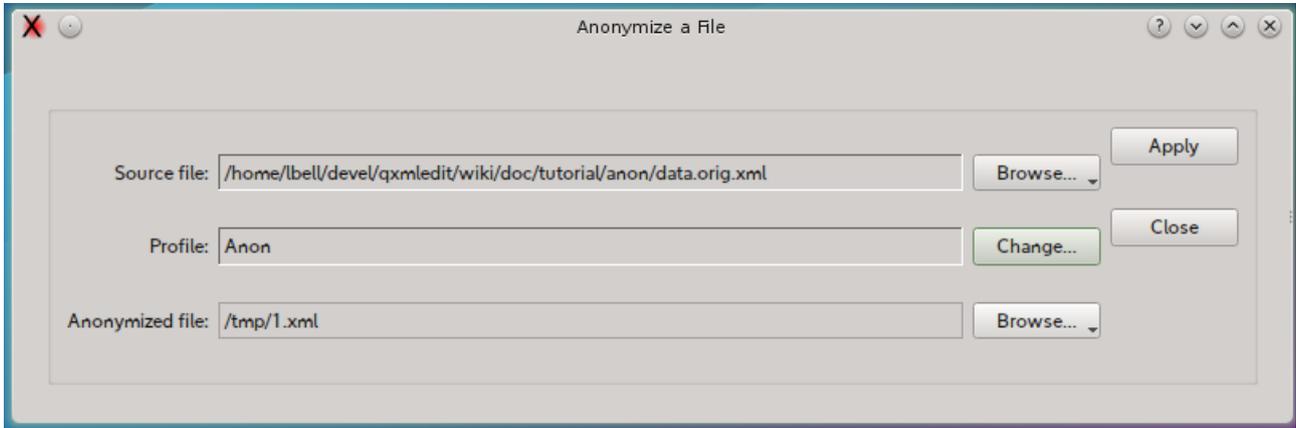
1. The XPath of the exception.
2. The inclusion criteria as integer:
  - I. 0: exclude (default)
  - II. 1: exclude with children
  - III. 2: include
  - IV. 3: include with children

3. Type of data mask used:
  - I. Use default.
  - II. Use a fixed value.
4. A string value if the fixed value replacement is chosen.
5. A numeric flag for processing namespaces.
  - I. 0: do not process namespaces
  - II. 1: process namespaces

## Anonymizing a file

When a profile has been created in the interactive anonymized dialog the profile can be used to process a file without loading it into the editor.

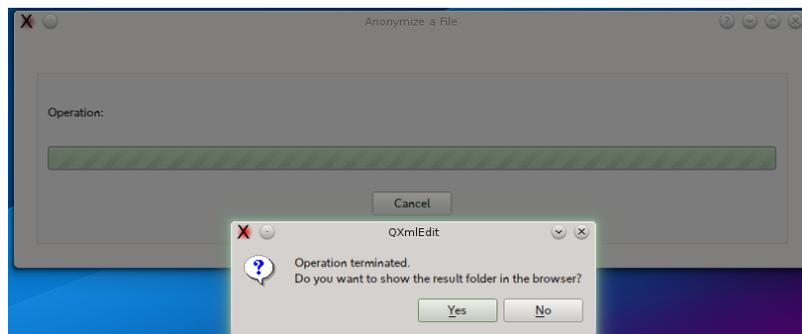
To activate the function use the menu **Tools > Anonymize File...**



In the anonymize panel you have the following options:

Name	Option
Source file	Name of the source file to anonymize. Use the " <b>Browse...</b> " button to select it
Profile	The name of the anonymizing profile. Use the " <b>Change...</b> " button to select it among the previously saved profiles.
Anonymized file	Name of the anonymized file. Use the " <b>Browse...</b> " button to select it. Note that this function will overwrite the file.

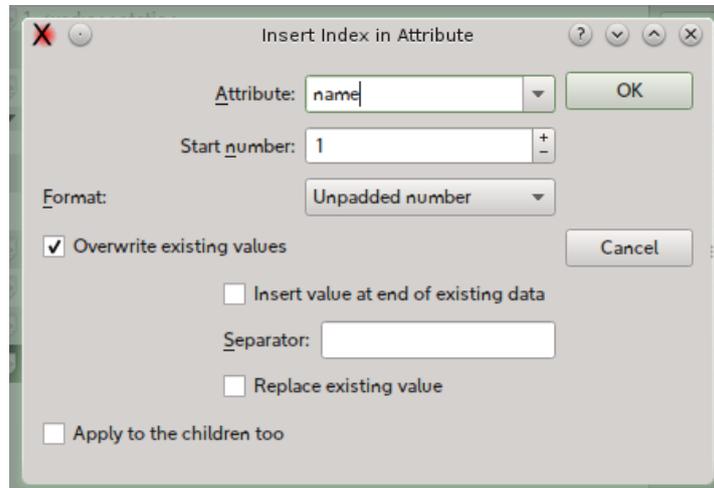
At the end of the operation the destination folder can be opened by the panel.



## Filling data

It is possible to insert a progressive number or alphabetic sequence selecting an element and triggering the menu “**Tools > Fill Serie...**”. A dialog box will allow to choose the attribute name that will contain the progression index.

This is the dialog and below each option is explained:



Options:

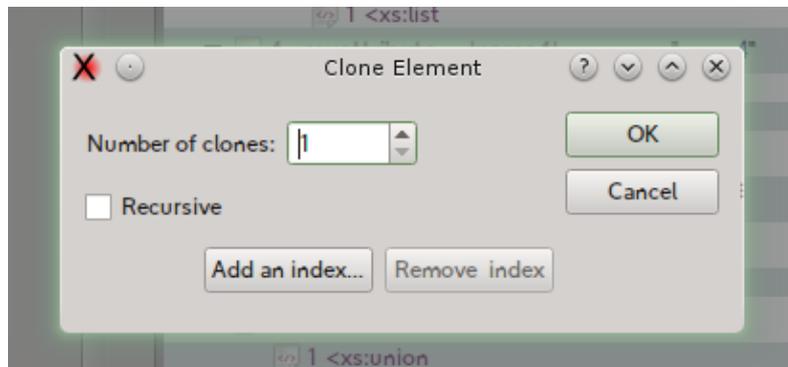
Name	Information
<b>Attribute Name</b>	Name of the attribute that will contain the progression index. If the attribute does not exist in the element, it will be created.
<b>Start number</b>	The index of the series fill. For the alphabetic enumeration '0' is equivalent to 'A'.
<b>Format</b>	The format of the index. The possible predefined formats are: Unpadded number (1,2,3) Padded number ( 01, 02, 03) Padded alpha (AA, AB, AC) Unpadded Alpha (A, B, C)
<b>Overwrite existing values</b>	This option allows the modification of an existing attribute. Without this authorization, the existing attributes are left untouched.
<b>Insert values at the end of existing data</b>	If the attribute for the index is existing and overwrite is enabled, the index will be written before or after the current data.
<b>Replace existing value</b>	Replaces an existing value of the attribute.
<b>Separator</b>	This separator is used when inserting the index into existing data.
<b>Apply to the children too.</b>	Apply recursively the index. Each series restarts from 0.

## Cloning Elements

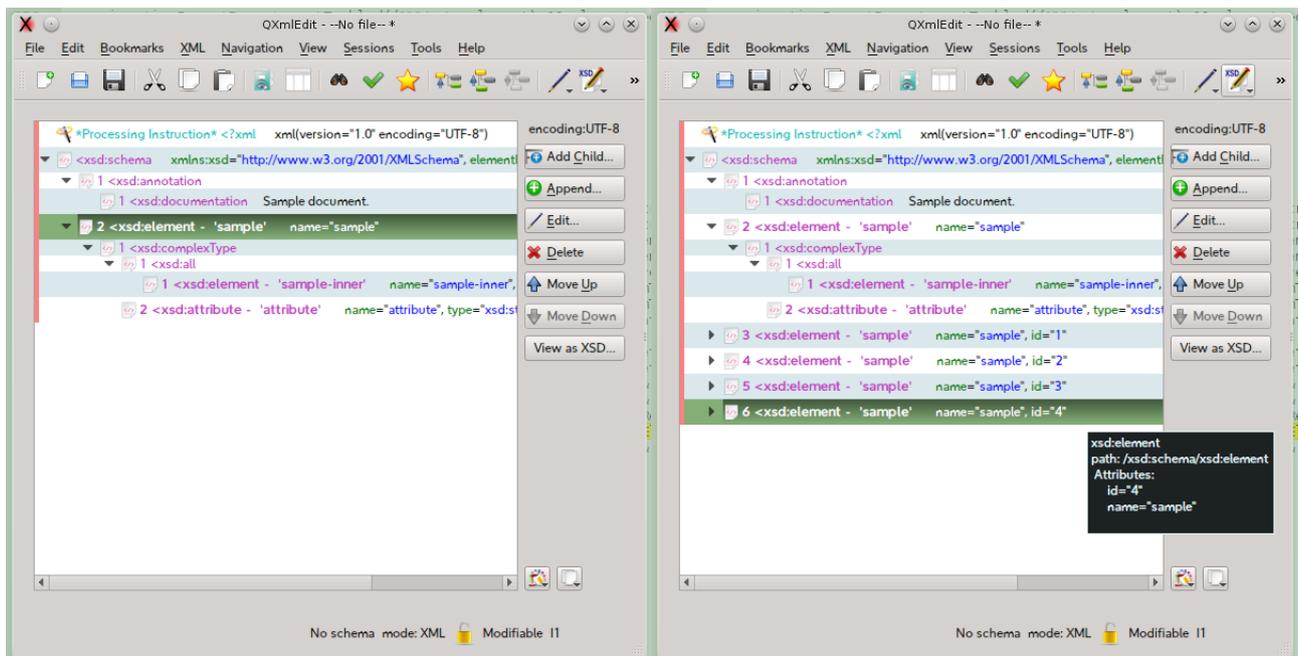
Cloning or replicating elements in the editor using an existing sample is activated by using the menu **Tools > Clone Elements** in the 0.9.2 version. Any element can be used as a sample except the root one.

The only options available to this feature is if to clone only the selected element or even its children recursively, the number of cloned objects and the use of a counter.

The counter that can be insert in an attribute, existing or created while cloning, is the same that can be recalled by the “Fill Serie” dialog,



This picture shows the result of the clone operation with the insert of an index using the “id” attribute:



## Working with Snippets

Snippets are pre-configured XML fragments that can be categorized and inserted in the main XML text. A snippet example can be a resource reference in a J2EE component.

To create a snippet:

select an XML element, then:

- select **Transform in Snippet** from the context menu

or

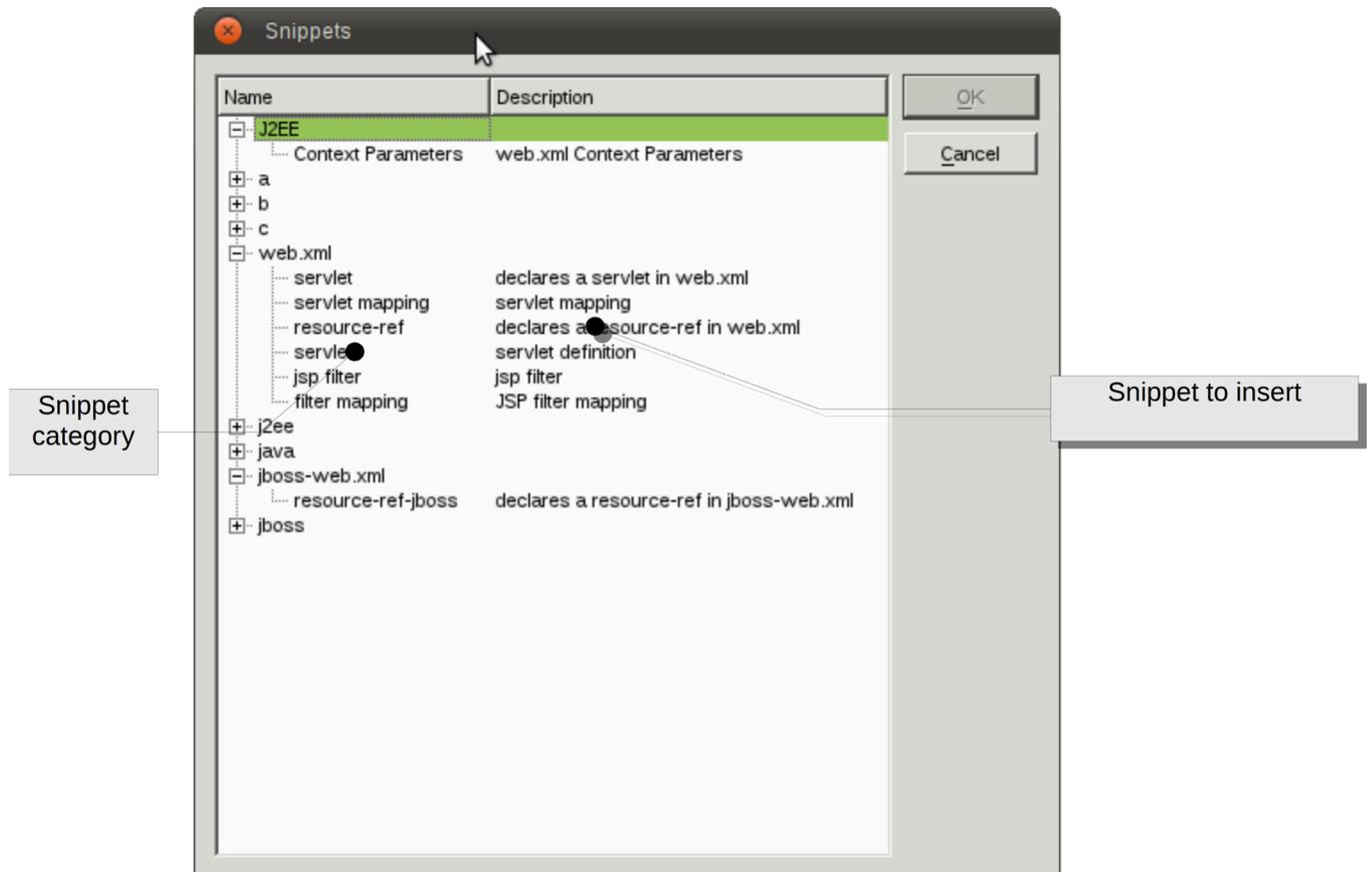
- select the **XML > Transform in Snippet** menu

To insert a snippet in the XML you are editing:

select an XML element that will be the father of the snippet, then:

select the menu **XML > Insert Snippet**.

in the panel that will appear chose the snippet category from the left, then choose the snippet to insert.



