QXmlEdit

User Manual

A guide to the examine XML data with QXmlEdit.

Version 0.8.11

Contents

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.

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://*code.google.com/p/qxmledit*.

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

What's New in QXmlEdit 0.8.11

New features: new XSD visualization, zoom on selected XSD elements, tools for manipulation of XML hierarchy.

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.

Copy selected data into the clipboard

Sessions panel view/hide

Encoding indicator

Main edit functions

Sessions data panel

XSD viewer

Data modification indicator

Styles selector

Read only status

Working mode indicator

Session status
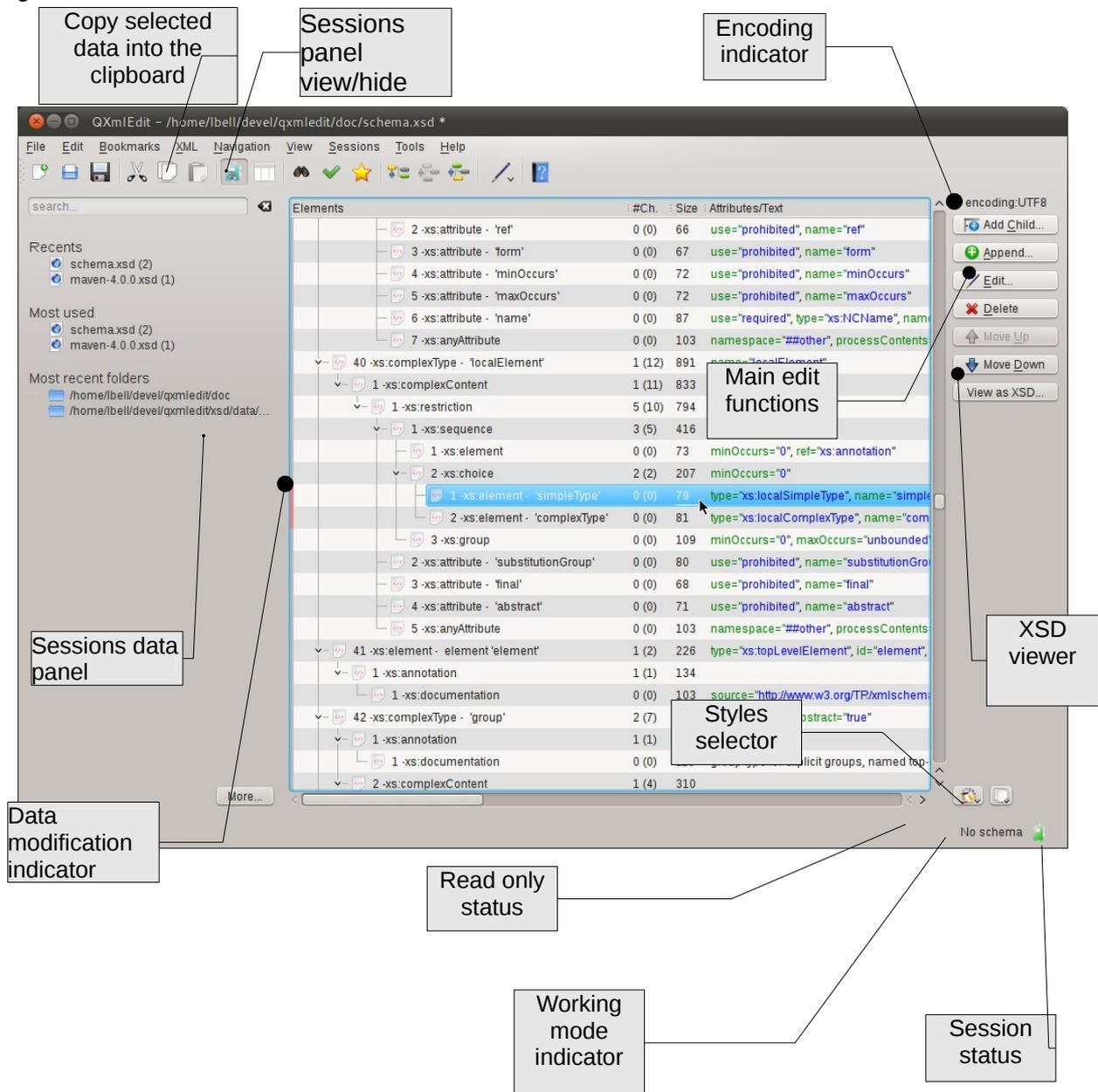
Operation with XML

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 on 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 sibling 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.

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.

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
Bookmarks	all about bookmarks
XML	XML tree manipulation (insertion, deletion, etc)
Navigation	XML tree navigation
View	view commands and options
Sessions	Session related commands
Tools	plugins and tools
Help	help

Working with Small 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.

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 maintains the current state.

Creating a File From Clipboard Contents

Sometimes you can create a new file by cutting 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

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.

Exporting the XML

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

- 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

Preferred directories

Remove a link

Insert a new link to directory
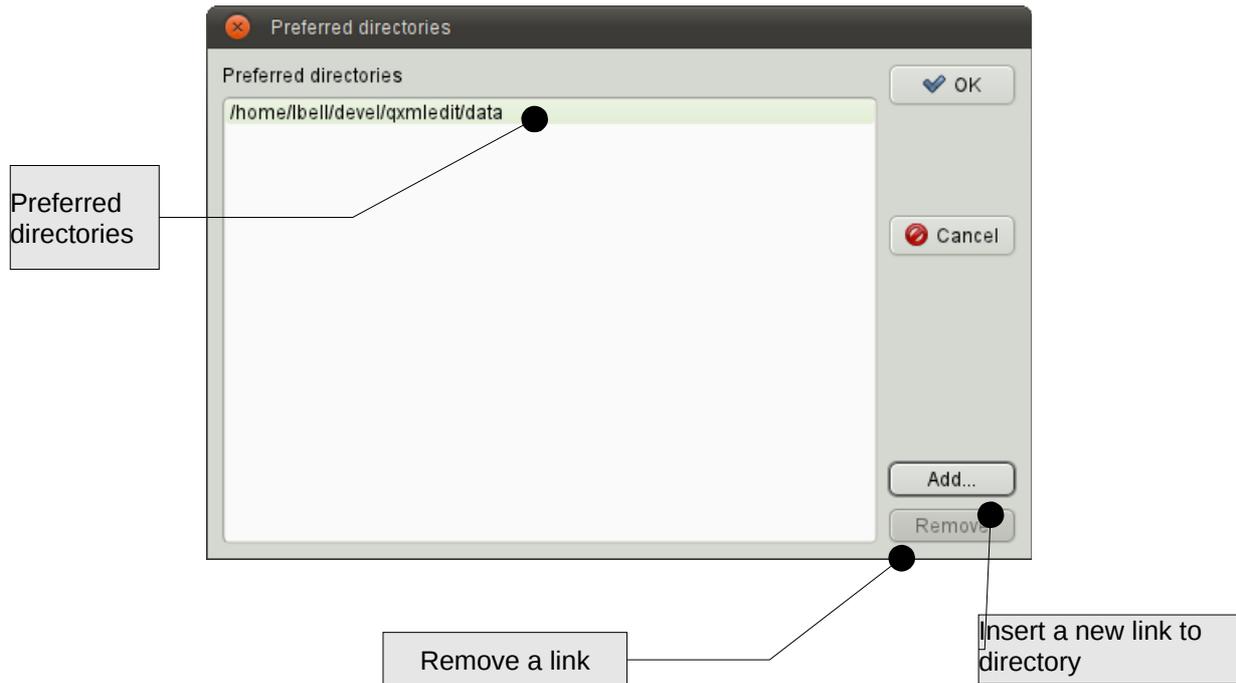
To add a new link to a preferred directory

- 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**

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

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

To show a more compact view:

- select the menu **View > Compact View**
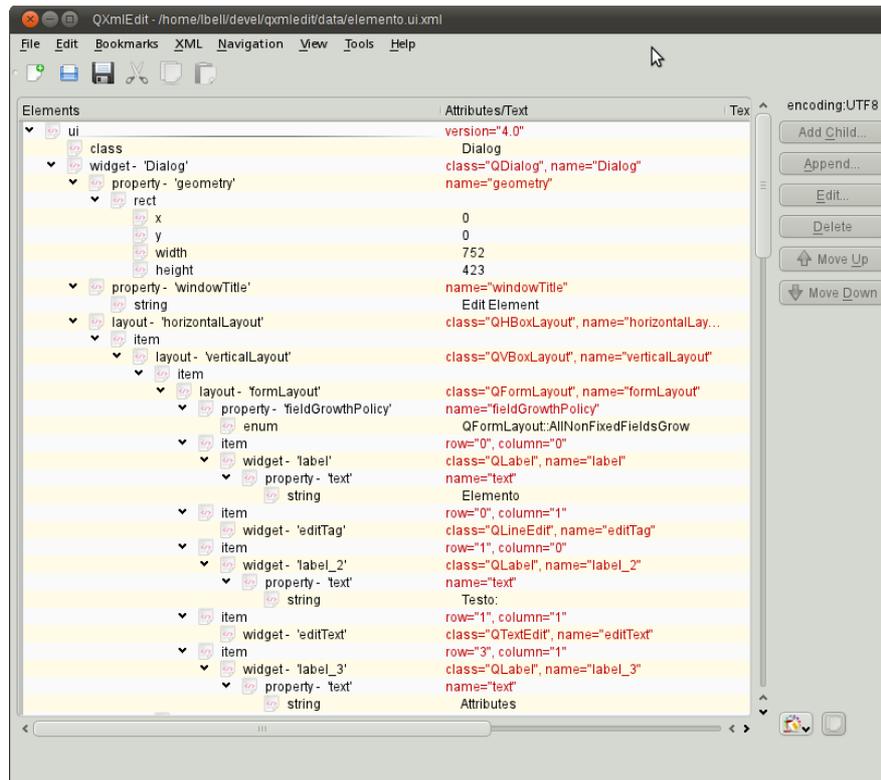
To show the attribute length near to their name:

- Select the menu **View > Show Attributes Length**.

To show attribute contents using a fixed size font, to better discover the presence of spaces:

- Select the menu **View > Show Attributes Length**.

To show in a separate column the text length of the elements:

- Select the menu **View > Show Text Length**.

To show the number and size of the children of each element:

- Select the menu **View > Show Element Size**.

To show the text contained into elements as base 64 encoded:

- Select the menu **View > Show Text as Base 64 Coded.**

> **Note:** this option can have an impact on visualization speed.

To expand all the closed branches of the XML tree:

- Select the menu **View > Expand**

To enlarge or reduce the size of the character used to display the data:

- Select the menu **View > Zoom In** or **View > Zoom Out.**

or

- Use the Control key while scrolling with the mouse wheel.

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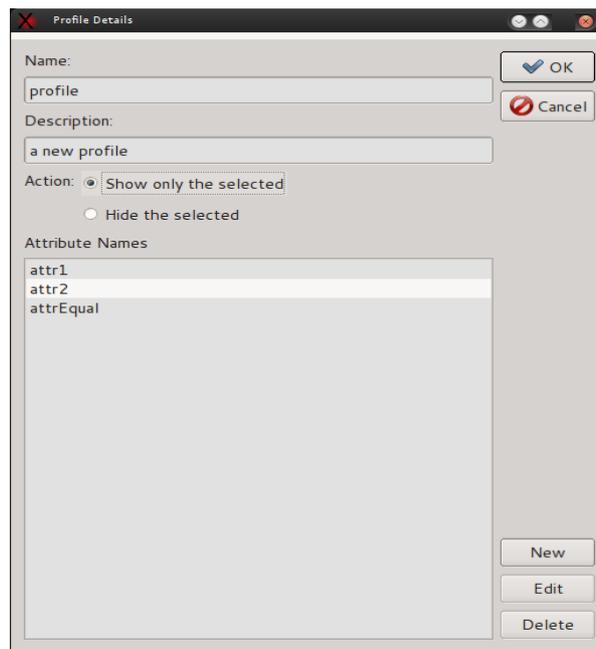
To apply a temporary profile, based on the current element:

1. Select an element.
2. Use the menu **View > Filter attributes > Apply a Temporary Profile**.
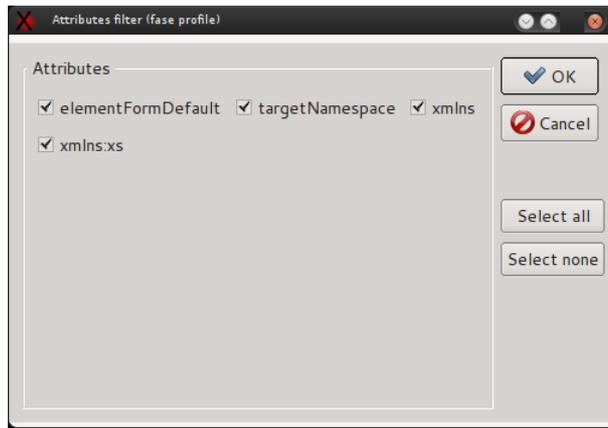
It is possible to transform the temporary profile into a regular one using the menu Use the menu **View > Filter attributes > Save the current Profile.**