To edit the snippets and their categories:

1. Select the menu **Edit > Configure Snippets**

2. In the panel that will open, insert, modify or delete a snippet at time.

To assign snippets values:

In the edit snippet panel you can supply the following informations:

Name	Information
Name	The name of the snippet
Description	The description that will be shown when choosing a snippet
Tags	Set of comma separated values that will be used as snippet categories. A snippet can have more than one category

Dialog box titled "Edit XML Snippet" with the following fields and values:

- Name:
- Description:
- Tags (comma separated:):
- created on: Fri Sep 10 21:52:41 2010
- last modified on: Fri Sep 10 21:53:55 2010
- Snippet: 

```
<servlet-mapping>
<url-pattern>/ADDRESS</servlet-class>
<servlet-name>SERVLET</servlet-class>
</servlet-mapping>
```

Buttons: OK, Cancel

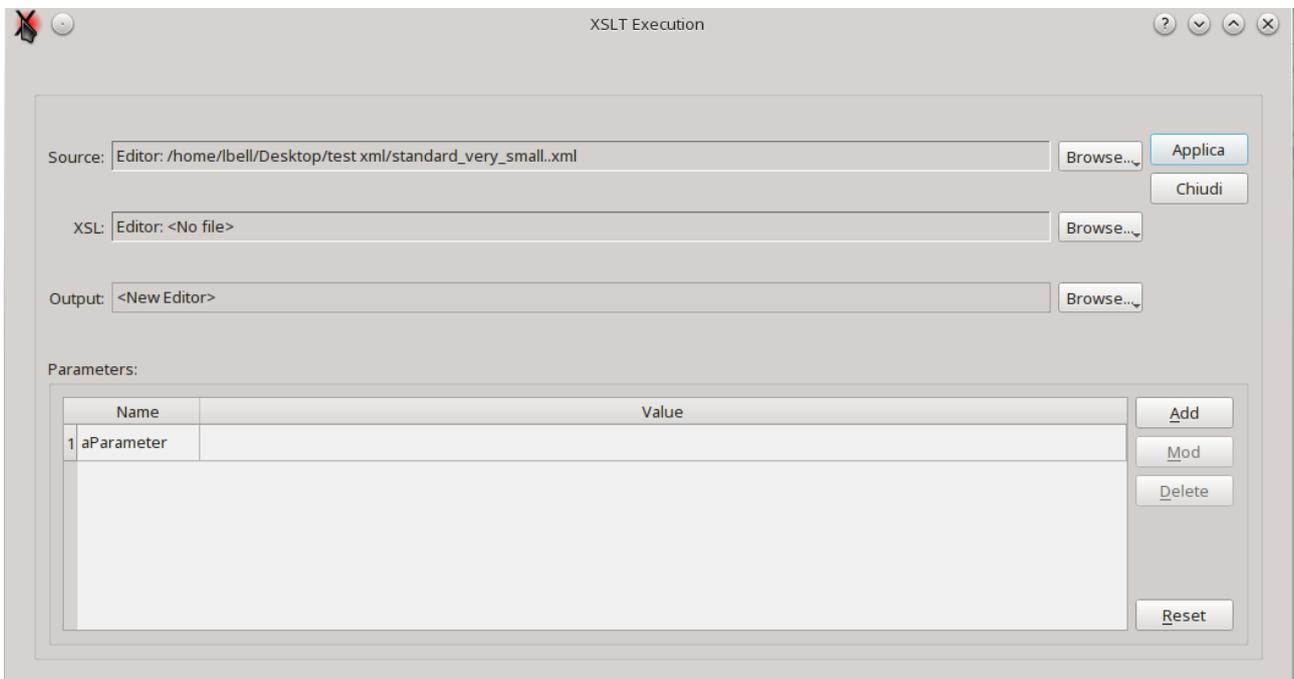
## Executing XSL style sheet

You have three options:

- Executing a XSLT using the current editor content as source.
  - Use the menu **Special > XSLT > Execute XSLT using this as source**.
- Executing a XSLT using the current editor content as Style Sheet.
  - Use the menu **Special > XSLT > Execute XSLT using this as source**.
- Using a panel to specify the source, the style sheet and the output.
  - Use the menu **Tools > XSL Executor**.

### XSLT Execution panel

Using the menu **Tools > XSL Executor** brings a panel from which you can specify all the details of the XSL execution.



In this panel you can specify:

- The source XML.
- The XSL template.
- The output.

When selecting the XSL template, the parameters, if any, are shown in the list.

If the internal engine has been selected as the XSL engine, you can choose the editors as source, if the external engine has been configured, only files are allowed.

Note: the support to XSL-T 2.0 in Qt is currently not complete.

You can use SAXON or another Java jar with same command line syntax (not part of QXmlEdit) if you configure it in preferences.

## Metadata

### *Introduction*

The metadata records carry information about the data in which it is located without modify the data itself. QXmlEdit supports descriptive metadata implementing it using an XML processing instruction whose target is “qxmledit”.

### *Metadata types*

There are two types of metadata: updatable and static.

### **Updatable Metadata**

This kind of metadata carries information about the current state of the document and it can be automatically updated at each document modification if the relative option is enabled. The data that can be automatically updated consists of: document creation user and date and the document revision.

### **Static metadata**

The values contain semantic information and cannot be automatically updated without explicit user intervention. Some data are predefined, others can be inserted by the user when needed.

### *Metadata implementation*

The metadata lives on the root level of the XML document. Each metadata record consists of a Processing Instruction composed by pseudo attributes. The first attribute is always “type” and denotes the metadata role. The updatable metadata is composed by pseudo attributes that live all together in a single “info” type. Other metadata uses a single processing instruction having the attribute “value” for the real data. No entities will be resolved in a metadata pseudo-attribute except the characters “double quote”, “single quote”, “greater than” and “minor than”, that are escaped using usual XML entities notation.

### *Operation*

#### **Creating a document**

The updatable metadata record is inserted manually via menu command. The editable ones are inserted using a command that insert the updatables ones too.

#### **Reading a document**

No metadata is automatically inserted on read.

## Saving a document

When saving, if the option is active, the updatable data are modified. If the metadata record does not exist in the XML document, it will be inserted if the user gives permission. If only partial compiled metadata exists, QXmlEdit will complete it. Other unknown metadata information in the same processing instruction will be preserved for future compatibility. If no creation data exists, the data will be taken from the file data if available, else they will be created anew.

## Editing metadata

The metadata can be inserted in an XML document using a menu command or by using a dialog. The user defined metadata can be handled using the dialog user interface. Malformed data inserted by hand will be overwritten.

## Metadata summary

### Updatable Metadata

Name	Value
creationUser	The login name of the user
creationDate	The date of the creation
updateUser	The login name of the user
updateDate	The date of the modification
revision	A progressive value incremented at each writing operation

### Static Metadata

The values have no special meaning respect to the user defined metadata, however they can be presented to the user in a configuration dialog.

Predefined data:

Name	Value
project	
name	
copyright	
domain	
version	

### Editing metadata

Using the menu Edit > Edit Metadata you can open the metadata edit panel. Only updatable metadata whose check box is selected will be written into the document.

**Descriptive Metadata**

Updatable Metadata

**Creation date:** 2014-01-19T16:15:33

**Creation user:** lbell

**Update date:** 2014-01-19T16:15:33

**Update user:** lbell

**Revision:** 0

**Metadata model:** 1.0

Other

**P**roject: my project

**N**ame: name of the file

**C**opyright: (c)

**V**ersion: 1.0

**D**omain:

OK

Annulla

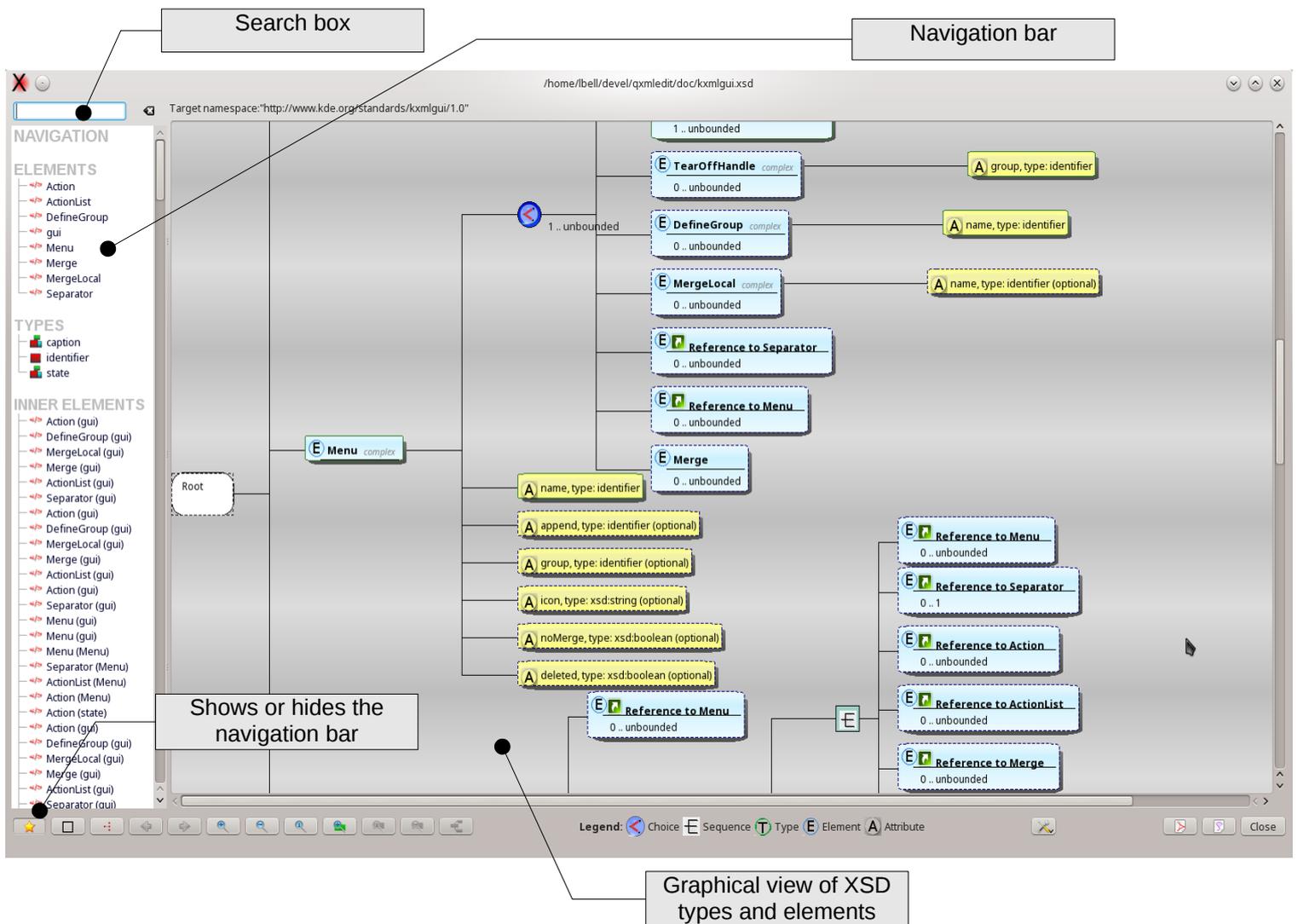
## Working With XML Schema Files (XSD)

When the file in editing declares itself as a XML Schema (XSD) through the root element and its namespace, the button **View as XSD** in the main window is enabled. This button leads to a window that display a graphic view of the file contents.