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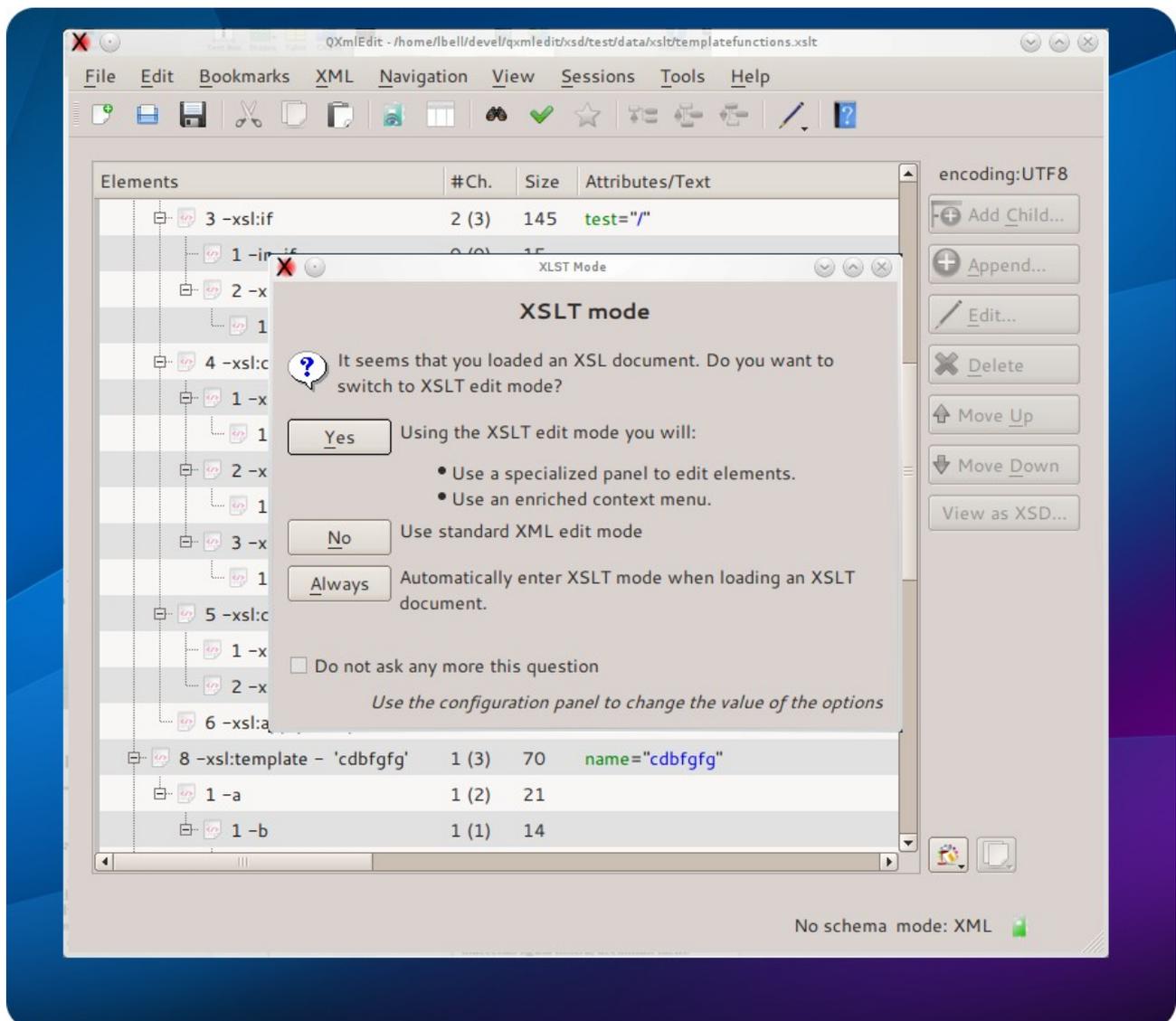
1. To activate the XSLT mode simply answer "Yes" or use the return key.
2. To have the mode always activated without answering the panel any more, select "Auto".

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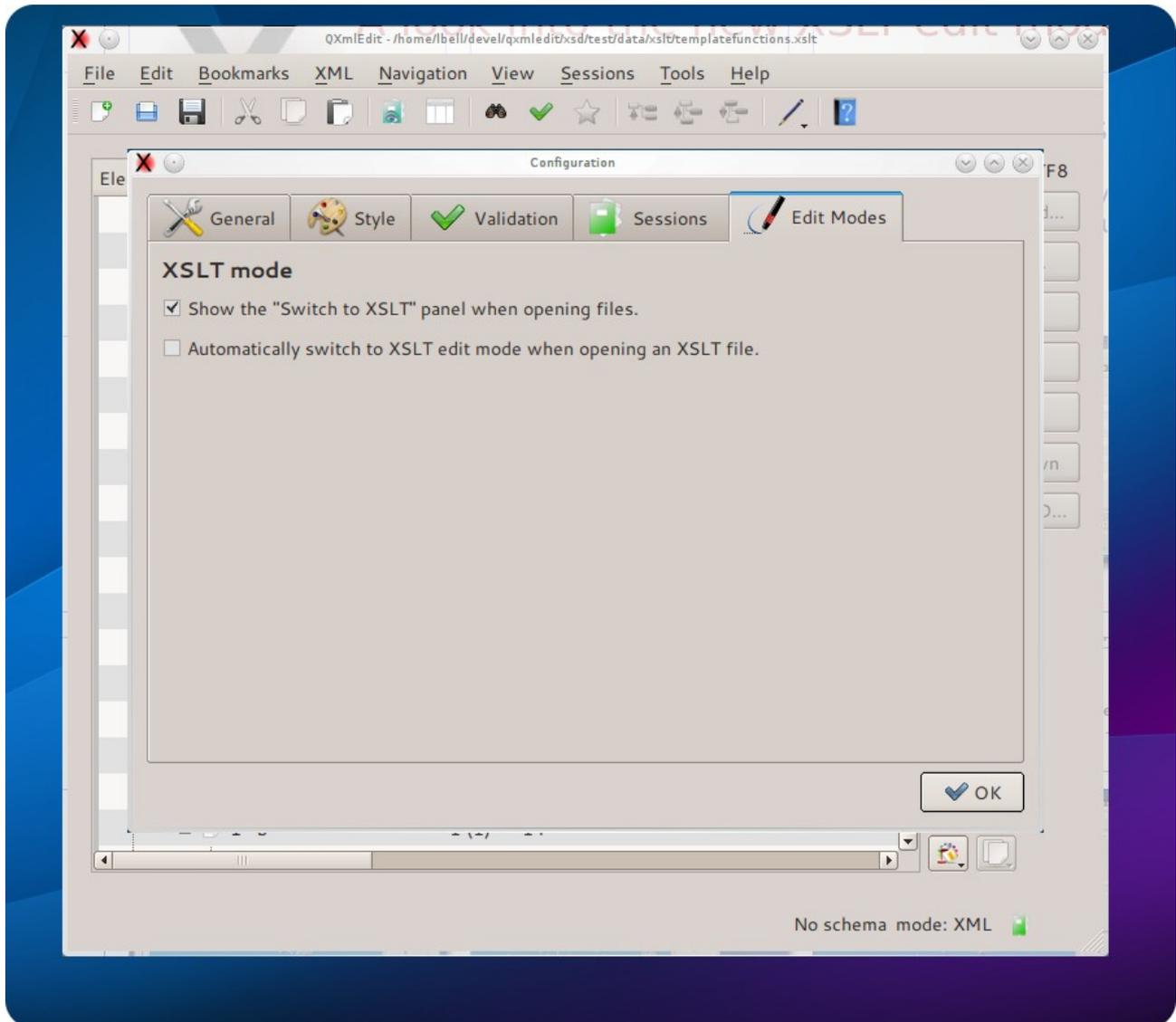
1. From the Edit menu choose "Configure", then "Edit modes".
2. Set if the XSLT edit mode should automatically engage.
3. You can hide the panel if you do not care to enter this mode.

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.
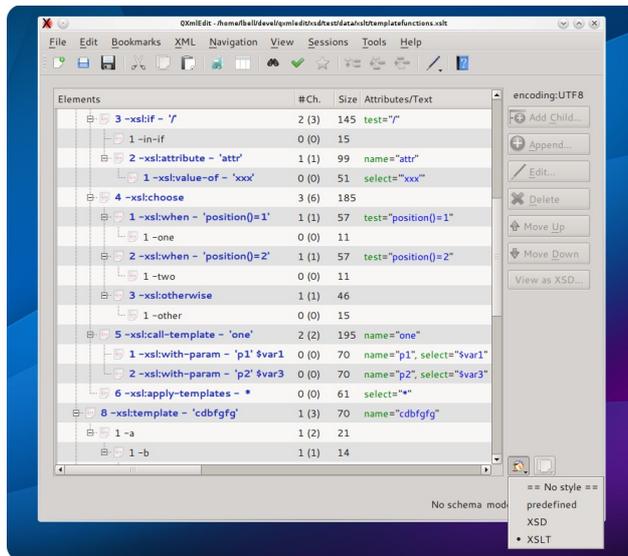
Specific edit panels

1- Double click an XSL element in the editor.

2- A special panel will open.

3- When you edit the "select" attribute, the autocompletion 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.
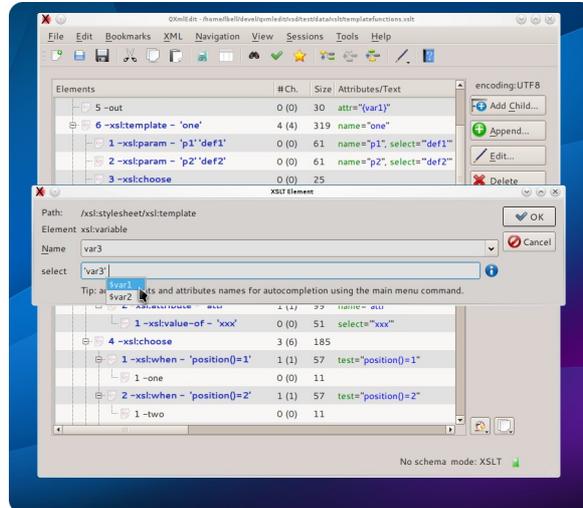
A specific context menu

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.
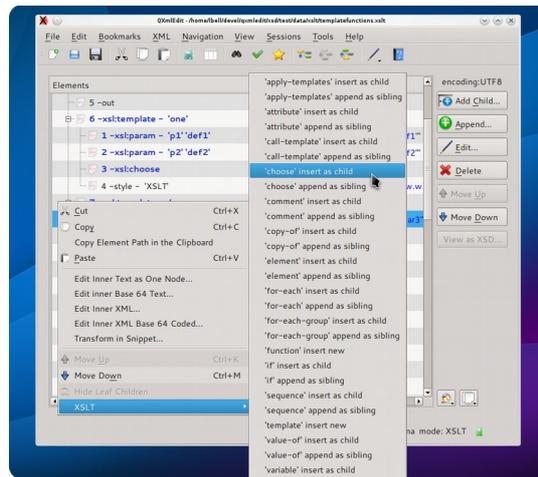
Using names for autocompletion

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

Use the menu **XML > Target XML**
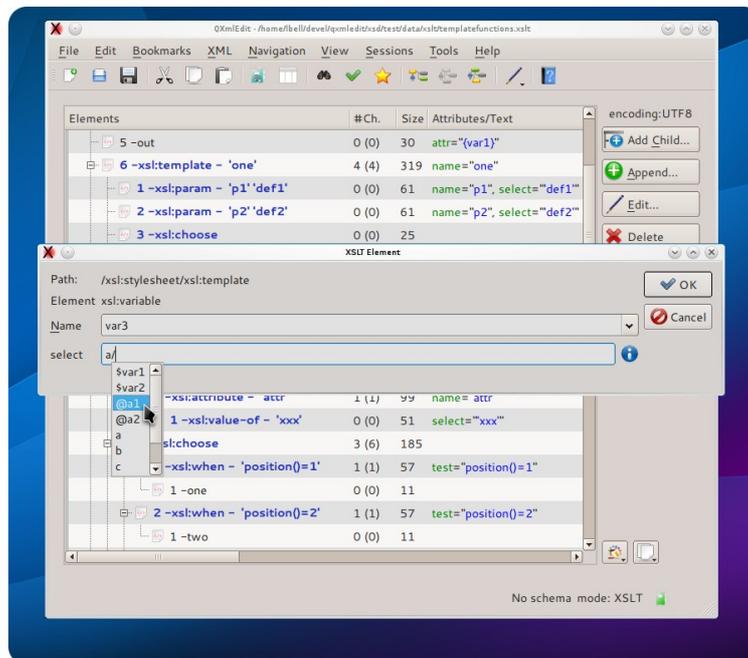
Navigate templates and functions

Using the menu **Navigation > Show XSL Navigator** you can access a panel listing all the templates and functions existing in the file. A double click will lead you to the template or function definition in the editor.
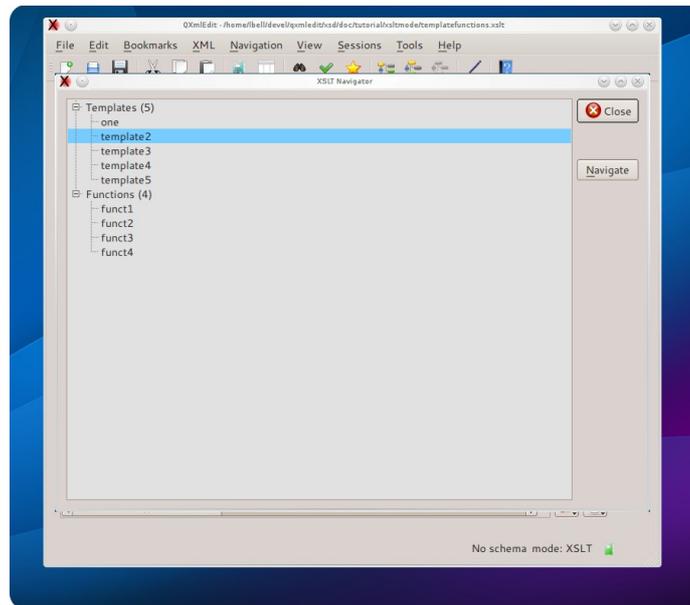
Choosing a Display Style

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 a XML Schema file is represented in the following screenshot:
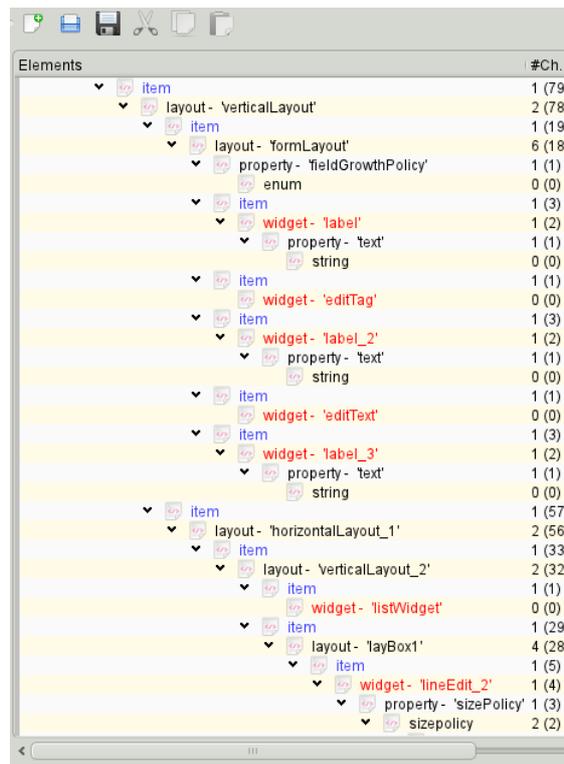
The style can be used to highlight the elements.
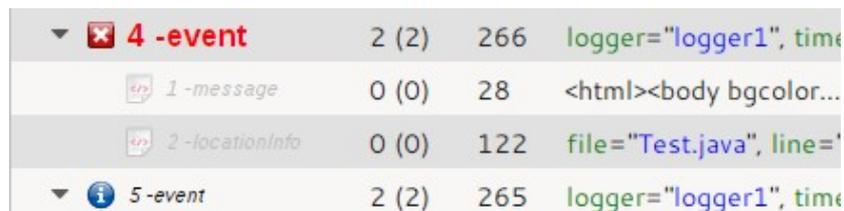
The Style can be based on rules, in the appendix there is the full detail.

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:
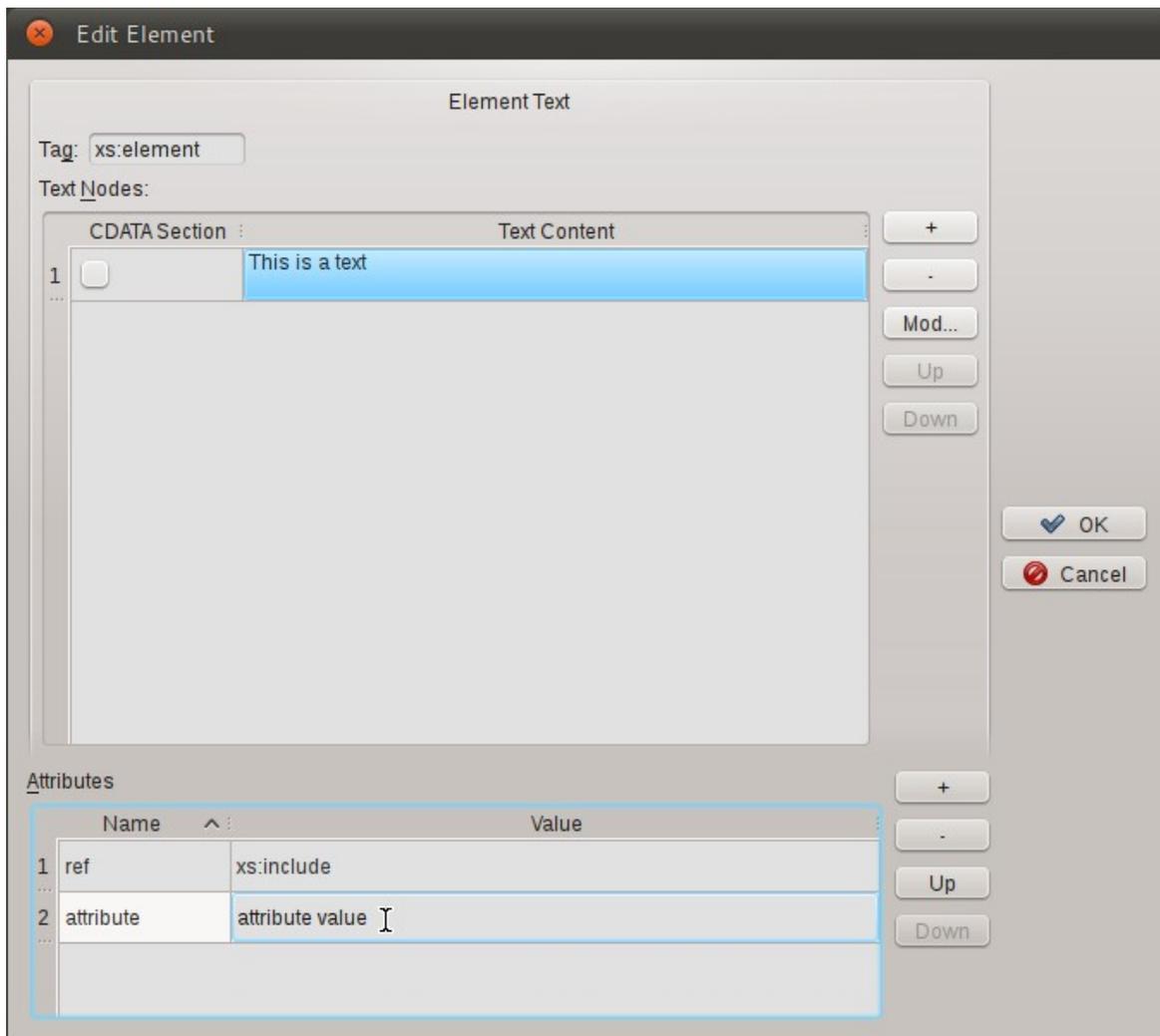
In the panel you can insert the following information:

Name	Information
Tag	Type 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.
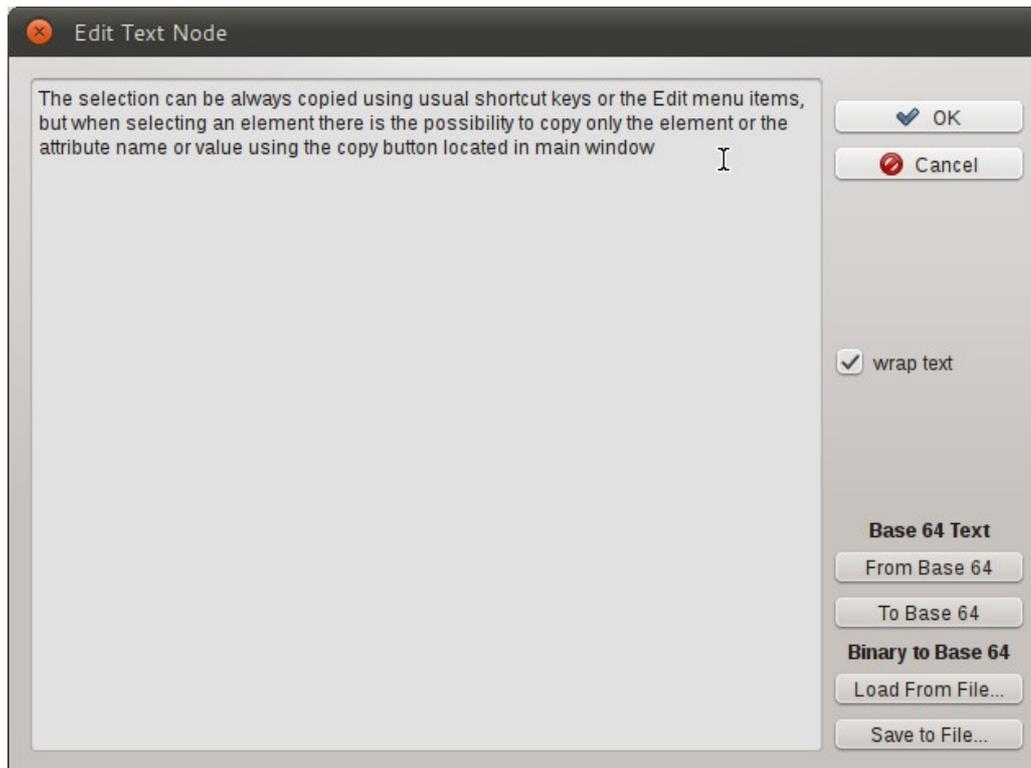
To insert a base 64 coded text:

- 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 text into a binary file, decoding the values:

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

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:
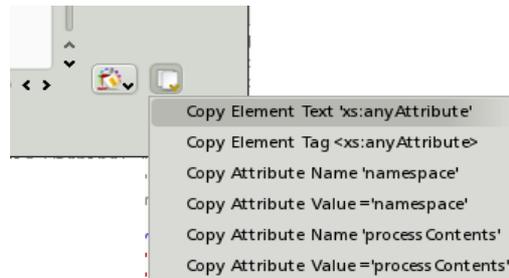
to copy the selected element tag:

- 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

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**

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 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 a XML prolog on document creation

You can insert a 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.

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.

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

• select the menu **XML > Validate Using Document Reference**

Another option is to load an XML Schema file to validate the current data

• select the menu **XML > Validate Using Specified File**

The file is cached until an explicit reload command is given.

To use a different 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 main window. To open the search panel:

- use the **Edit > Find** menu

or

- press Control+F
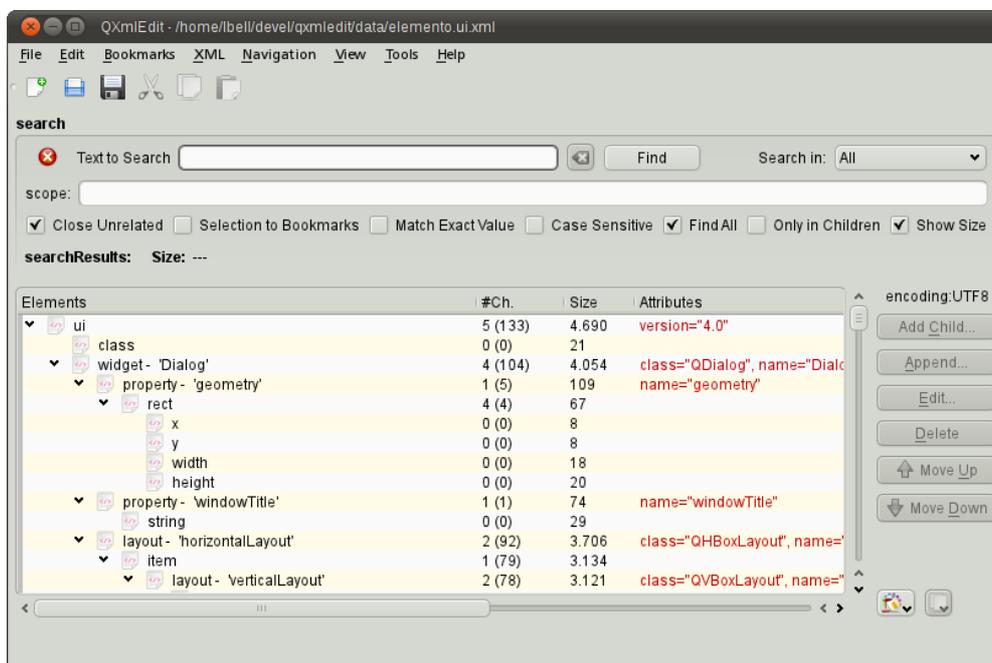
In the panel you can do the following operations:

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

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	
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.