Alternatively you can display the current XSD file in this manner:

1. Load the XSD file as a normal XML file.
2. Use the menu **Tools > Plugins > XsdPlugin**

A window will open with the graphical representation of the data:



The symbols denote:

- **E:** element
- **T:** type

- **A:** attribute

The icons in the elements represent:

Icon	Description
	There are annotations relative to the element that will be shown in the tooltip.
	The object is a reference to another one.

## Visualization Types

The visualization modes are two:

- regular: allows to view all the schema entities (low level view)
- outline: allows to view the resulting structure hiding details like restrictions and extensions

The selection of the view mode is done using the “Outline” button.

## Changing the background color and pattern.

In the tools button there is the menu command to open the diagram background inspector. The settings are immediately applied and saved.

To open the inspector:

**Tools button > Configure aspect...**



## Exporting XSD Graphical view

In the window there are buttons that permit to export the view as PDF, HTML or SVG file. The reports can be:

- physical: the children of the entities that refer to other entities are expanded inline,
- logical: (default) the references to other objects are kept.

Command	Description
HTML Report	Open a panel where a report of the XSD diagram is shown.
Save as HTML	A HTML report of the XSD, with the diagram image, is saved to a file.
Save as PDF	A PDF report of the XSD, with the diagram image, is saved to a file.
Export as SVG	A SVG image of diagram is exported.

The preferences panel in the “Validation” tab has some option to customize the report generation.

## Working With Big XML Files

QXmlEdit is targeted to files with dimensions not too big (< 100 MB). If you need to explore files very big files, you can enter “Explore Mode” where only the element structure is loaded, without data and attributes. In this mode, denoted by this icon in the status bar: . You cannot edit the XML document, but you can search in it as usual.

To activate Explore Mode:

- Select the menu **File > Explore Structure...**

To exit from Explore Mode:

- Load another XML file or start a new document.

## Searching in Files and counting elements

It is possible to search in external files the occurrences of a given pattern using a fast parser or group and count elements.

The occurrence count will simply count how many of the elements stated in the search pattern exist in the file. The grouping show the aggregation occurrences of the elements.

Let's do an example; if the input XML file is like that:

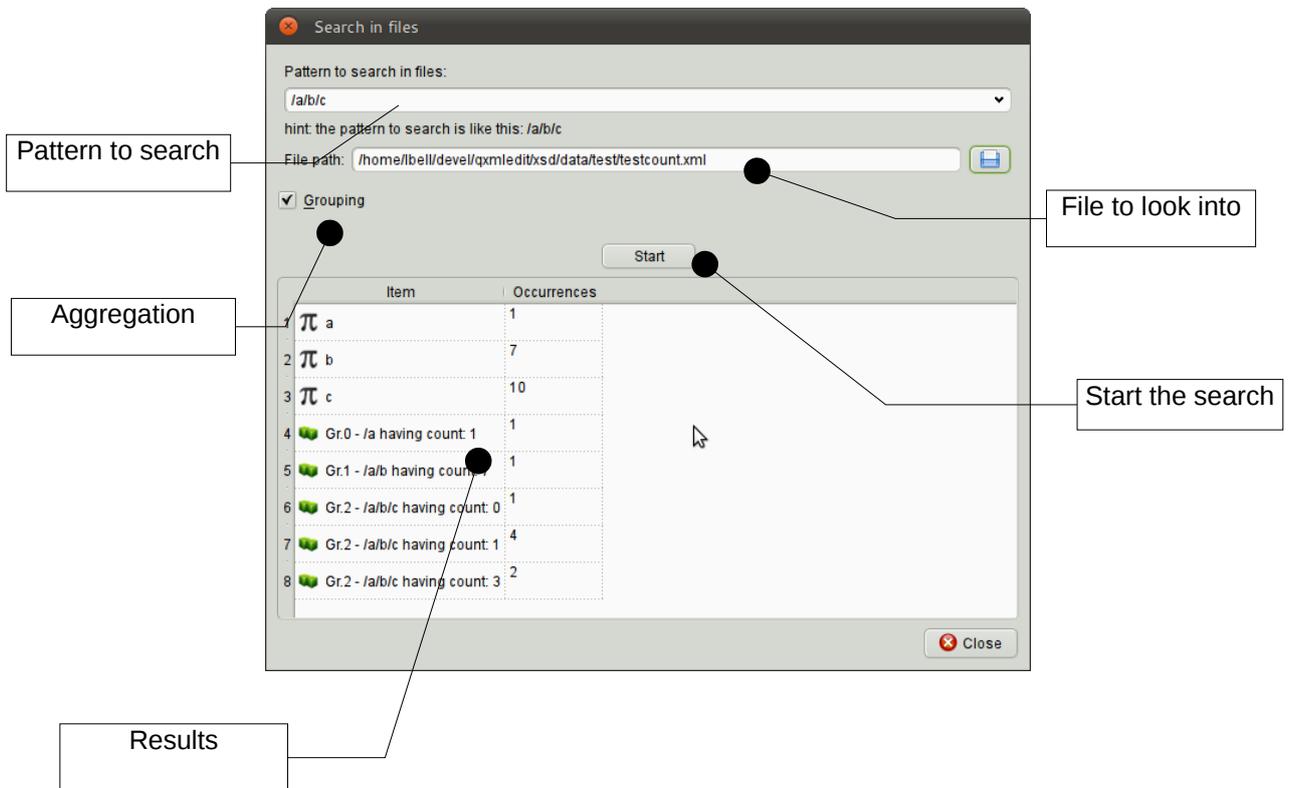


the number of elements 'a' will be 1, 'b' 2 and 'c' 3, but if we look at the groups they form, we will have 1 group composed of one 'a' element (the root one), one group of 'b' elements (the children of a ) and two groups of 'c' elements: one composed by one item (indicated by the green color), and the other one composed by two items (red color).

To search in files:

- Select the menu **Tools > Search in Files...**

The search window will open:



In this window you can insert the following informations:

Information	Description
Pattern to search	The pattern as XPath (elements only) for example: /a/b/c
File Path	Path of the file to search into.
Grouping	The scan operation will calculate element groups too.
Start	Starts the search operation.

The icons in the result grid have these meanings:

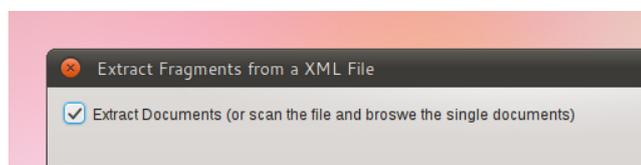
Icon	Description
π	The item is an occurrence count.
	The item is an aggregation report.

## Split a XML file

There is the possibility to split a XML file in smaller fragments and/or to examine the fragments directly in the editor. This feature is handy mainly to explore very big files that does not even fit in memory.



When starting, the program will open a welcome dialog leading to the most common operations. Choose **“Split a file”**. If you choose to hide the dialog next time the program restart and change your mind, you can reactivate it in the configuration panel. You can access this feature also from the menu **Tools**.



### Decide if you want to extract information or browse it

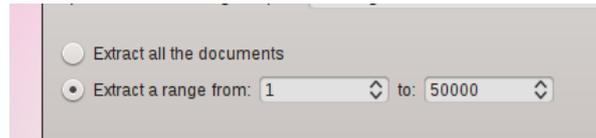
If you want only navigate the data without creating any file, do not check **'Extract Documents'**. Even if you don't extract the XML fragments into smaller file, QXmlEdit scans the file and records the information for an interactive review.



### Choose the file and how to fragment it

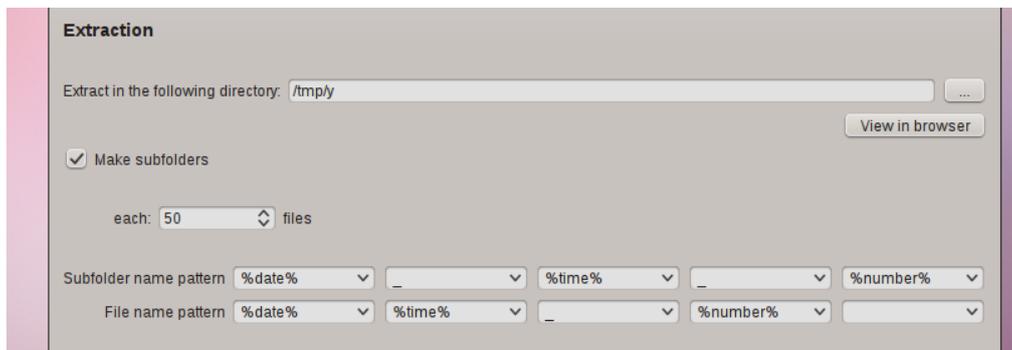
Browse to the file to examine, the insert the XPath that identifies a fragment. Insert a string of the

form /ROOT/DATA and so on.



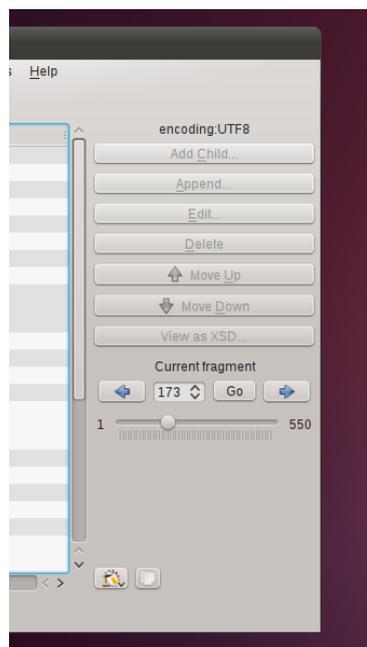
### **Limit the extracted fragments number**

Selecting a range permits to extract only a small subset of the original file. You are not obliged to waste space on your disk if you want only a single fragment.



### **Decide the output folder and the naming**

Enter the location where write the extracted fragments and decide if you want to create a folder each N files. Decide also how to name the files and the folders. The combo boxes have some predefined values to ease the task, like a timestamp, or a progressive counter.



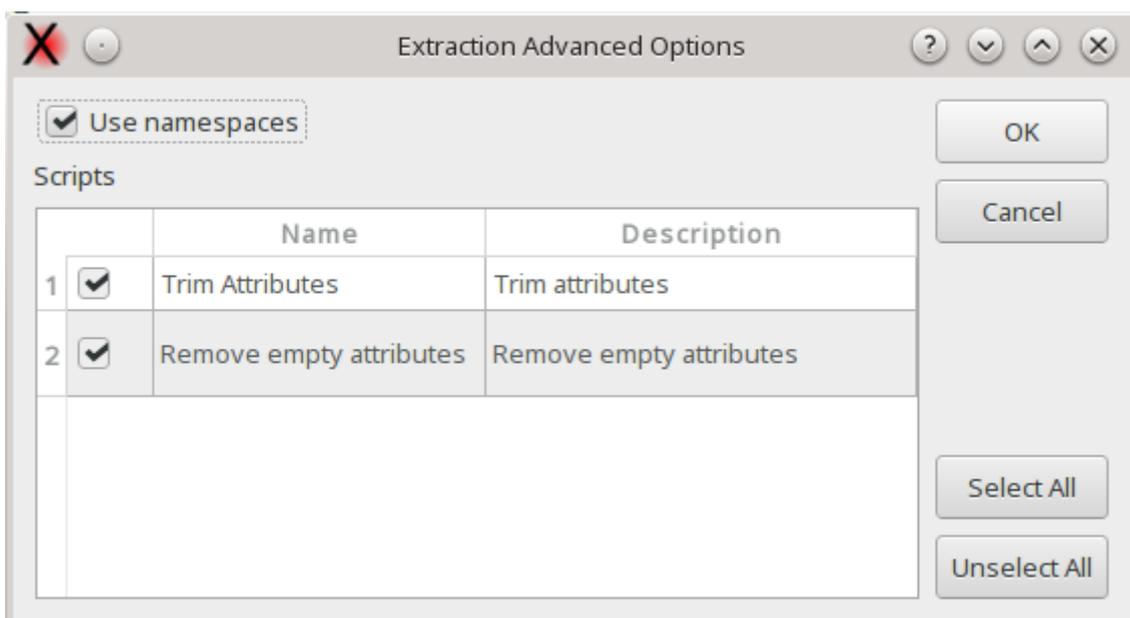
## Advanced Options

Clicking **Advanced** the advanced options dialog opens. In the dialog the following items can be selected:

- Use namespaces (it is the default behavior) when reading and creating files.
- Transform the input data using the following process:
  - Trim attributes
    - all the attributes will be trimmed *at the end*
  - Remove empty attributes
    - all the attributes that evaluate to empty string will be excluded form the resulting file

The transformations can be chosen independently one from another.