Search results

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

Matched items
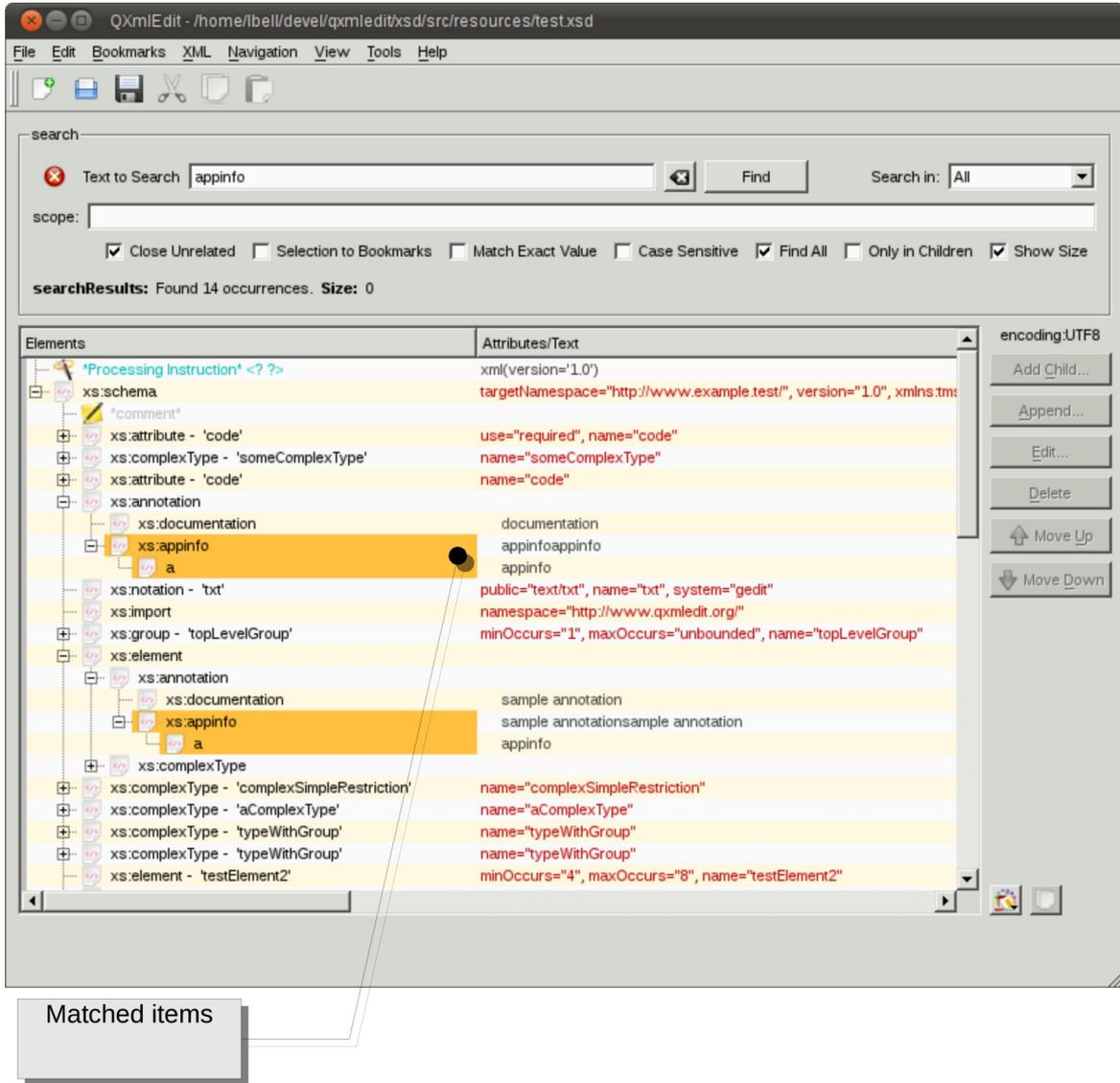
XPath syntax

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	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.
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"

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:
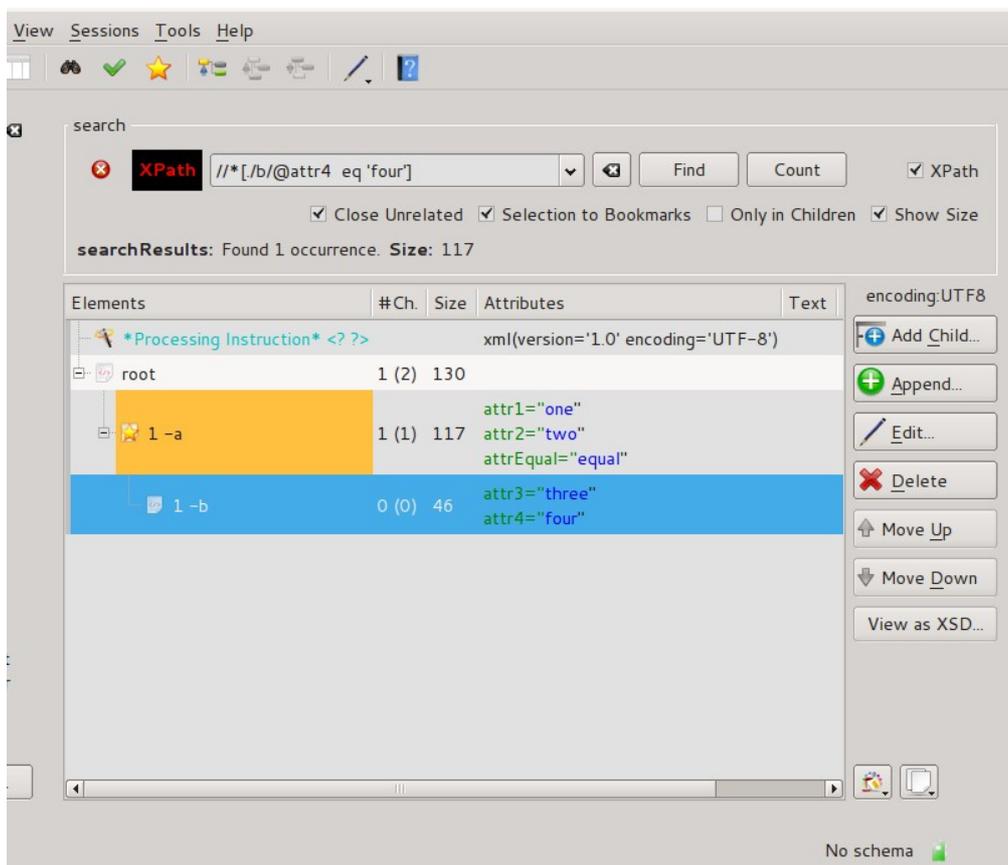
Browsing the search results

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.

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
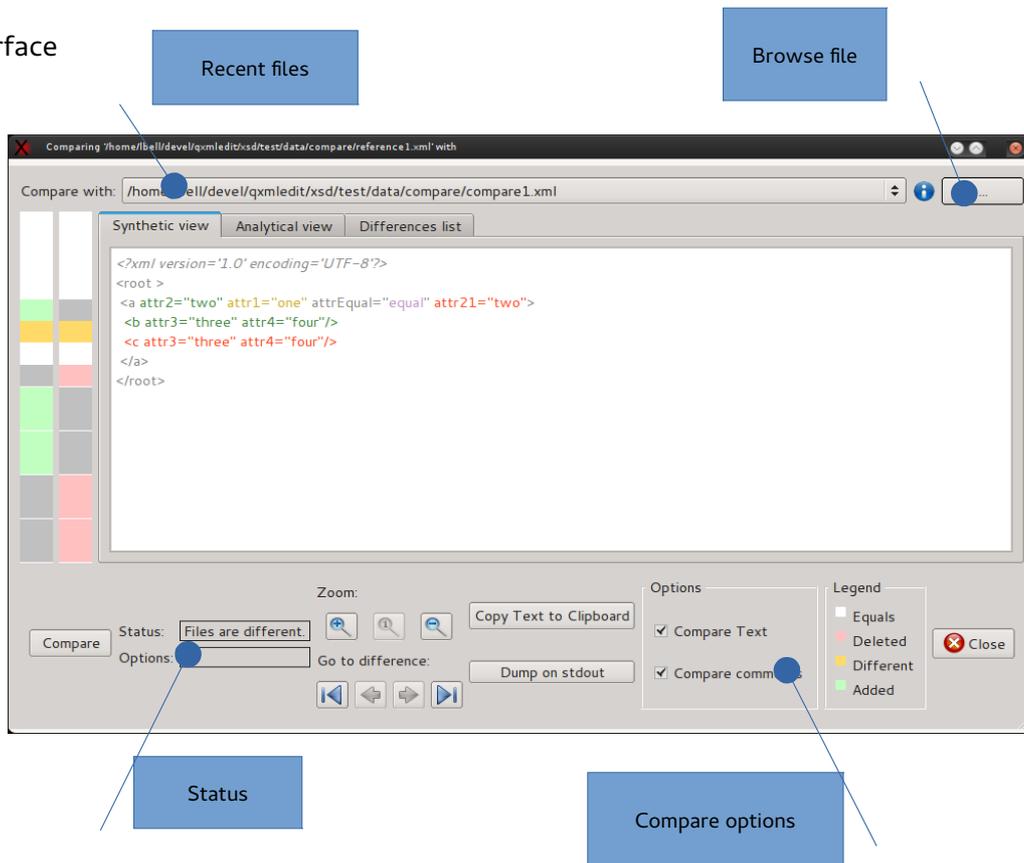
Analytical view
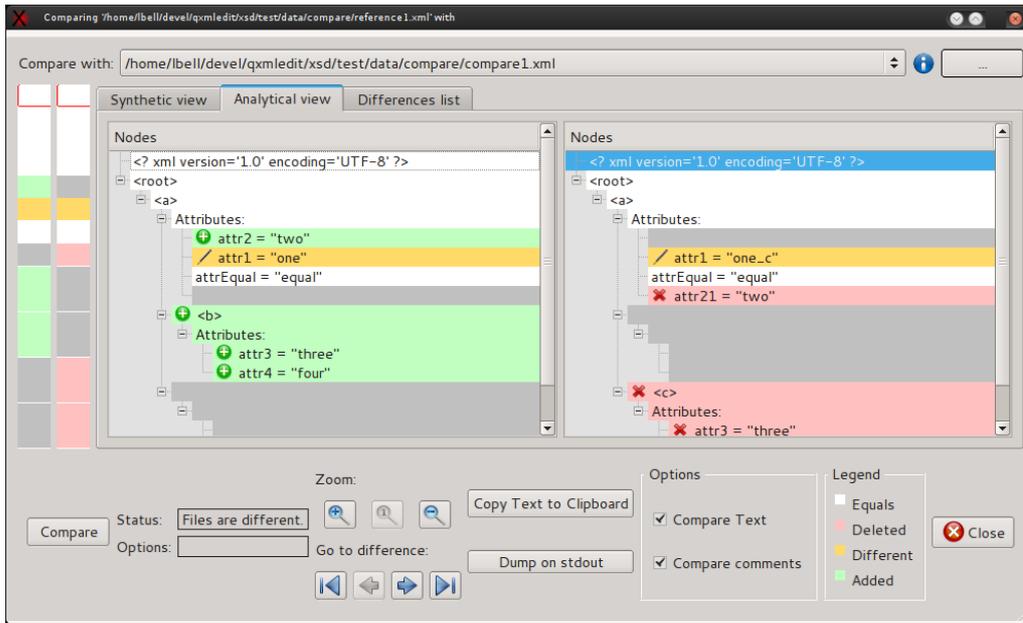
Differences list
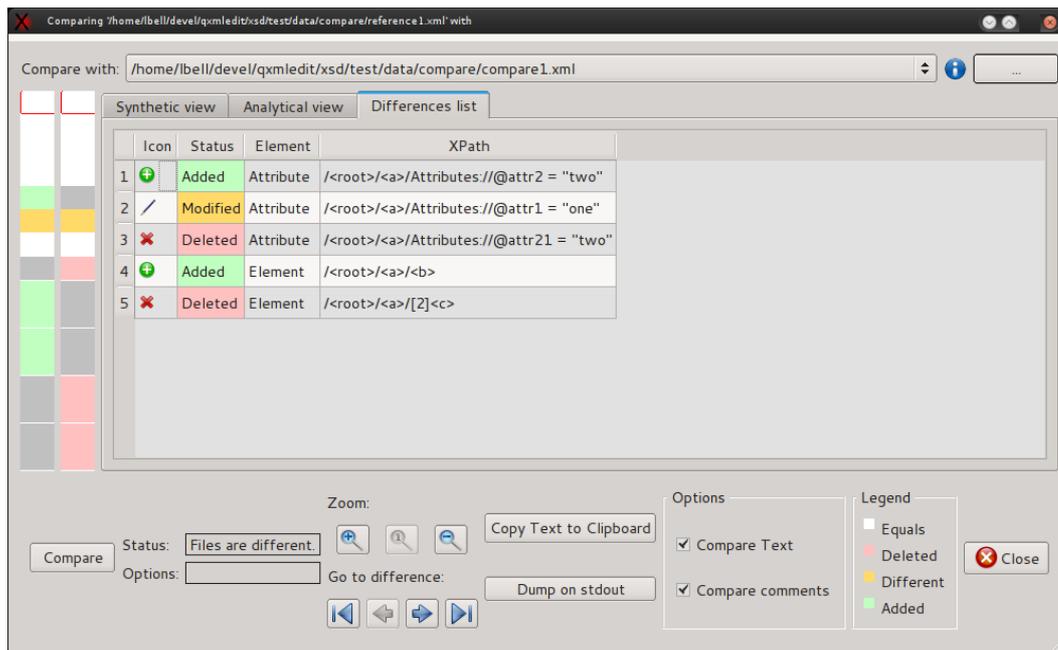
Working With Snippets

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.
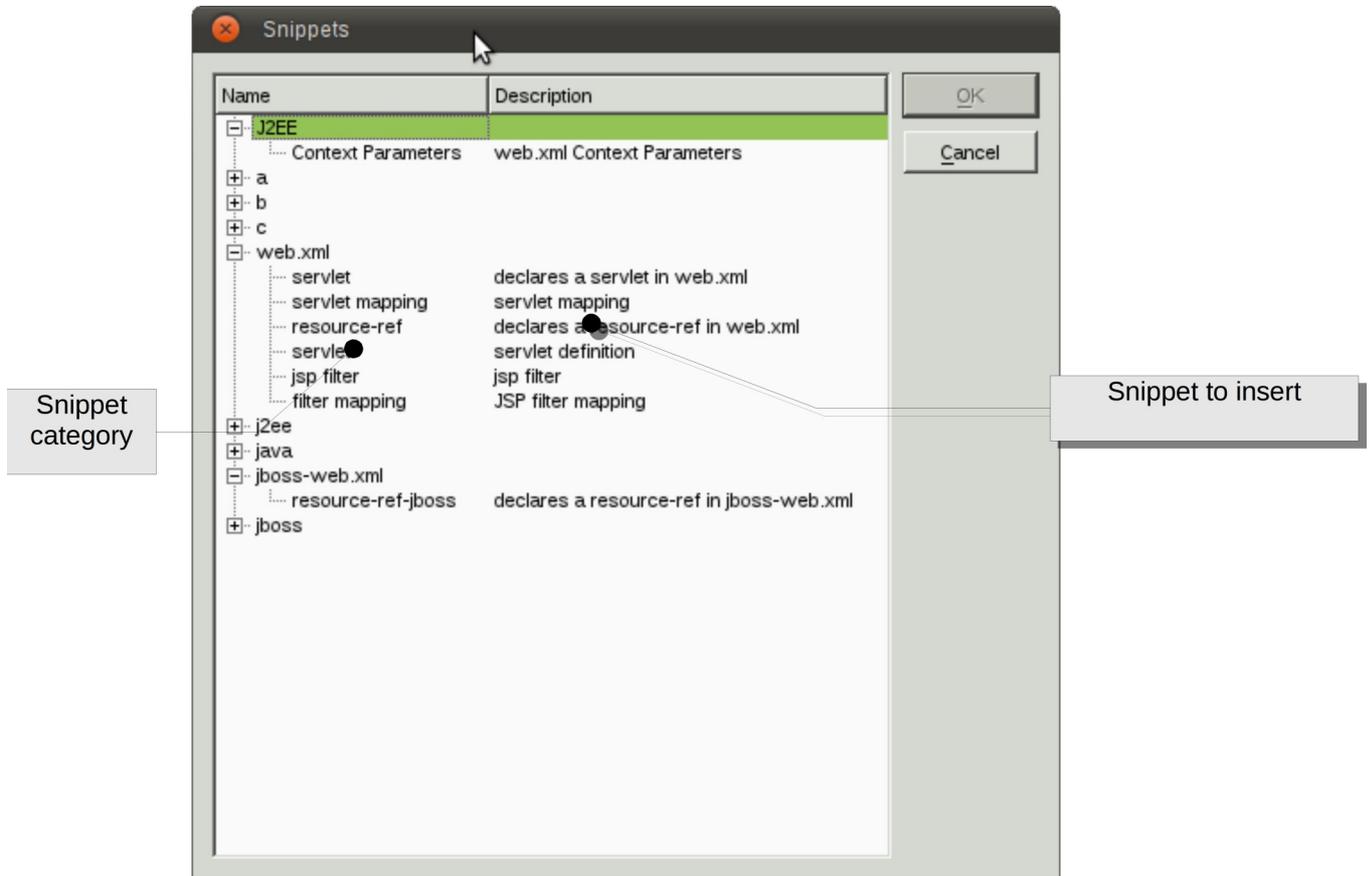
To edit the snippets and their categories:

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

Edit XML Snippet

Name:

servlet mapping

Description:

servlet mapping

Tags (comma separed:)