Advanced options dialog:



Information in this panel:

Information	Description
Use namespaces	Default choice is ON. Manage namespaces.
Scripts	List of scripts that can be executed.
Select All	Select all the scripts.
Unselect All	Unselect all the scripts

## Go and examine data

After the start of the operation, in the main window a navigation box appears. The fragments found in the input file are accessible directly with a random access using the information collected in the previous phases. If you choose to split the file in fragments, you can examine them in the extraction

directory.

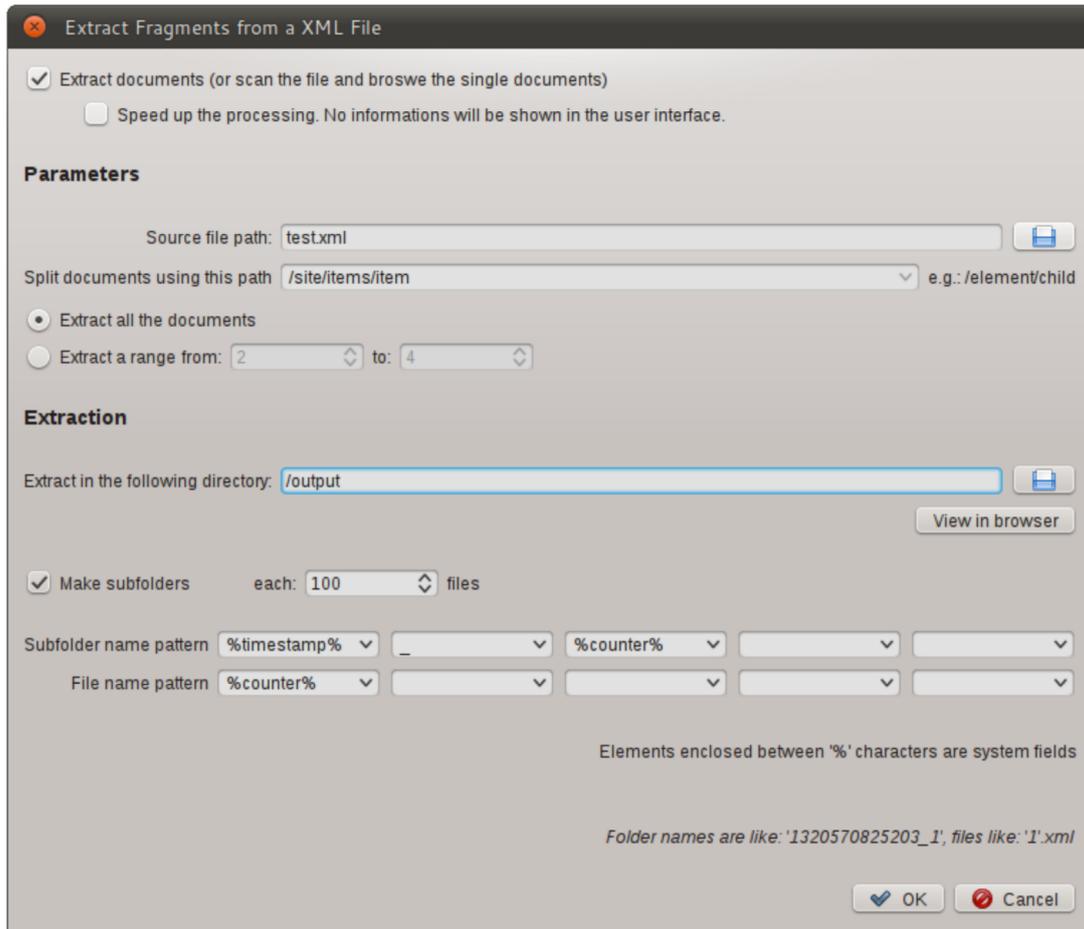


Illustration 1: The Extraction Panel

Information in this panel:

Information	Description
<b>Extract documents (or scan the file and browse the single documents)</b>	If selected the fragments found in the input file will be written to files as XML documents, one per fragment
<b>Speed up the processing. No information will be shown in the user interface.</b>	If selected, the extraction process will be faster, but after the extraction, it will not possible to browse the input file fragments in real time using the user interface. Use it if you need the output files only.
<b>Source file path</b>	The path of the input file
<b>Split documents using this path</b>	The XPath used to find the XML fragments in the input file. The path must be entered using a syntax like this: <i>/root/element/element</i>
<b>Split using the depth</b>	Use the depth of the element as criterion. Root element has depth one. Note: the elements will be extracted independently from their tag.
<b>Extract all the documents</b>	If selected causes the extraction of all the documents found in the input file.
<b>Extract a range</b>	If selected, only the documents that fall in the specified range will be extracted.
<b>Extract in the following directory</b>	The destination directory for the fragments.

Information	Description
<b>Make subfolders</b>	With this option the extracted files are created in a subdirectory when their number exceeds the configured threshold. Use it when the input file contains thousand of fragments that will make impossible the browse of the destination directory if all the files are in a single place.
<b>Subfolder name pattern</b>	The naming to assign to the subdirectories created in the extraction process. You can use a predefined token that will be expanded at run time or any other thing you like.
<b>File name pattern</b>	The naming to assign to the files created in the extraction process. The files will have a '.xml' suffix automatically apposed. You can use a predefined token that will be expanded at run time or any other thing you like.
<b>Advanced</b>	Open the advanced options dialog.

Token that can be used in the name pattern section:

Information	Description	Example Output
<b>%counter%</b>	The counter of the object created. This is an unique number that can be used to identify the object	1
<b>"%date%"</b>	The current date in the YYYY_MM_DD format	2011_10_07
<b>"%time%"</b>	The current time in HH_MM_SS_millis format	10_43_30_560
<b>"%timestamp%"</b>	The current date and time in a numeric format	123456789
<b>"%space%"</b>	A blank space used a separator Since 0.6 version.	" "

## **Flex Code Generation From Balsamiq Source**

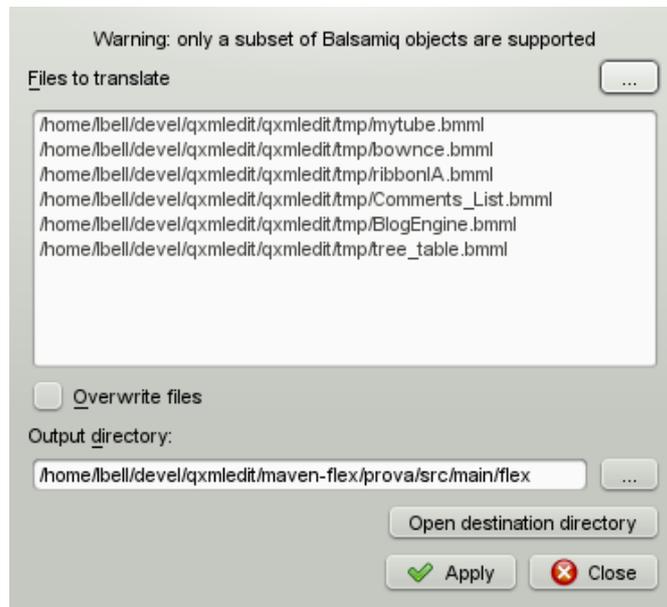
There is a plugin to generate Flex code from Balsamiq mockup program source. You can choose a set of source files that will be transformed to a Flex application in the output directory.

Only a subset of Balsamiq controls are supported, controls not supported are converted to mx:Label

List of supported controls:

- Button
- Label
- LinkBar
- DataGrid
- List
- TabBar
- Vruler
- Hruler

- TextInput
- ComboBox
- TextArea
- Paragraph
- CheckBox
- RadioButton
- Tree



## Life With Sessions

Sessions are information about a set of files and access dates, related to a work or an activity than can be recalled if needed. A file can be shared among different sessions, with different accesses. The sessions can automatically collect information of the files that the program access. Only one session can be active at time and there is no need to have a session active for QXmlEdit to work.

A session can be active (collecting information), paused or closed.

When sessions are active, there is the possibility to view the most used or recent files or folders in a list in the main view.

In the main menu the following commands are available:

Command	Information
New	Creates a new session. The current one, if existing, is closed.
Pause	Pauses the current session when active. Paused sessions does not collect information.
Resume	Resumes a paused session.
Close	Closes the current session. No session is active, then.
Details	Opens a dialog with information on the current session.
Manage	Opens a dialog where have information about all the sessions.

## The session user interface

The session panel can be shown or hidden using the toolbar button or the command in the **View** menu.

The following lists are shown:

- **Recent:** the most recent files accessed in the session. Double clicking a file opens it in the editor.
- **Most used:** the most used files of the session. Double clicking a file opens it in the editor.
- **Most recent folders:** the most recently used folders containing session related files. Double clicking a folder entry shows the open file dialog on that folder.

## Sessions Management Dialog

The screenshot displays the QXmlEdit application window. The Sessions Management Dialog is open, showing three categories of files and folders:

- Recents:** oracle.xml (1), test\_mixed.xml (1)
- Most used:** oracle.xml (1), test\_mixed.xml (1)
- Most recent folders:** /home/lbell/devel/qxmledit/xsd/test/data/xsd (last access: mercoledì 29 febbraio 2012 21:43:13 CEST)

The main editor area shows an XML tree with the following elements:

Element	#Ch.	Size	Attributes	Text
xml(version='1.0')				
birthdate	3 (3)	75		
month	0 (0)	22		January
day	0 (0)	13		21
year	0 (0)	17		1983

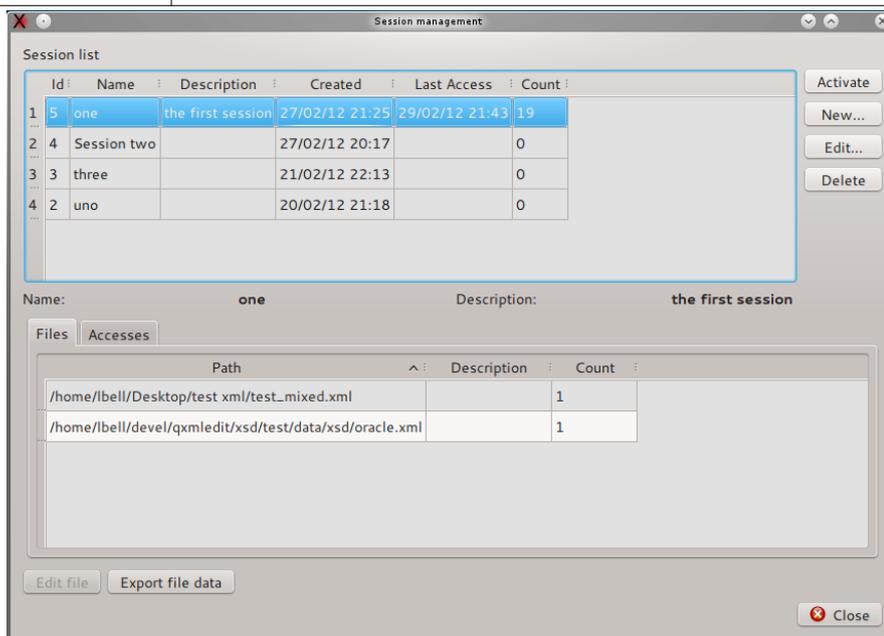
Callout boxes provide the following instructions:

- Show or hide the browser (points to the browser icon in the toolbar)
- Clicking on the files opens them in the editor (points to a file in the Recents list)
- Clicking on the folder, opens it in the open file dialog (points to a folder in the Most recent folders list)
- Session data browser (points to the Recents and Most used sections)

In this dialog you can view the sessions list. From here you can activate a session or manage sessions, creating, deleting or editing them. When a session is selected, its accesses and files are shown in the lower half of the dialog. The list of files can be copied in the clipboard. Double clicking on a file loads it in the editor.

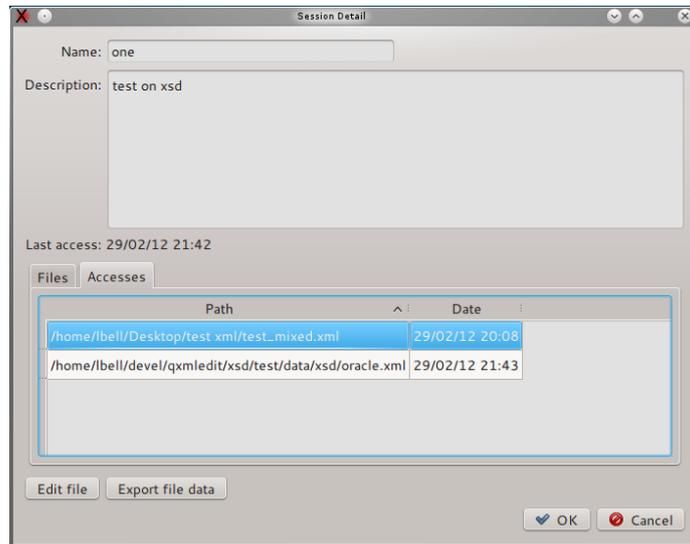
The following commands are available:

Command	Information
<b>Activate</b>	Activates the selected session, closing automatically the current one, if any.
<b>New</b>	Creates a new session.
<b>Edit</b>	Opens the session detail dialog of the currently selected session.
<b>Delete</b>	Delete the currently selected session.
<b>Edit file</b>	Loads in the editor the file selected in the lower pane.
<b>Export file data</b>	Put in the clipboard the path of the files.