j2ee,java,web.xml

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>
```

OK

Cancel

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

- ☑ Project: `my project`
- ☑ Name `name of the file`
- ☑ Copyright: `(c)`
- ☑ Version: `1.0`
- ☑ Domain: ` `

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:
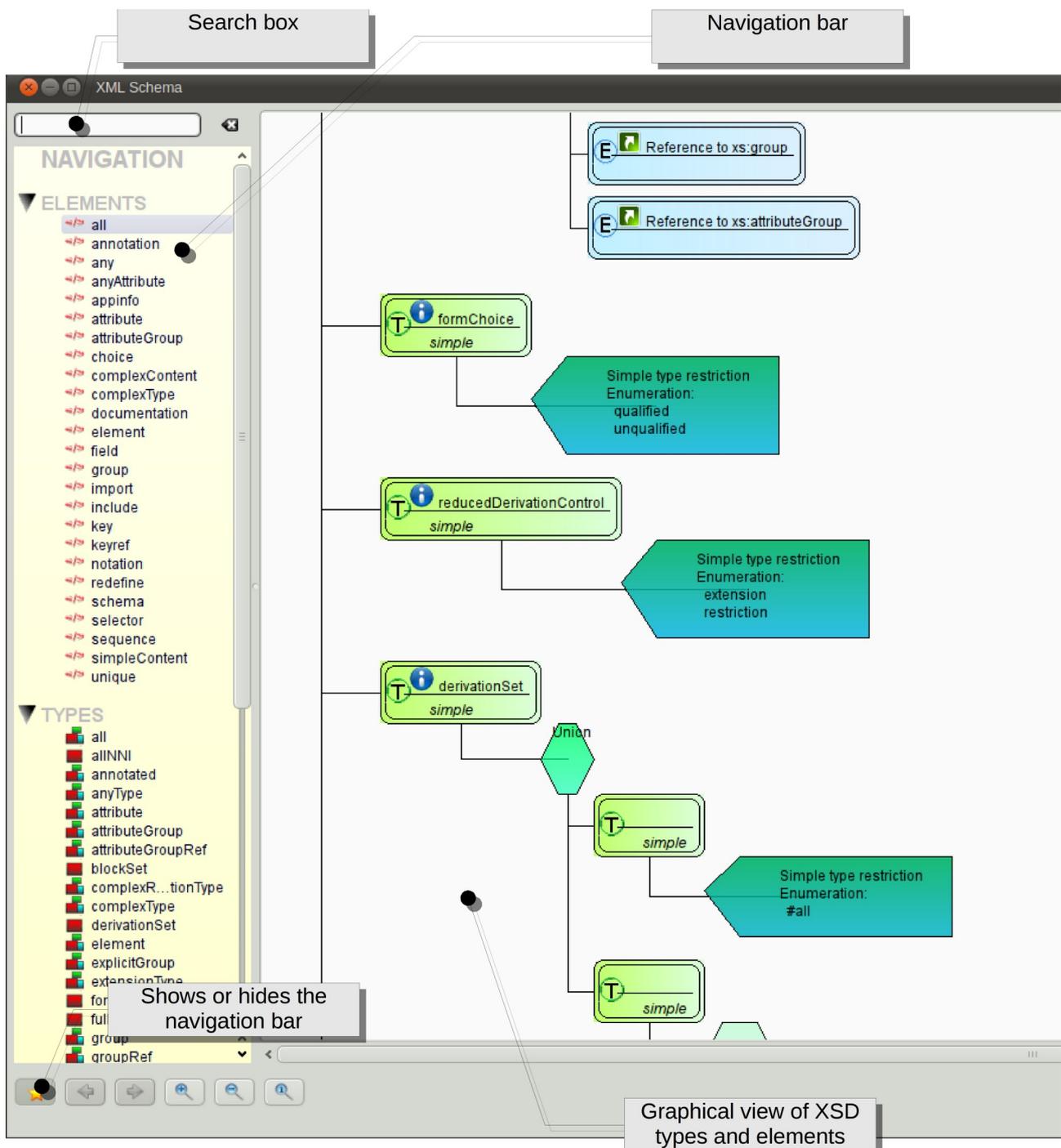
The symbols denote:

- **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.

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...***
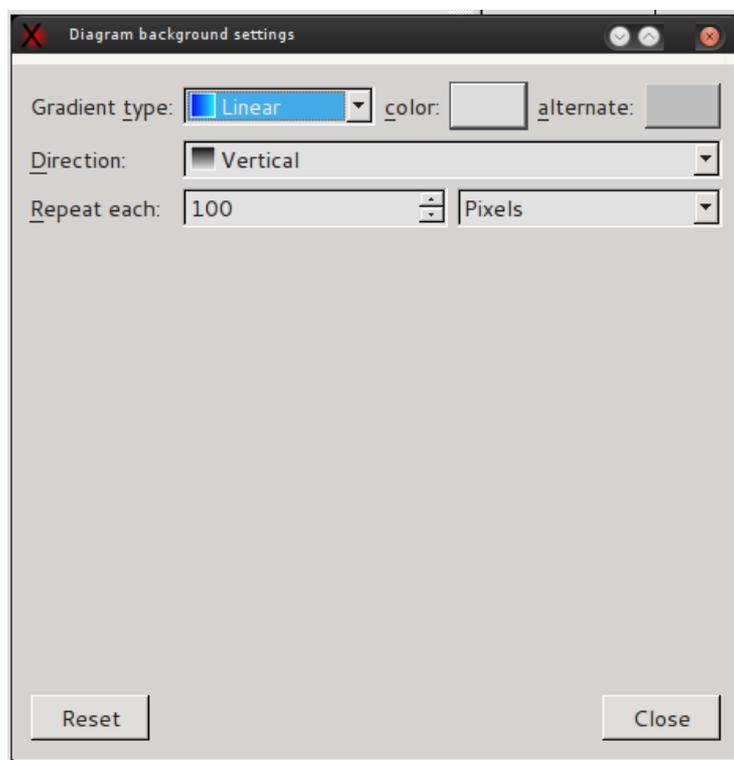
Exporting XSD Graphical view

In the window there are buttons that permit to export the view as PDF or SVG file.

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: **Q** . 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:
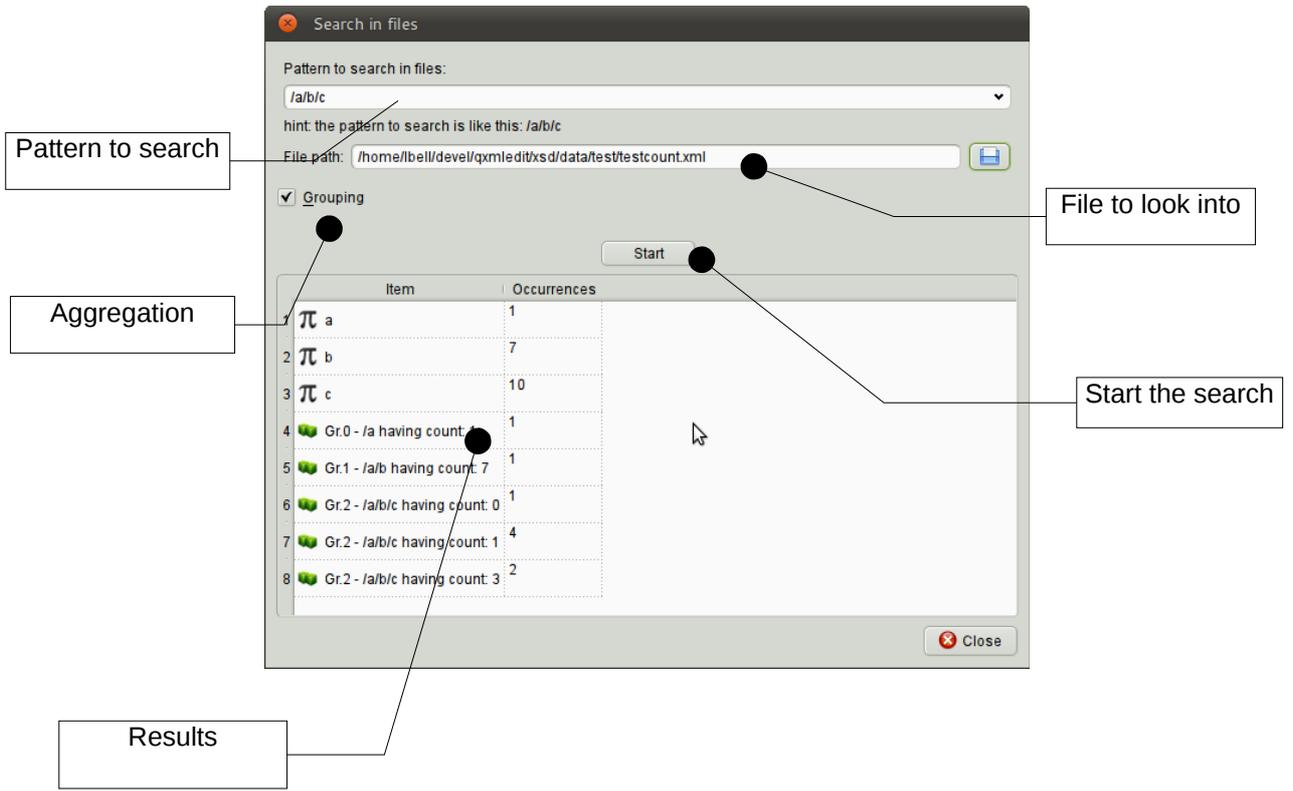
In this window you can insert the following informations:

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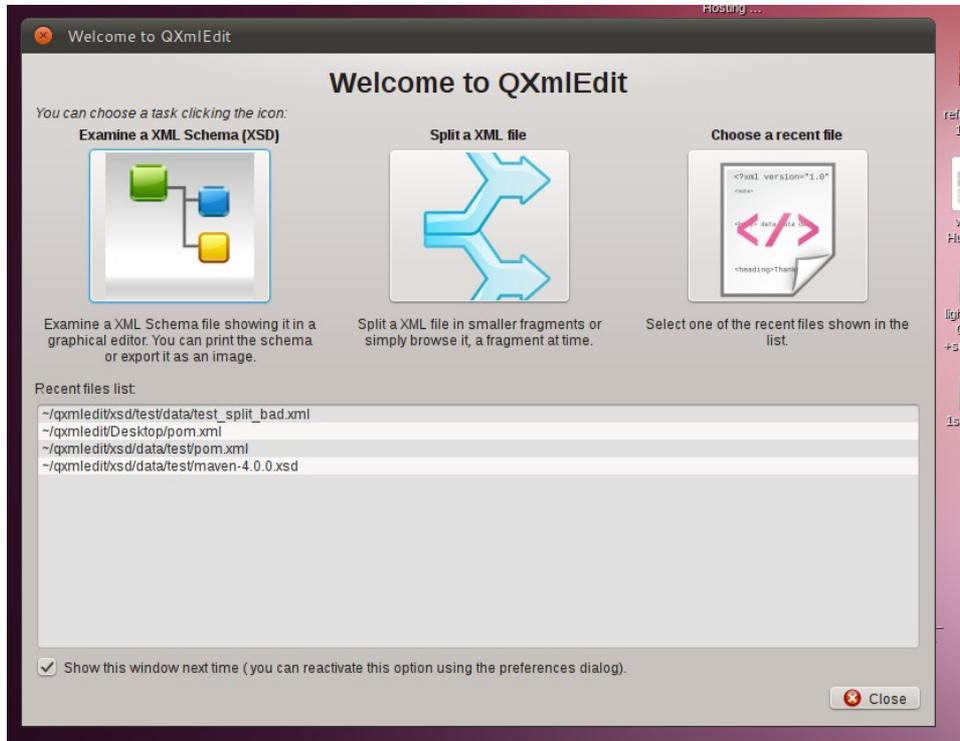
When starting, the program will open a welcome dialog leading to the most common operations. Choose "**Split a file**". If you choose to hide the dialog next time the program restart and change your mind, you can reactivate it in the configuration panel. You can access this feature also from the menu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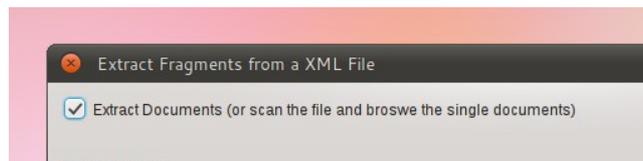
Decide if you want to extract information or browse it

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.
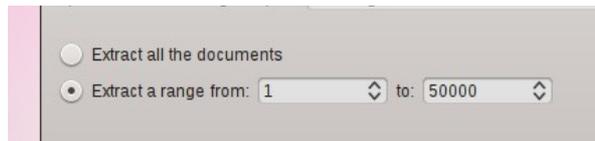
Limit the extracted fragments number

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.
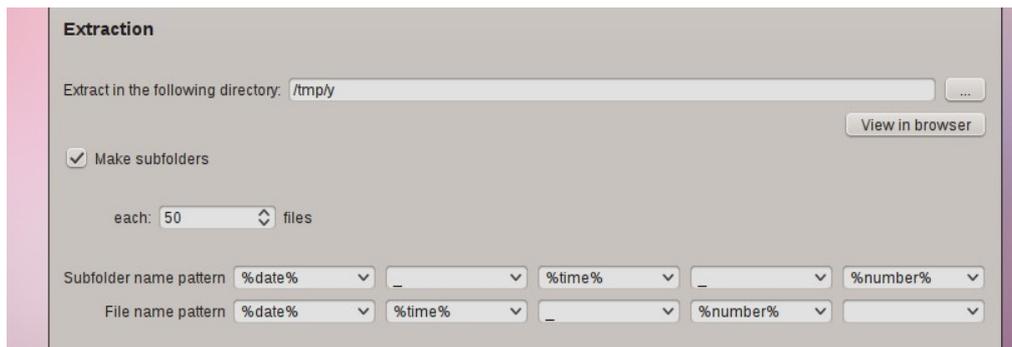
Decide the output folder and the naming

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.
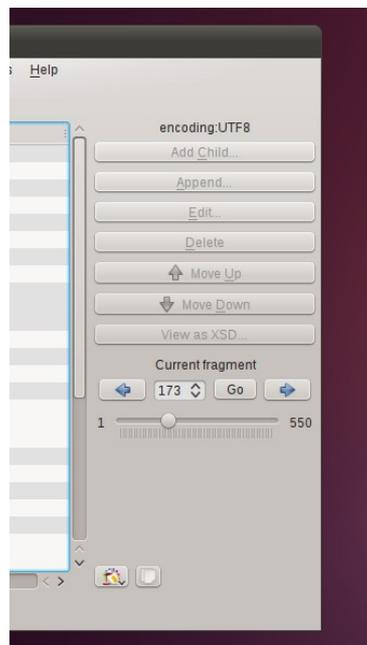
Go and examine data

After the start of the operation, in the 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.
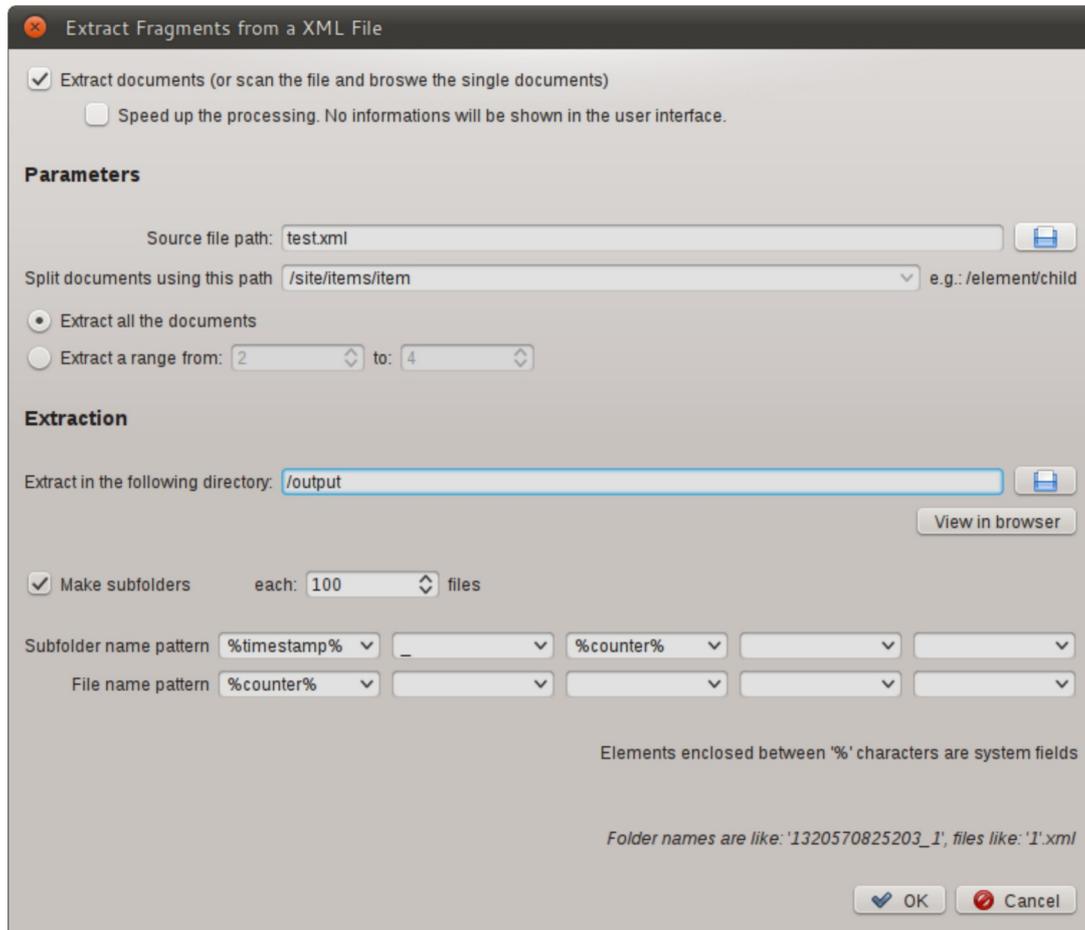
Illustration 1: The Extraction Panel

Information in this panel:

Information	Description
Extract documents (or scan the file and browse the single documents)	If selected the fragments found in the input file will be written to files as XML documents, one per fragment
Speed up the processing. No information will be shown in the user interface.	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.
Source file path	The path of the input file
Split documents using this path	The XPath used to find the XML fragments in the input file. The path must be entered using a syntax like this: */root/element/element*
Split using the depth	Use the depth of the element as criterion. Root element has depth one. Note: the elements will be extracted independently from their tag.
Extract all the documents	If selected causes the extraction of all the documents found in the input file.
Extract a range	If selected, only the documents that fall in the specified range will be extracted.

Information	Description
Extract in the following directory	The destination directory for the fragments.
Make subfolders	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.
Subfolder name pattern	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.
File name pattern	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.

Token that can be used in the name pattern section:

Information	Description	Example Output
%counter%	The counter of the object created. This is an unique number that can be used to identify the object	1
"%date%"	The current date in the YYYY_MM_DD format	2011_10_07
"%time%"	The current time in HH_MM_SS_millis format	10_43_30_560
"%timestamp%"	The current date and time in a numeric format	123456789
"%space%"	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
- Vrule

- Hrule

- TextInput

- ComboBox

- TextArea

- Paragraph

- CheckBox

- RadioButton

- Tree
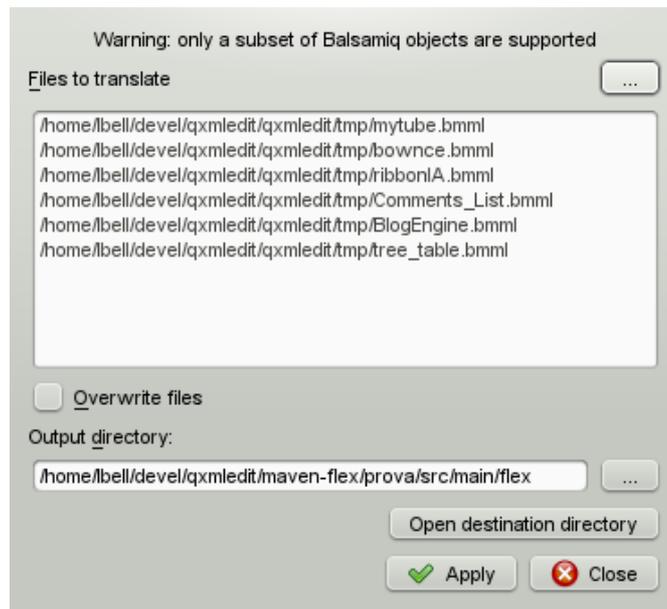
Life With Sessions

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 informations.
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.
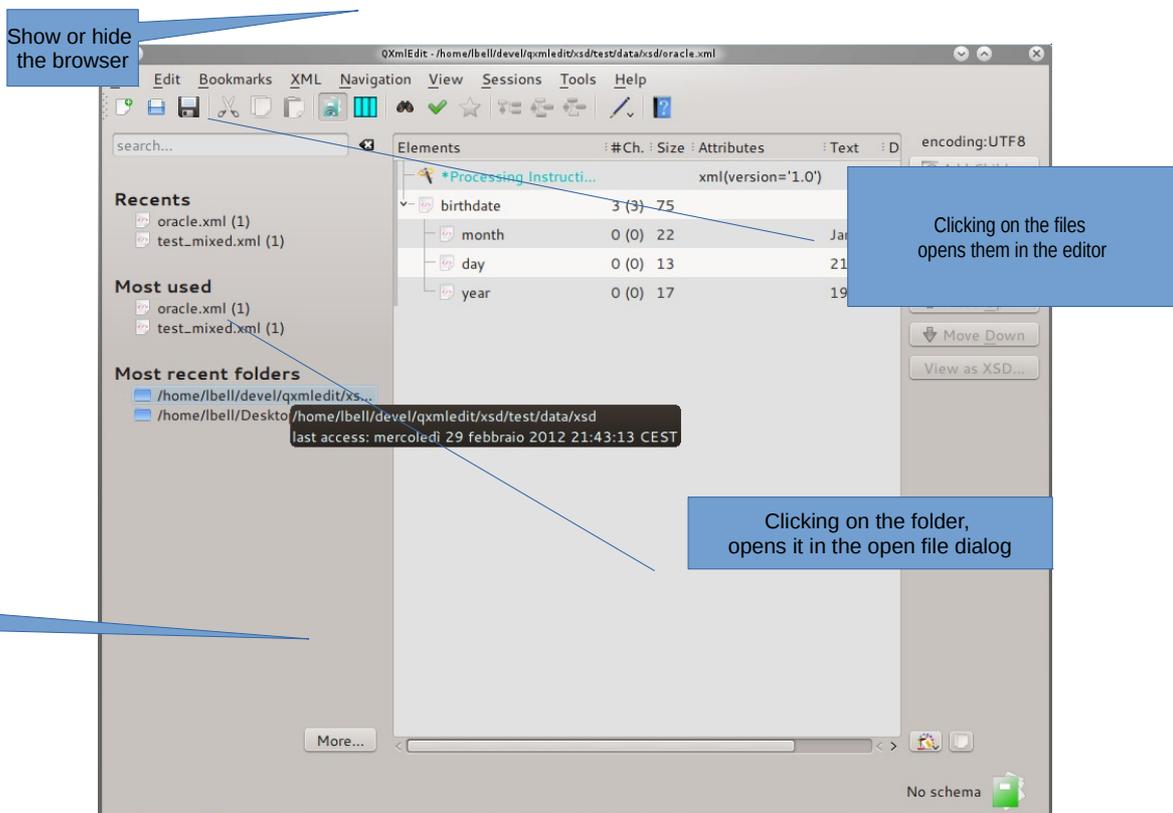
Sessions Management Dialog

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
Activate	Activates the selected session, closing automatically the current one, if any.

| New	Creates a new session.
Edit	Opens the session detail dialog of the currently selected session.
Delete	Delete the currently selected session.
Edit file	Loads in the editor the file selected in the lower pane.
Export file data	Put in the clipboard the path of the files.
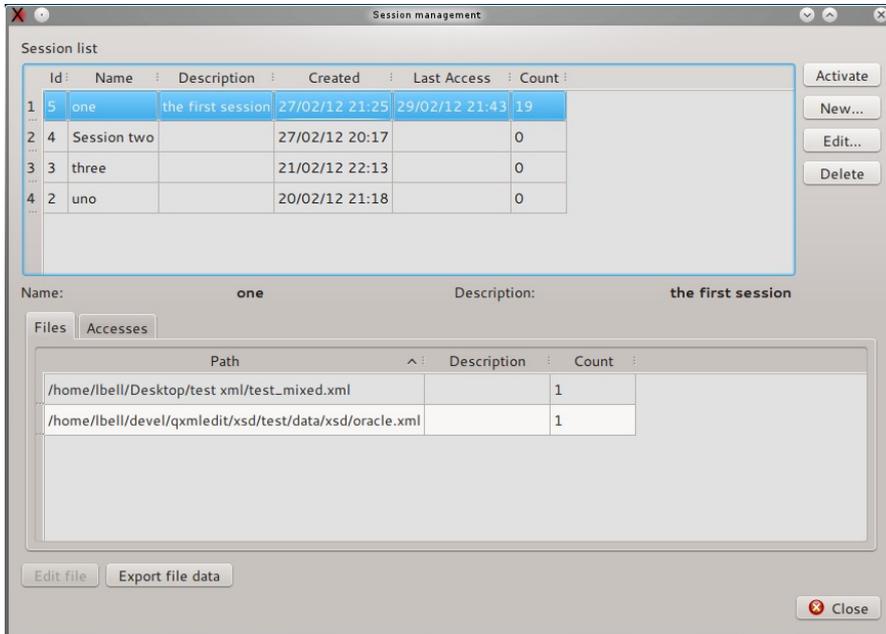
Session Properties

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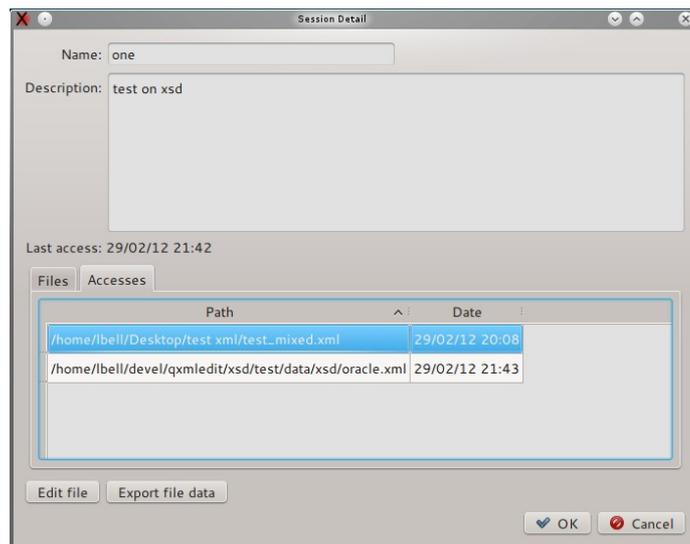
You can set or change the session name and description. The list of files can be copied in the clipboard. Double clicking on a file loads it in the editor.

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
Delete all sessions data	Completely erase the sessions data.
Delete data older than…	Delete the data older than the specified date. The data are deleted as follows: - all the accesses to a file registration older than the specified date. - all the sessions that have last access date older than the specified date and have no access data. If the **delete files information** 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.

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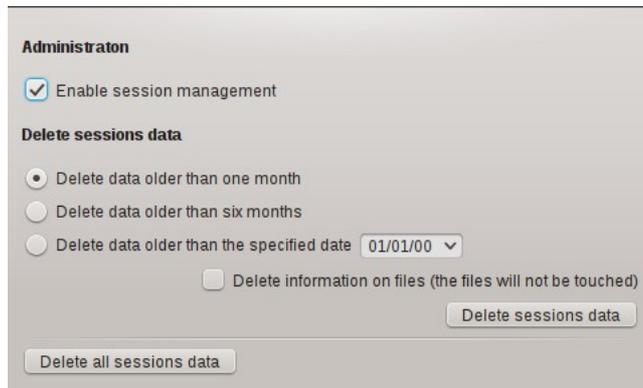
The session configuration panel allows to enable or disable session handling. There is the possibility to clean session data as explained in details in the paragraph "Managing sessions data".

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
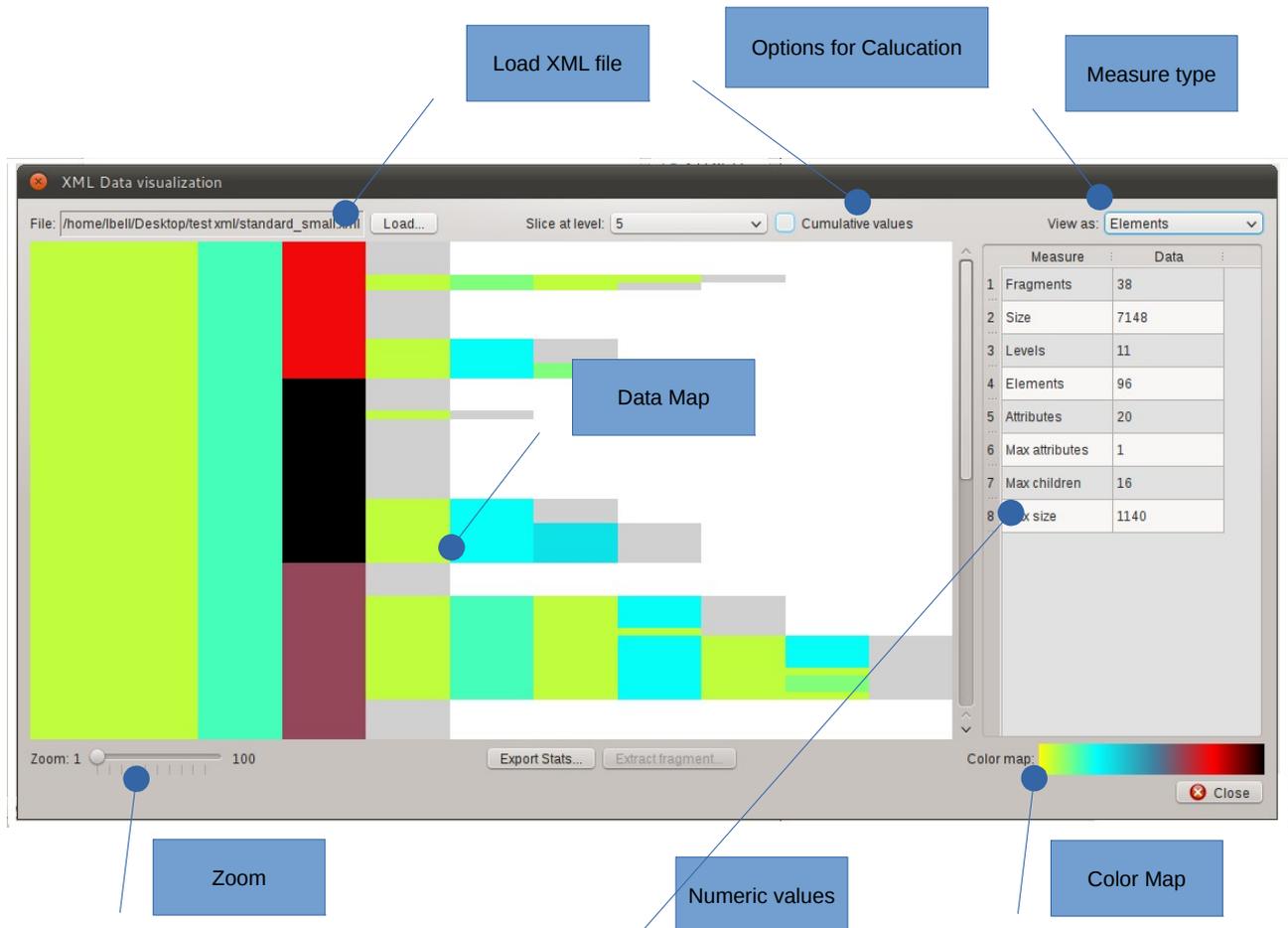
To access to the Visualization use:

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 eventually the zoom level.
6. Explore data hovering with the mouse on the interest points, a tool tip will appears.
7. Use the context menu (right mouse button) to export data.

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

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 distribute evenly in the view.

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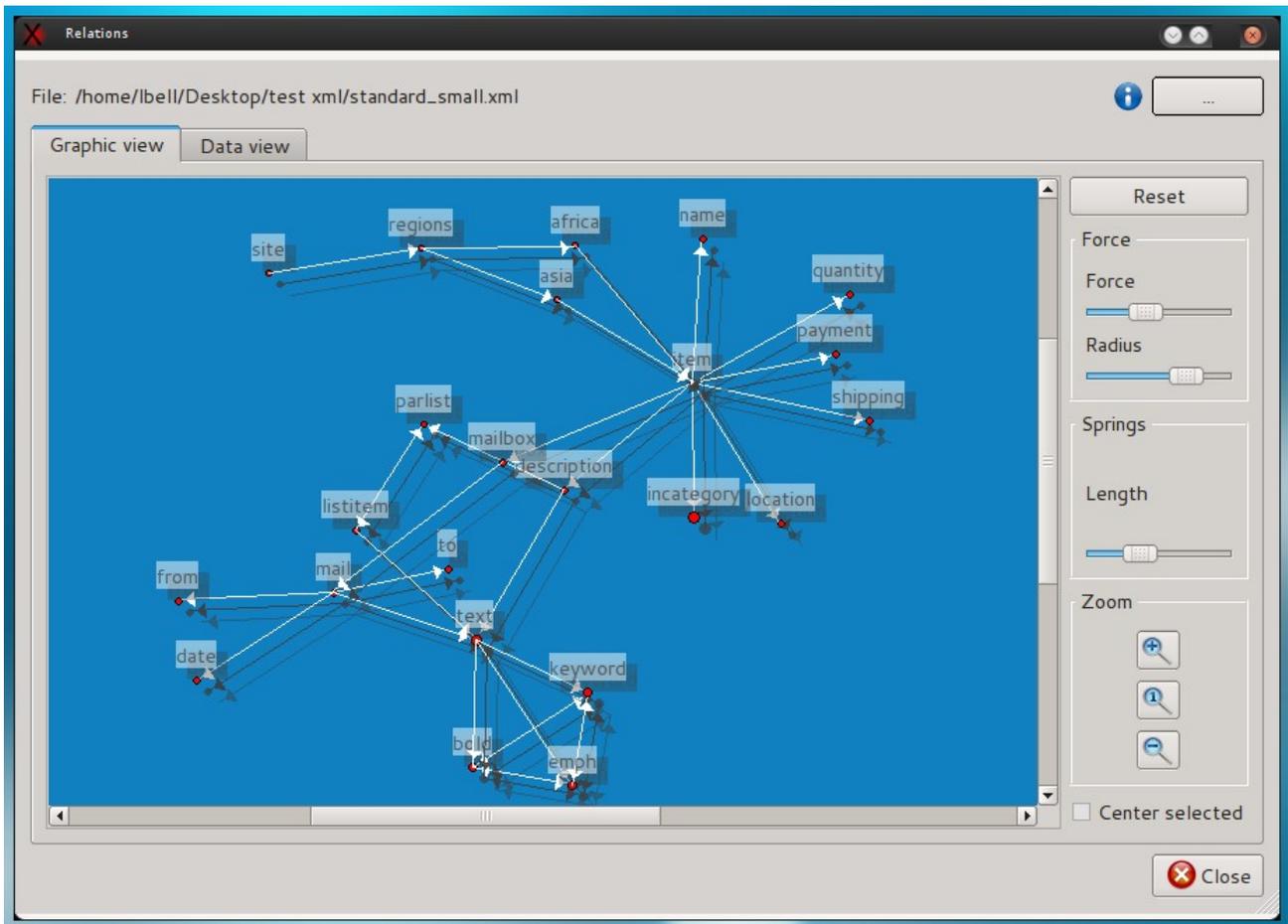
Options:

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"**

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.
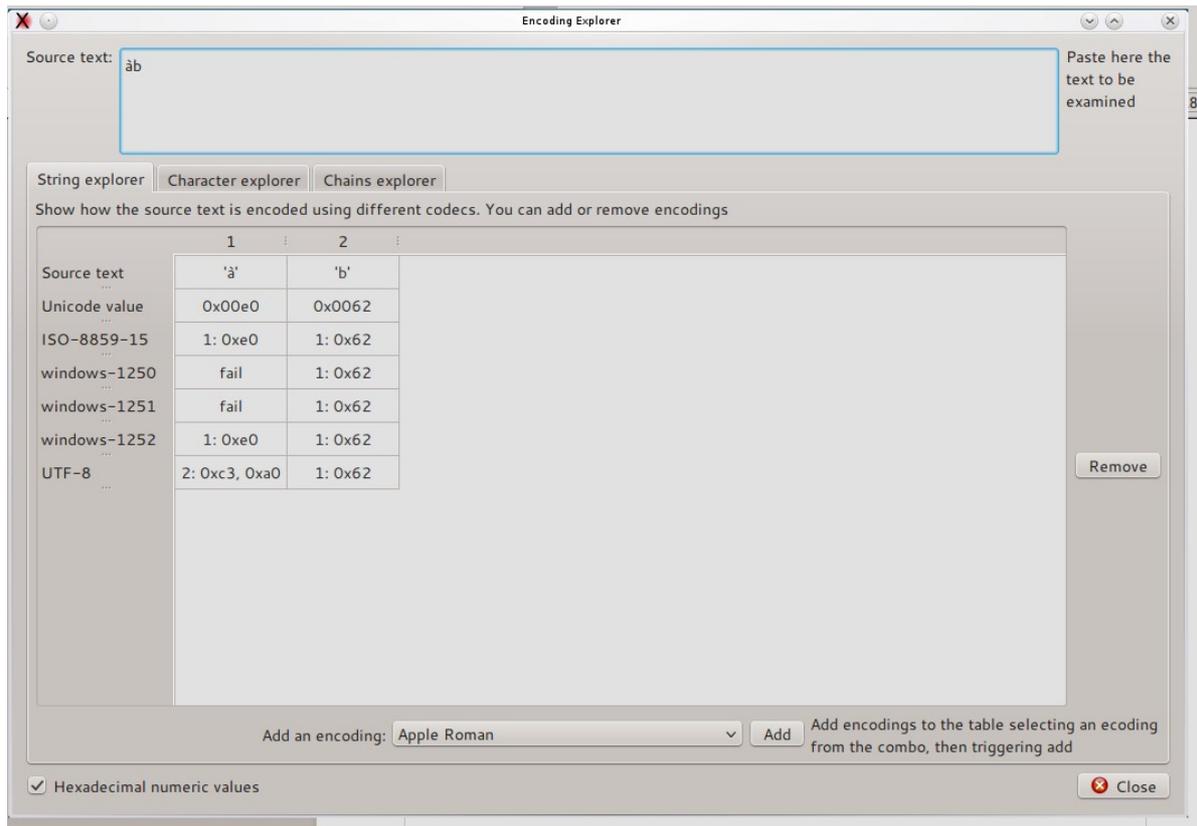
Explore how binary values are recognized by different encodings

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

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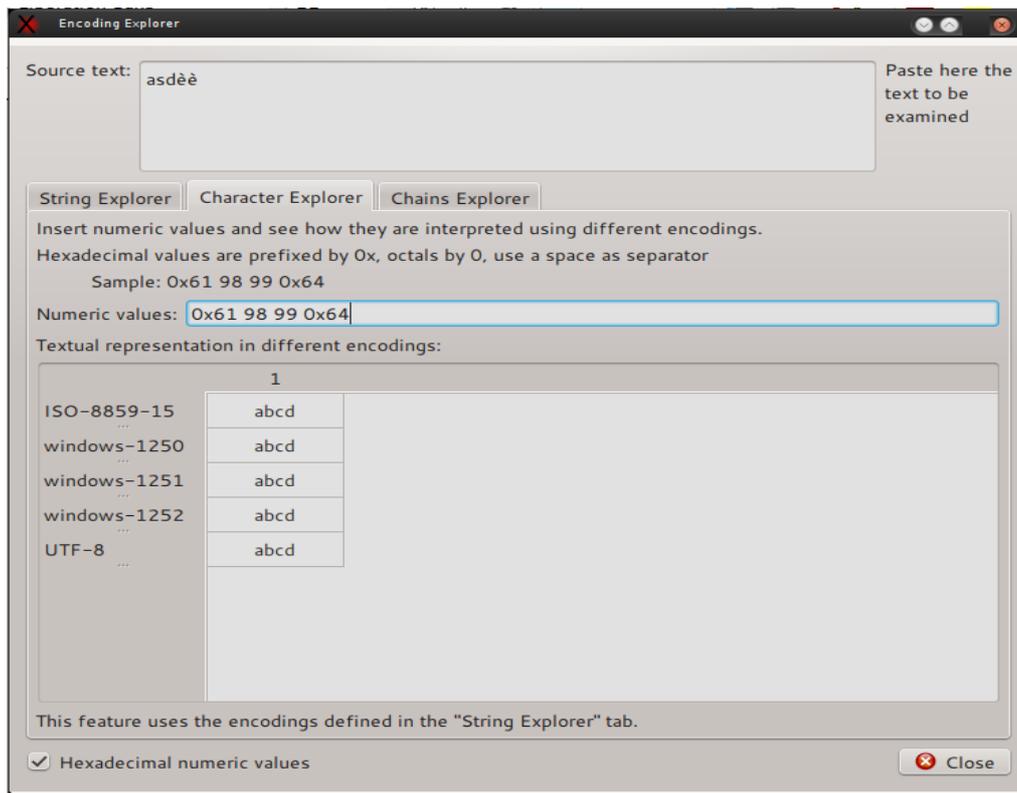
Explore how a string will be translated when written in some encoding and read in another one

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.