## Session Properties

In this dialog you can set the session properties and examine its accesses to files. The name and the description of a session can be changed and saved.



You can set or change the session name and description. The list of files can be copied in the clipboard. Double clicking on a file loads it in the editor.

## Managing sessions data

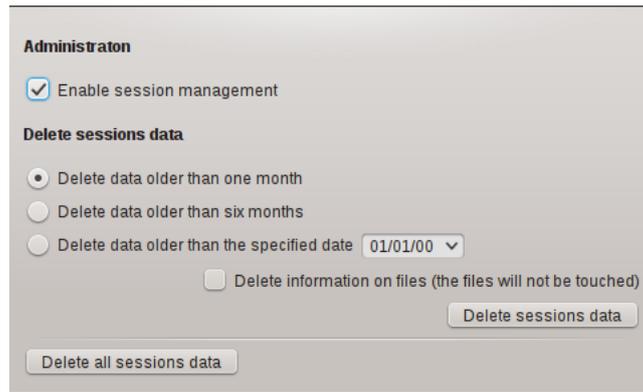
Sessions data are stored in the user data folder as indicated by the operating system using a sqlite database file. To reduce sessions data file dimensions use the configuration dialog. The cleaning procedure operates as follows:

- Completely erase the data.
- Erase sessions data on date criteria.

The following table explains in detail the options:

Option	Operation
<b>Delete all sessions data</b>	Completely erase the sessions data.
<b>Delete data older than...</b>	<p>Delete the data older than the specified date. The data are deleted as follows:</p> <ul style="list-style-type: none"> <li>- all the accesses to a file registration older than the specified date.</li> <li>- all the sessions that have last access date older than the specified date and have no access data.</li> </ul> <p>If the <b>delete files information</b> is selected, the files information without access data are deleted. The file themselves are not altered, the operation is done only on session registration inside QXmlEdit.</p>

## Sessions configuration



The session configuration panel allows to enable or disable session handling. There is the possibility to clean session data as explained in details in the paragraph “Managing sessions data”.

## Viewing data

Use the Visualization feature to graphically visualize the composition or the structure of an XML file. The file can be view as a map; only the summary data are loaded in the program main memory, so the consumed memory is of the order of magnitude of the data itself.

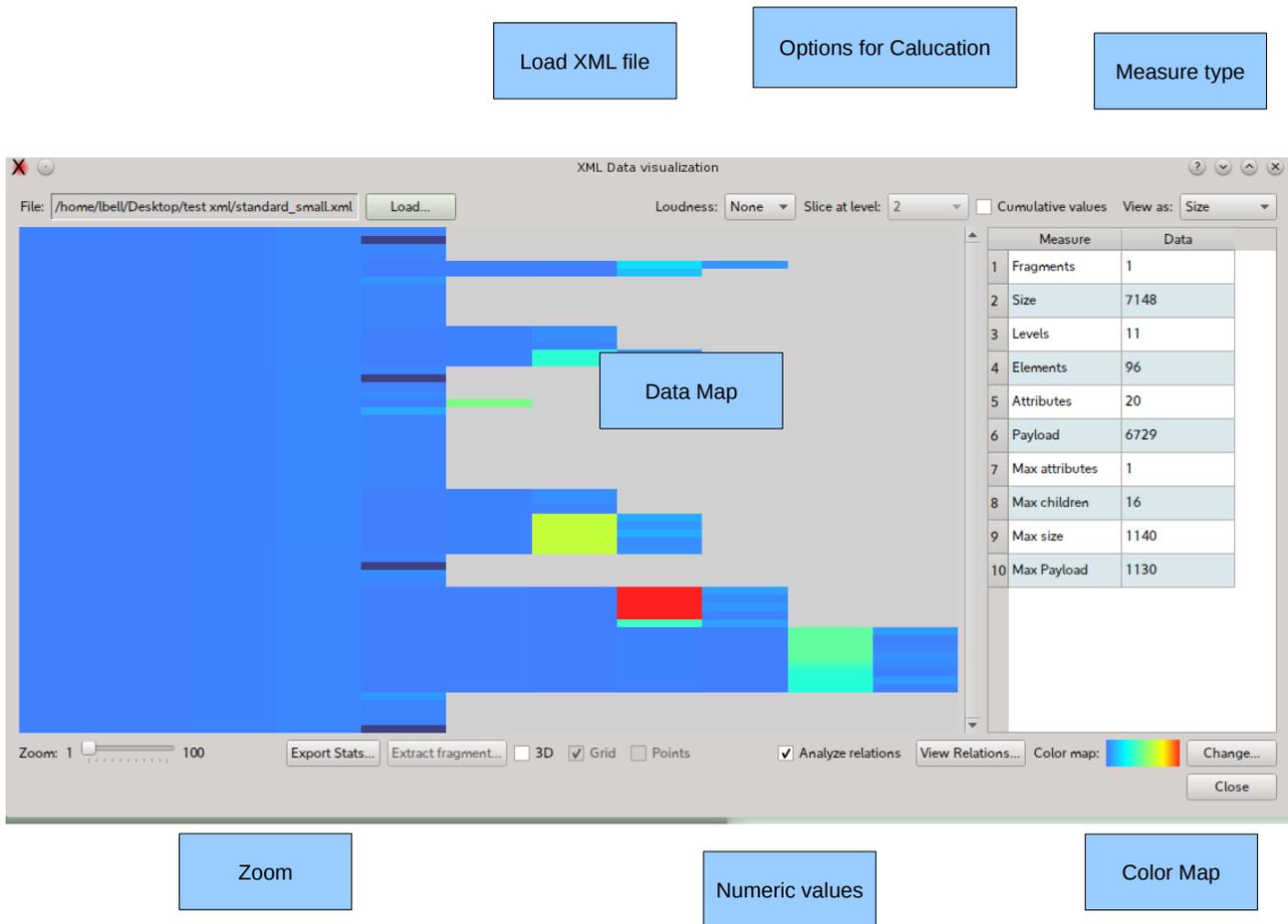
The data that can be shown are:

1. *Size*: size of the elements, attributes and text nodes.
2. *Elements*: number of children elements.
3. *Attributes*: number of attributes.
4. *Structure*: the data can be split on their depth from the root.
5. *Payload*: the size of the text nodes children of the elements.

The data values can be cumulative or relative. The cumulative measuring shows the sum of the quantity associated with a given element *and all its children*. This kind of measurement gives an idea of how the hierarchy is structured. The non cumulative measurement spots what elements have the most relevant values.

The data relevance is underlined using a color map, shown in the dialog itself.

## User Interface



To access to the Visualization use:

1. The welcome dialog.
2. The **Tools > View Data** menu.

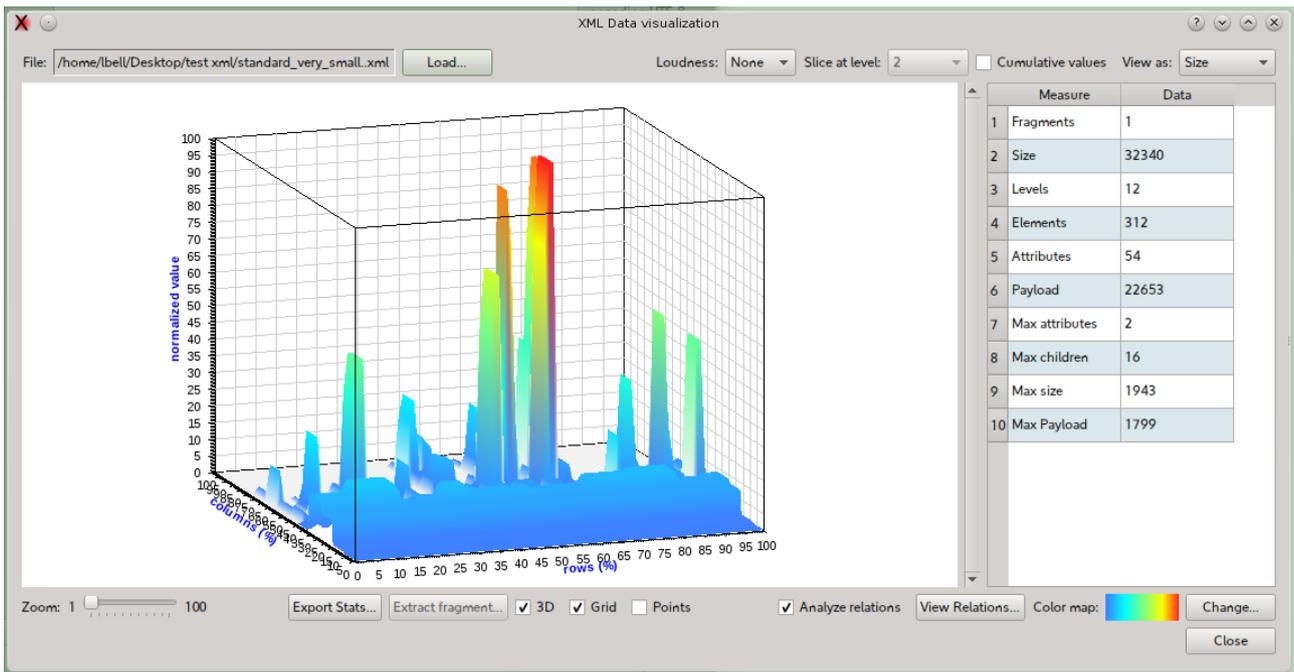
The operation sequence is as follows:

1. Load a file using the **Load file...** button.
2. Choose the measurement type from the **View as** combo box.
3. Select the **Cumulative values** if you want to display cumulative values.
4. The data are now show.
5. Select the zoom level.
6. Explore data hovering with the mouse on the interest points, a tool tip will appear.
7. Use the context menu (right mouse button) to export data.

Checking the "Analyze relations" button will give access to the relations explorer dialog.

## 3D Visualization

If QXmlEdit is compiled with 3D support, a 3D view can be enabled using “3D” check box.



The available options are:

Command	Information
Grid	Show a grid in the back walls.
Points	Show only points, do not draw surfaces.

## Measurement types

The following measurement types are available:

### Size

The size is the size in bytes of the element, its attributes and its text nodes. The cumulative measurement is the sum of the size of the element and all its children. The root element in the cumulative mode has the maximum size of all the tree.

To activate the size measurement select **Size** in the **View As** combo box.

**Note:** the calculated size can differ from the physical file because when the data are loaded, the formatting is discarded.

### Payload size

The payload size is the size in bytes of the text nodes children of an element. The cumulative measurement is the sum of the payload size of the element and all its children. The root element in

the cumulative mode has the maximum size of all the tree.

To activate the size measurement select **Payload** in the **View As** combo box.

### Attributes count

The attributes counting is the number of the attributes of each element. The cumulative measurement is the sum of the number of the attributes of the element and all its children. The root element in the cumulative mode has the maximum total of all the tree.

To activate the attributes count measurement select **Attributes** in the **View As** combo box.

### Children Elements count

The children element counting is the number of the children of each element. The cumulative measurement is the sum of the number of the children of the element and all its children. The root element in the cumulative mode has the maximum total of all the tree.

To activate the elements count measurement select **Elements** in the **View As** combo box.

### Structure

The structure view shows how the elements are divided in blocks using the depth from the root information independently from the element tag. It shows the relative width of each element at a given level. There is no cumulative option.

To activate the structure measurement select **Structure** in the **View As** combo box.

To change the depth level, use **Slice At Level** combo box.

## Commands available in the contextual menu of the map

The map exposes the following features in the contextual menu or with mouse hovering.

Command	Information
Copy Data to Clipboard	Copies the current measurement data, relative to the portion of the visible map to the clipboard. The data consist of an header and the values ordered by row.
Extract this fragment	Opens the extraction dialog with depth and fragment numbers information relative to the current point.
Copy Path to the Clipboard	Inserts into the clipboard the current element XML path.
Show current element information	Implemented as a tool tip that appears hovering with the mouse on a point of the map.

### Visualize Relationship

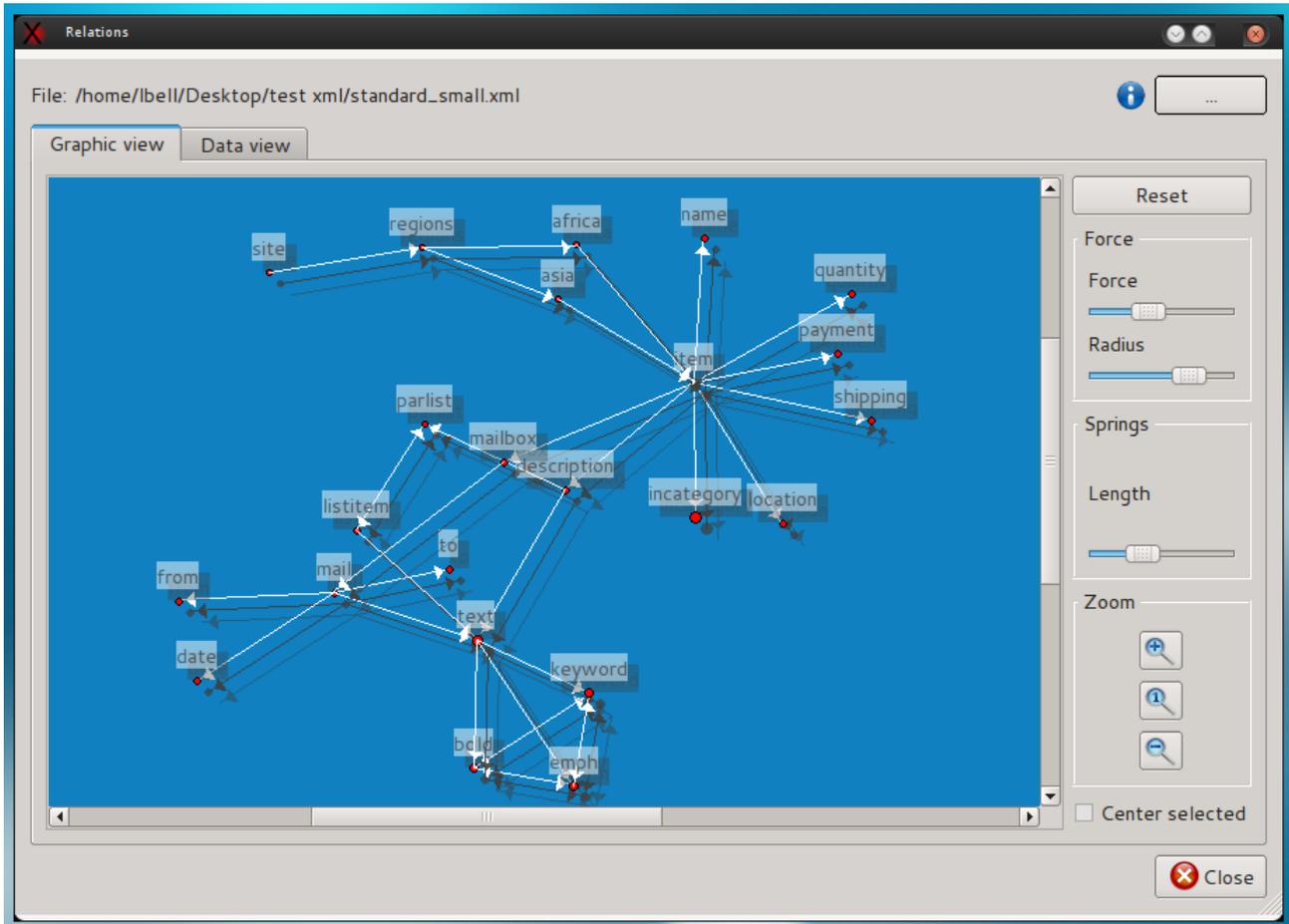
The relations panel shows in graphical form and using a table, the relationship between the nodes.

**Graphic view:** show the elements existing in the file and their relationships. The elements that are in relation father-child are connected by a line.

The elements are disposed on the canvas in random position at each invocation. The nodes are

subject to a repulsive force and tend to be distributed evenly in the view.

**Data view:** show a table of nodes and the number and the type of connected elements.



Options:

Panel	Option	Description
Force	Force	The value of the repulsive force between the nodes.
Force	Radius	The effective radius of the force.
Spring	Length	The length of the spring.
	Reset	Reset the position of the nodes.
Zoom	Zoom in	Execute a zoom in
Zoom	Zoom 1:1	Restore the view to the default size.
Zoom	Zoom out	Executes a zoom out

The relation panel can be accessed in the following modes:

- To show the data relative to the file loaded in the editor: **View->View Relations...**

- To show the data relative to a file: **Tools->Relations...**
- To show the data in the map view: check “**Analyze relations**”, load a file then “**View relations**”

## Attributes summary

In the same panel in the “Attributes View” tab a detail of the attributes is shown. The allowed options are:

Option	Description
Load Whitelist	Load a whitelist of attributes XPATH.
Load Blacklist	Load a blacklist of attributes XPATH.
Reset Lists	Clear whitelist and blacklist.

The behavior is as follows:

- if there is no whitelist and no blacklist all the attributes are included in the same summary,
- if there is a whitelist only, the attributes in the whitelist are grouped together and the remaining form another group,
- if there is a blacklist only, the attributes non in the blacklist are grouped,
- if there is a blacklist and a whitelist, the attributes in the whitelist form a group, then all the other attributes non in the blacklist are added to the group.

## Testing XML Files

This operation allows to test an XML file without opening it in the editor and can be used to test for XML syntax errors in huge files.

To access this operation:

**Menu > Tools > Test XML**

*A file choosing panel will let you select the file to be tested. At the end of the operation a report will be shown.*

## Base 64 Explorer

This dialog permits to encode and decode text from and to base 64 coding, using one of the common character encodings.

To access Base 64 Explorer:

**Menu > Tools > Base 64 Tools...**

- To encode text in base 64:
  - Write text into the “**text**” panel. The base 64 coding will appear in the base 64 panel.
- To decode from base 64:

- Choose an encoding using the “**encoding**” combo box.
- Write or paste the base 64 coded text in the “**Base 64**” panel. The text will automatically appear using the selected encoding.

## Encoding tools

This dialog will permit to see how the text is transformed from unicode to different encodings.

To access the dialog:

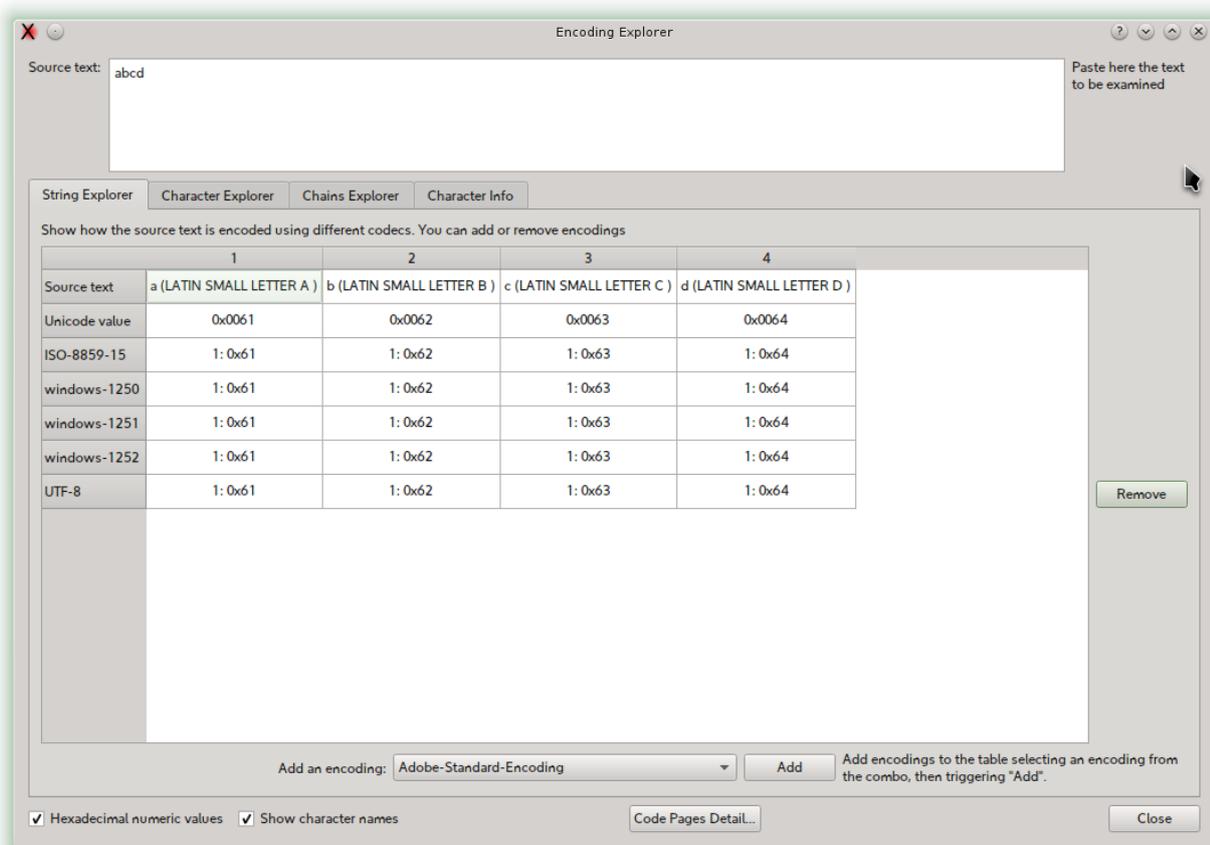
**Menu > Tools > Encoding Tools...**

### Explore how a string will be coded in a specific encoding

To explore how a string will be coded in different encodings:

1. Select the tab “String Explorer”.
2. Write the text into the Source Text box.

The encoded text will be displayed in the grid, character by character along its Unicode value.



To add an encoding to the grid:

- Select an encoding in the **encoding** combo box, then trigger **add** button.

To remove an encoding:

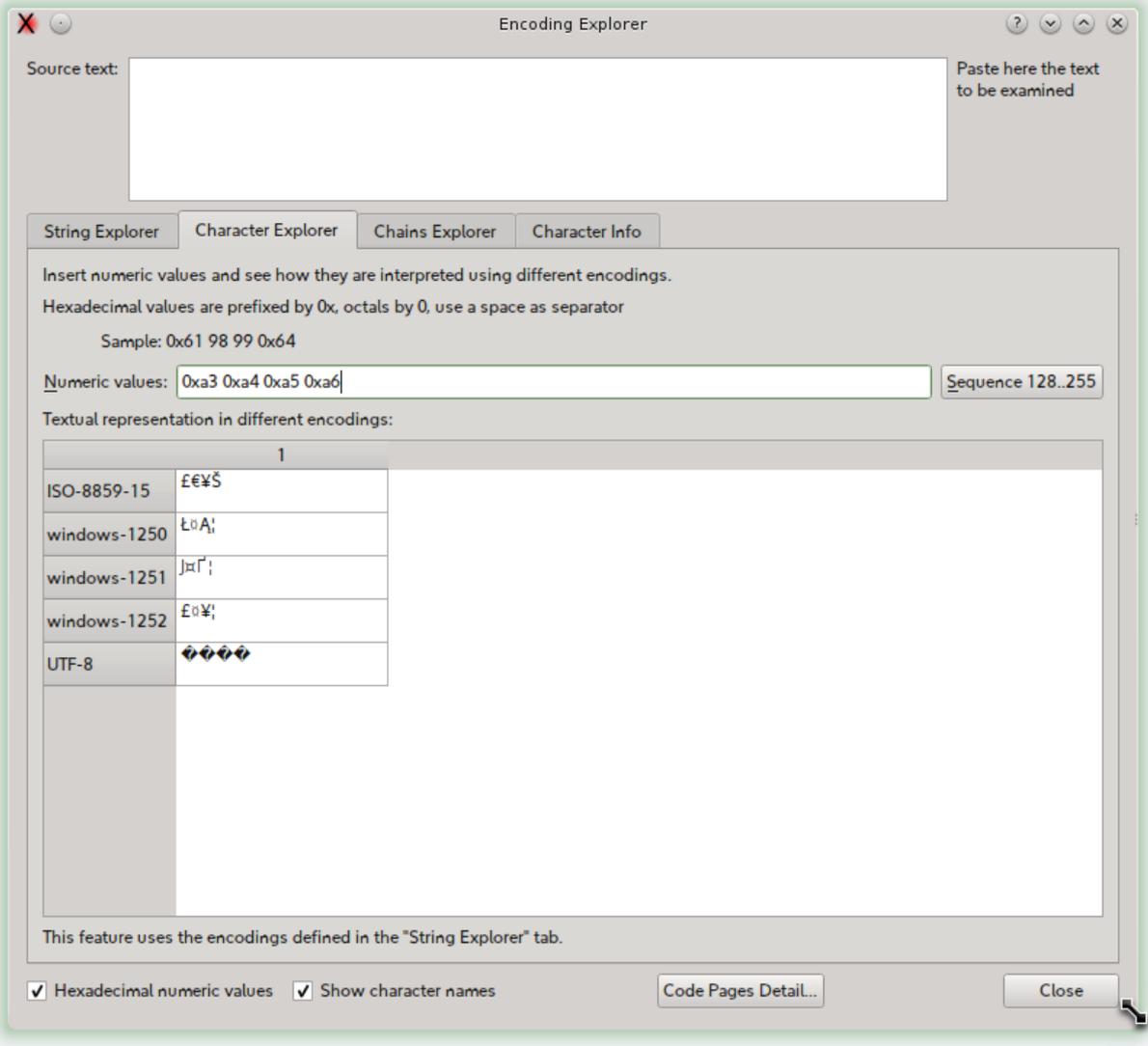
- Select the encoding row, then press **Remove** button.