Encoding Explorer

Source text: | asdèè| | Paste here the text to be examined

String Explorer | **Character Explorer** | **Chains Explorer**

1   text written using the encoding   windows-1252 ⌄

	1
Source text ...	asdèè
Binary Values ...	0x61, 0x73, 0x64, 0xe8, 0xe8

2   read using this encoding, then written again using the encoding   UTF-8 ⌄

	1
Source text ...	asd�
Binary Values ...	0x61, 0x73, 0x64, 0xef, 0xbf, 0xbd

3   read using this encoding   windows-1250 ⌄

	1
Source text ...	asdďż˝
Binary Values ...	0x61, 0x73, 0x64, 0xef, 0xbf, 0xbd

☑ Hexadecimal numeric values   ⊗ Close

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.
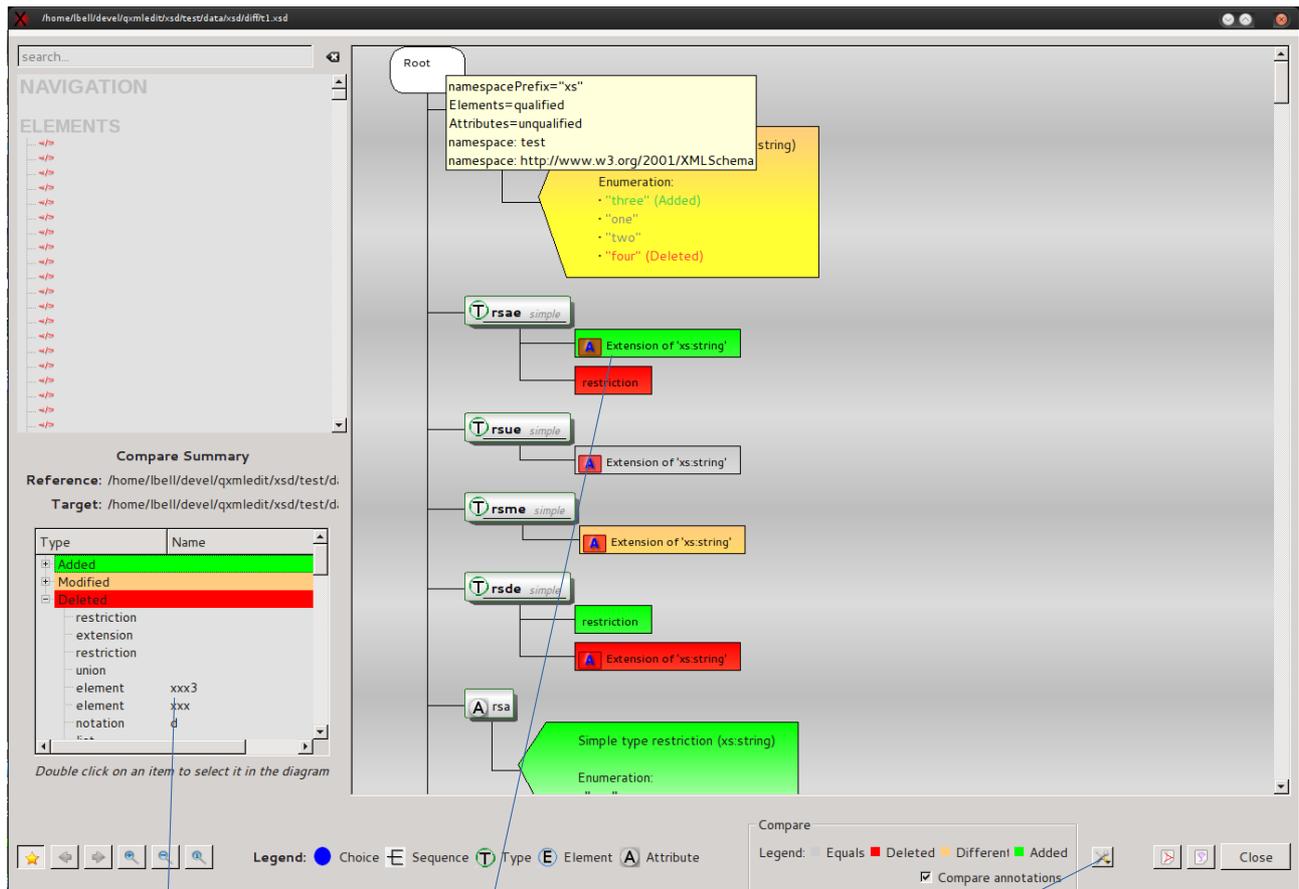
Tools menu

Items, with different colors depending on state

Differences report

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.

Appendix

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:	• "name" style name as shown in the user interface • "description": a description • "namespace": an optional namespace	• "keywords" • "styles" • "ids"

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	• "id" unique identifier of the style. It is a string • "color": hexadecimal representation of a color used to paint the text. Example: "FF0000" • "backColor": optional, the background color in hexadecimal format with optional alpha at the end. Example: FF0000 or FF000080 • "family": font family if different from default. • "size": font size if different from default. • "bold" set to 'true' or a numeric value different from zero to force bold style on font • "italic": same of bold, but for italic style The only mandatory attribute is id, the others are activated if set.	

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	• "keyword" the name of the element to mark • "idStyle": the identifier of the associated style	

Element "ruleSet"

Contains a set of "rule" and "ruleSet" using implicit AND connector

Tag name	attributes	Child elements

ruleSet	• "connector" the connector of the children
	  ○ and (default)
	  ○ or
	• "idStyle": for a top level rule identify the style to be used
		"rule", "ruleSet"

Element "rule" (inner)

Tag name	attributes	Child elements

rule	• "entity": denotes and attribute or an element, default: attribute
	  ○ attribute
	  ○ element
	• "name": if attribute, the name of the attribute or element
	• "op": denotes the comparison operator
	  ○ EQ: equals
	  ○ NE: not equals
	  ○ EX: value exists
	  ○ NX: value non existent
	  ○ GT: greater than
	  ○ LT: less than
	  ○ GE: greater or equal
	  ○ LE: less or equal
	  ○ SS: substring (string only)
	  ○ ST: starts with (string only)
	• "type": type of comparison, optional, default: string
		none

	○ s: string ○ n: number ○ p: position "value": the comparison value. If position, the value should be: ○ numeric ○ first (literal) ○ last (literal) "case": if the comparisons take the case of the strings into account, default: true ○ true: case sensitive ○ false: not case sensitive	

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	"id" name of the attribute that is considered as an identifier. "alpha": if true, force the enclosing of the value between quotes.	

This is a complete sample style file:

```xml
<?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>
 <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.