## Explore how binary values are recognized by different encoding

You can insert numeric values and examine how they are interpreted by different encoding.

To access this feature:

- Select the “**Character Explorer**” tab.



Insert the numeric values in the input box, using the prefix “**0x**” for hexadecimal numbers, “**0**” for octal.

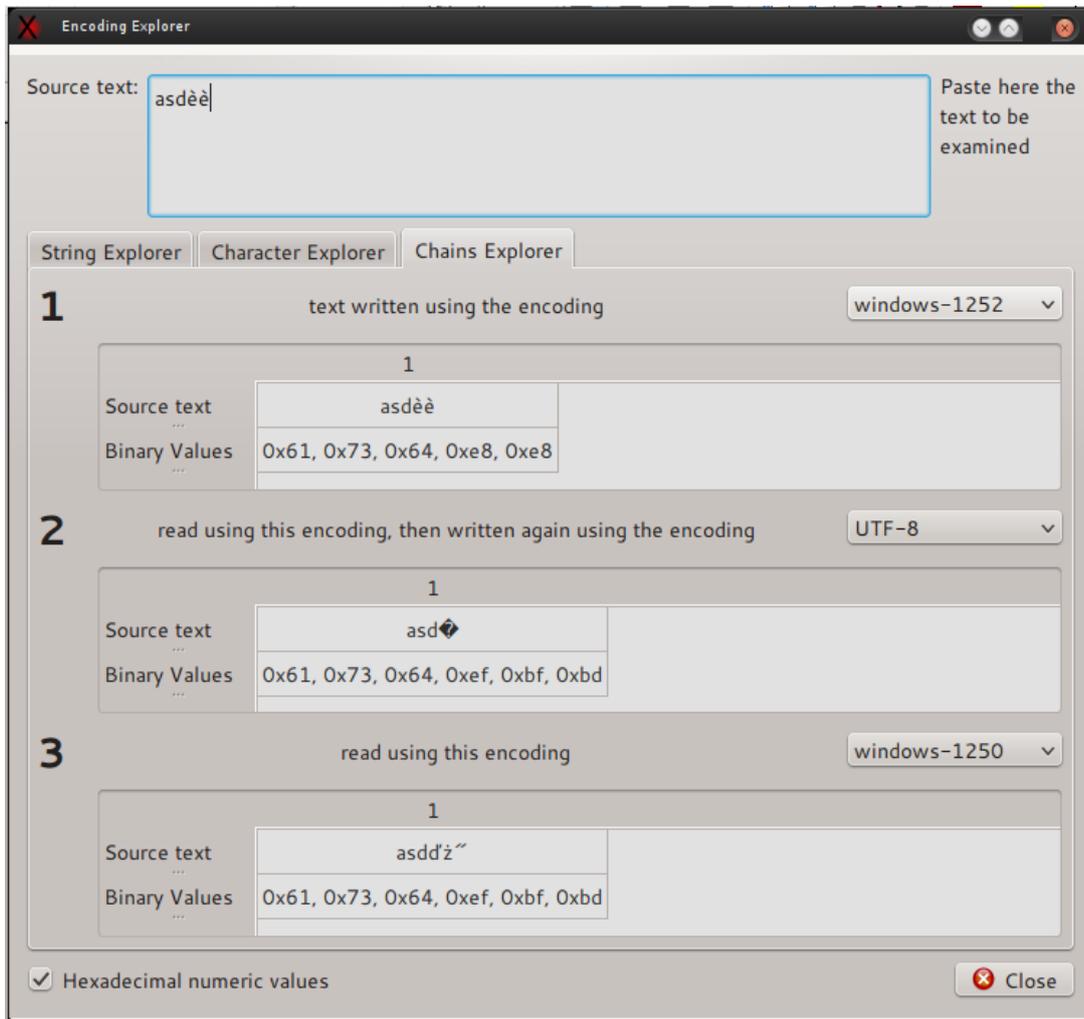
To add or remove encodings, use the “**String Explorer**” tab.

Use “**Sequence**” button to insert a sequence of 128-255 characters; if you press the Control key with the “**Sequence**” button, the characters from range 32-127 will be inserted.

## Explore how a string will be translated when written in some encoding and read in another one

Select the **Chains Explorer** tab, select an encoding (up to three are available), then write the string into **source text** box.

The grid will show the string transformation up to three steps.

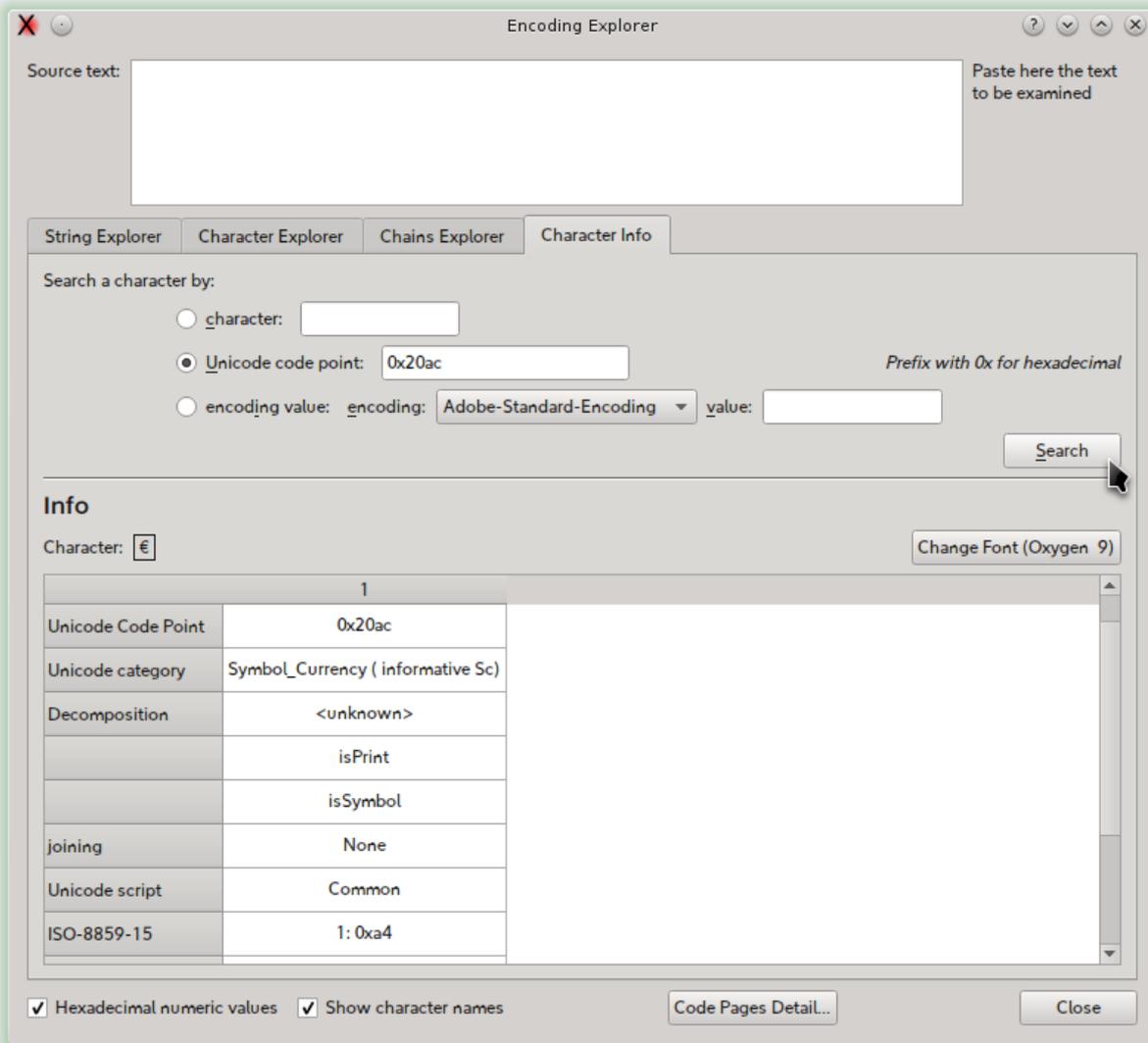


## Get information about a character

Select the **Character Info** tab, select a character using one of the methods:

- Literal character typed in a box.
- Unicode code point written in decimal or hexadecimal.
- A value of a given encoding.

You can change the character display using the “**Change Font**” button.



## Code Pages explorer

Using the menu **“Tools” > “Code Pages”** you can open the code page explorer where you can examine the code pages supported by QXmEdit.

The possible operations are:

Change the encoding using the **“Encoding”** combo box.

Search a character using:

- The literal value.
- The Unicode code point.

The search result will be highlighted in the table and reported in a box.

Option for display of the characters:

Show the UTF-8 encoding.

Show the Unicode name of the character.

Show the value of the character.

The screenshot shows the 'Code Pages' window with the encoding set to 'windows-1252'. The main table displays characters from 0x00 to 0x70. The character 'a' at position 97 (0x61) is highlighted in green. The search result box on the right shows the following information:

```

Search result:
a
Position:97 (61)
Unicode: 0x61
Name: LATIN SMALL LETTER A
  
```

	0x01	0x02	0x03	0x04	0x05
0x00	1 (0x1) U:x1 <<control> START OF HEADING 0x01	2 (0x2) U:x2 <<control> START OF TEXT 0x02	3 (0x3) U:x3 <<control> END OF TEXT 0x03	4 (0x4) U:x4 <<control> END OF TRANSMISSION 0x04	5 (0x5) U:x5 <<control> ENQUIRY 0x05
0x10	17 (0x11) U:x11 <<control> DEVICE CONTROL ONE 0x11	18 (0x12) U:x12 <<control> DEVICE CONTROL TWO 0x12	19 (0x13) U:x13 <<control> DEVICE CONTROL THREE 0x13	20 (0x14) U:x14 <<control> DEVICE CONTROL FOUR 0x14	21 (0x15) U:x15 <<control> NEGATIVE ACKNOWLEDGE 0x15
0x20	33 (0x21) U:x21 EXCLAMATION MARK 0x21	34 (0x22) U:x22 QUOTATION MARK 0x22	35 (0x23) U:x23 NUMBER SIGN 0x23	36 (0x24) U:x24 DOLLAR SIGN 0x24	37 (0x25) U:x25 PERCENT SIGN 0x25
0x30	49 (0x31) U:x31 1 DIGIT ONE 0x31	50 (0x32) U:x32 2 DIGIT TWO 0x32	51 (0x33) U:x33 3 DIGIT THREE 0x33	52 (0x34) U:x34 4 DIGIT FOUR 0x34	53 (0x35) U:x35 5 DIGIT FIVE 0x35
0x40	A 65 (0x41) U:x41 LATIN CAPITAL LETTER A 0x41	B 66 (0x42) U:x42 LATIN CAPITAL LETTER B 0x42	C 67 (0x43) U:x43 LATIN CAPITAL LETTER C 0x43	D 68 (0x44) U:x44 LATIN CAPITAL LETTER D 0x44	E 69 (0x45) U:x45 LATIN CAPITAL LETTER E 0x45
0x50	Q 81 (0x51) U:x51 LATIN CAPITAL LETTER Q 0x51	R 82 (0x52) U:x52 LATIN CAPITAL LETTER R 0x52	S 83 (0x53) U:x53 LATIN CAPITAL LETTER S 0x53	T 84 (0x54) U:x54 LATIN CAPITAL LETTER T 0x54	U 85 (0x55) U:x55 LATIN CAPITAL LETTER U 0x55
0x60	a 97 (0x61) U:x61 LATIN SMALL LETTER A 0x61	b 98 (0x62) U:x62 LATIN SMALL LETTER B 0x62	c 99 (0x63) U:x63 LATIN SMALL LETTER C 0x63	d 100 (0x64) U:x64 LATIN SMALL LETTER D 0x64	e 101 (0x65) U:x65 LATIN SMALL LETTER E 0x65
0x70	q 113 (0x71) U:x71 LATIN SMALL LETTER Q 0x71	r 114 (0x72) U:x72 LATIN SMALL LETTER R 0x72	s 115 (0x73) U:x73 LATIN SMALL LETTER S 0x73	t 116 (0x74) U:x74 LATIN SMALL LETTER T 0x74	u 117 (0x75) U:x75 LATIN SMALL LETTER U 0x75

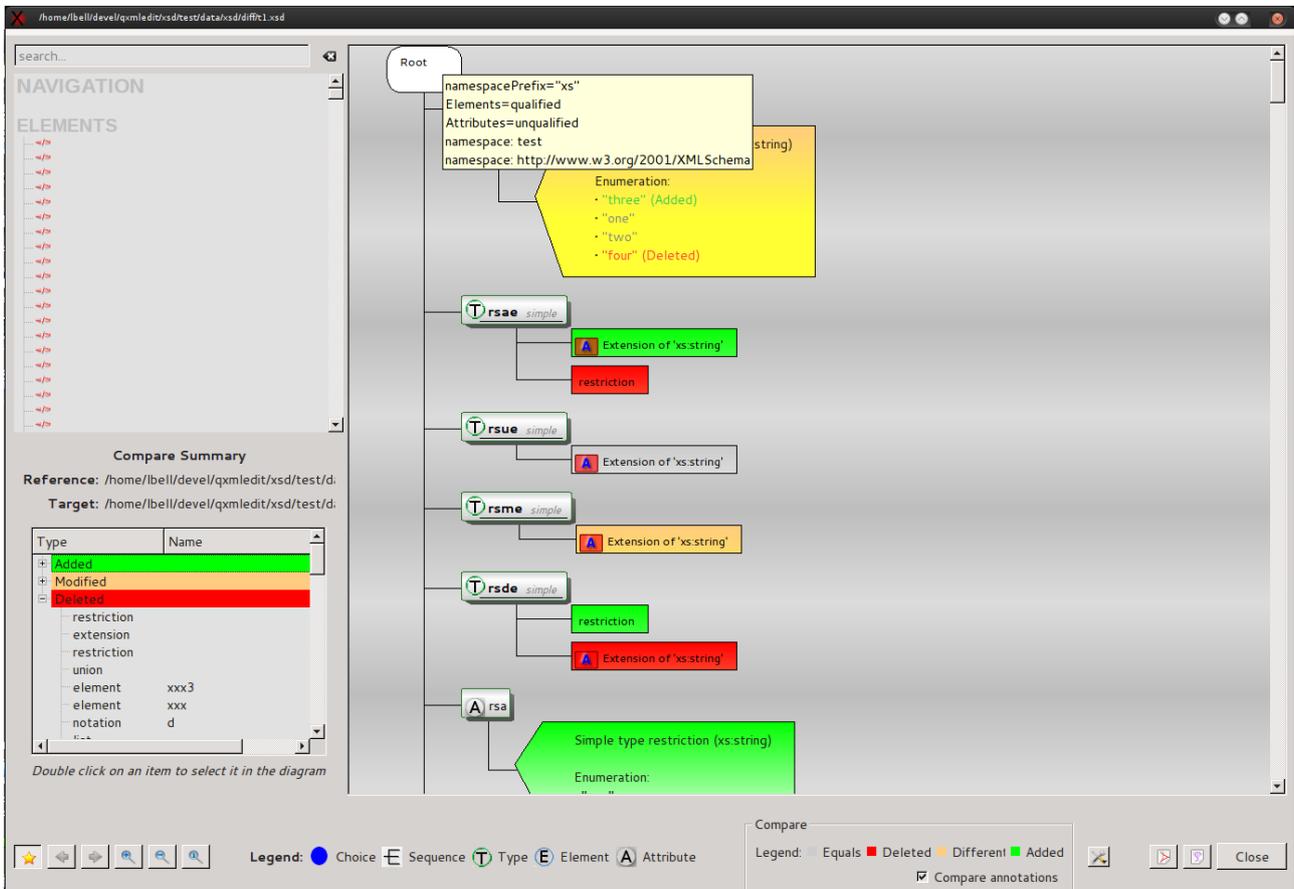
## Visually Comparing XML Schema files

The compare operation is started loading a XML Schema file, then selecting the menu operation. After the compare a window will open, showing the elements with different colors depending on their state. The comparison can be done taking into account the annotations or not. Even the single items of restriction enumerators values are shown.

To start the compare operation:

**Menu > File > Compare XSD...**

Options available: exchange source and reference and compare annotations or not.



Differences report

Items, with different colors depending on state

Tools menu

## Binary files viewer

Using the menu **Tools > Binary files** viewer you can view the raw data of a file. The data can be decoded in text using one of the supported encoding, but each row of 16 bites is decoded separately. The input file is divided in pages to not overloading the memory. In this panel you can:

- View the raw data using a text encoding.
- Search text using the selected encoding .
- Go to an offset into the file.

The screenshot shows the 'Binary files viewer' window. At the top, the file path is '/mnt/ramdisk/nonamespace.xsd' and the file size is '313 bytes'. A 'Search box' is highlighted in blue. The encoding is set to 'ISO-8859-15'. Below the search box is a table with three columns: 'Offset', 'Binary values', and 'Characters'. The table shows data for offsets from 0 to 160. A 'Raw value' callout points to the binary values for offset 32. A 'Select encoding' callout points to the encoding dropdown menu. At the bottom, there is a 'Go to address:' field with a 'Go' button and a 'Hexadecimal value' checkbox. A warning message is displayed at the bottom left: 'Warning: the character representation could be misleading if the selected encoding is not 8 bit.'

Offset	Binary values	Characters
0	3C 3F 78 6D 6C 20 76 65 72 73 69 6F 6E 3D 22 31	<?xml version="1
16	2E 30 22 20 65 6E 63 6F 64 69 6E 67 3D 22 75 74	.0" encoding="ut
32	66 2D 38 22 3F 3E 0A 3C 68 65 6D	f-8"?>.<xs:schem
48	61 20 65 6C 65 6D 65 6E 44 65 66	a elementFormDef
64	61 75 6C 74 3D 22 71 75 61 6C 69 66 69 65 64 22	ault="qualified"
80	20 78 6D 6C 6E 73 3A 78 73 3D 22 68 74 74 70 3A	xmlns:xs="http:
96	2F 2F 77 77 77 2E 77 33 2E 6F 72 67 2F 32 30 30	//www.w3.org/200
112	31 2F 58 4D 4C 53 63 68 65 6D 61 22 3E 0A 20 20	1/XMLSchema">.
128	3C 78 73 3A 65 6C 65 6D 65 6E 74 20 6E 61 6D 65	<xs:element name
144	3D 22 61 22 3E 0A 20 20 20 3C 78 73 3A 63 6F	="a">. <xs:co
160	6D 70 6C 65 78 54 79 70 65 3E	lexType>.

## Command Line arguments

QXmlEdit can be started with the following arguments:

Option	Operation
-vis <file>	Opens the data visualization panel
-anonymize <inputfile> <profile> <outputfile>	Anonymizes a file and exit
-xslt [-saxon] -xsl=<xslFile> -output=<outputfile> {-p<name>=<value>}* <inputfile>	Executes XSL transformation and exit. Parameters can be supplied using the '-p' options.

any other argument is used as a file to open.

## Appendix

### Note on “Single Application” Mode

The single application mode is enabled by default, but can be disabled using an option in the Options panel.

It consists in opening a local server when starting, After that, if a command line argument is used by the shell (e.g. the “Open With” in Windows Explorer), the arguments are sent to the existing instance via the local socket.

### Style file format

This section describes the structure of a style file.

A style file is an XML file with the following structure:

#### Root Tag:

Tag name	Attributes	Child elements
style:	<ul style="list-style-type: none"><li>“name” style name as shown in the user interface</li><li>“description”: a description</li><li>“namespace”: an optional namespace</li></ul>	<ul style="list-style-type: none"><li>“keywords”</li><li>“styles”</li><li>“ids”</li></ul>

#### Element “styles”

This element is simply a collection of “style” elements plus one optional “default” that will be used as default style, if existing.

#### Element “default”

It contains one “style” element.

#### Element “style”

Tag name	Attributes	Child elements
style	<ul style="list-style-type: none"><li>“id” unique identifier of the style. It is a string</li><li>“color”: hexadecimal representation of a color used to paint the text. Example: “FF0000”</li><li>“backColor”: optional, the background color in hexadecimal format with optional alpha at the end. Example: FF0000 or FF000080</li><li>“family”: font family if different from default.</li><li>“size”: font size if different from default.</li><li>“bold” set to 'true' or a numeric value</li></ul>	

	<p>different from zero to force bold style on font</p> <ul style="list-style-type: none"> <li>• “italic”: same of bold, but for italic style</li> </ul> <p>The only mandatory attribute is id, the others are activated if set.</p>	
--	---	--

### Element “keywords”

This element is simply a collection of “keyword” and “rule” elements

### Element “keyword”

A keyword, with associated style. Each element tag in the data file that is enrolled in this section will be printed with the indicated style

Tag name	Attributes	Child elements
keyword	<ul style="list-style-type: none"> <li>• “keyword” the name of the element to mark</li> <li>• “idStyle”: the identifier of the associated style</li> </ul>	

### Element “ruleSet”

Contains a set of “rule” and “ruleSet” using implicit AND connector

Tag name	attributes	Child elements
ruleSet	<ul style="list-style-type: none"> <li>• “connector” the connector of the children <ul style="list-style-type: none"> <li>◦ and (default)</li> <li>◦ or</li> </ul> </li> <li>• “idStyle”: for a top level rule identify the style to be used</li> </ul>	“rule”, “ruleSet”

### Element “rule” (inner)

Tag name	attributes	Child elements
rule	<ul style="list-style-type: none"> <li>• “entity”: denotes and attribute or an element, default: attribute <ul style="list-style-type: none"> <li>◦ attribute</li> <li>◦ element</li> </ul> </li> <li>• “name”: if attribute, the name of the attribute or element</li> <li>• “op”: denotes the comparison operator <ul style="list-style-type: none"> <li>◦ EQ: equals</li> <li>◦ NE: not equals</li> <li>◦ EX: value exists</li> <li>◦ NX: value non existent</li> <li>◦ GT: greater than</li> </ul> </li> </ul>	none

	<ul style="list-style-type: none"> <li>○ LT: less than</li> <li>○ GE: greater or equal</li> <li>○ LE: less or equal</li> <li>○ SS: substring (string only)</li> <li>○ ST: starts with (string only)</li> <li>• “type”: type of comparison, optional, default: string <ul style="list-style-type: none"> <li>○ s: string</li> <li>○ n: number</li> <li>○ p: position</li> </ul> </li> <li>• “value”: the comparison value. If position, the value should be: <ul style="list-style-type: none"> <li>○ numeric</li> <li>○ first (literal)</li> <li>○ last (literal)</li> </ul> </li> <li>• “case”: if the comparisons take the case of the strings into account, default: true <ul style="list-style-type: none"> <li>○ true: case sensitive</li> <li>○ false: not case sensitive</li> </ul> </li> </ul>	
--	--	--

### Element “ids”

This element is simply a collection of “id” elements

### Element “id”

This element contains the name of attributes whose content will be printed next to element tag

Tag name	attributes	Child elements
id	<ul style="list-style-type: none"> <li>• “id” name of the attribute that is considered as an identifier.</li> <li>• “alpha”: if true, force the enclosing of the value between quotes.</li> </ul>	

This is a complete sample style file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- this is a sample QXmlEdit style file -->
<style name="Log" description="log test style">

<keywords>
<keyword keyword="resource" idStyle="1"/>
<ruleSet idStyle="2">
  <ruleSet connector="and">
    <rule entity="element" op="eq" type="s" value="event"/>
    <rule entity="attribute" name="level" op="eq" type="s" value="fatal" case="false" />
  </ruleSet>
</ruleSet>
<ruleSet idStyle="3">
  <ruleSet connector="or">
    <rule name="level" op="eq" type="s" value="ERROR" />
    <ruleSet connector="and">
      <rule entity="attribute" name="level" op="eq" type="s" value="INFO" case="true" />
      <rule name="pri" op="gt" type="n" value="12"/>
    </ruleSet>
  </ruleSet>
</ruleSet>
</style>
```

```

    </ruleSet>
</ruleSet>
<ruleSet idStyle="4">
  <rule name="level" op="eq" value="WARNING" case="true" />
</ruleSet>
<ruleSet idStyle="5">
  <rule name="level" op="eq" type="s" value="INFO" case="true" />
</ruleSet>
</keywords>

<styles>
<style id="1" color="2080FF" />
<!-- fatal -->
<style id="2" color="FF0000" size="12" family="Arial" bold="true" italic="false" icon=":E:" />
<!-- error -->
<style id="3" color="FF4080" size="10" family="Arial" bold="true" italic="false" icon=":E:" />
<!-- warning -->
<style id="4" color="C0C080" size="10" family="Arial" bold="false" italic="false" icon=":W:" />
<!-- info -->
<style id="5" color="000000" size="8" family="Arial" bold="false" italic="true" icon=":I:" />
<default>
  <style id="default" color="C0C0C0" size="8" family="Arial" bold="false" italic="true" />
</default>
</styles>

<ids>
  <id id="name" alpha="true" />
  <id id="id" alpha="" />
</ids>

</style>

```

### Installation of new styles

Given that styles are so simple, they can be created by the user with a simple text editor. Styles are searched in a directory configured via '**Configure...**' menu.