QXmlEdit
User Manual

A guide to the examine XML data
with QXmlEdit.

Version 0.6.1

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:

- Configurable elements and attributes display.

- Fast XML edit and easy navigation.

- Handling of base 64 coded text and XML inserted as data.

- Powerful search.

- XML snippets.

- XML compare utility.

- XML Schema viewer.

- Flex code generation from Balsamiq source

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 5

The big changes respect to the version 4 are the XML Schema files viewer plug-in and a new working mode called 'explore mode' that loads the structure of big XML files without data.

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.
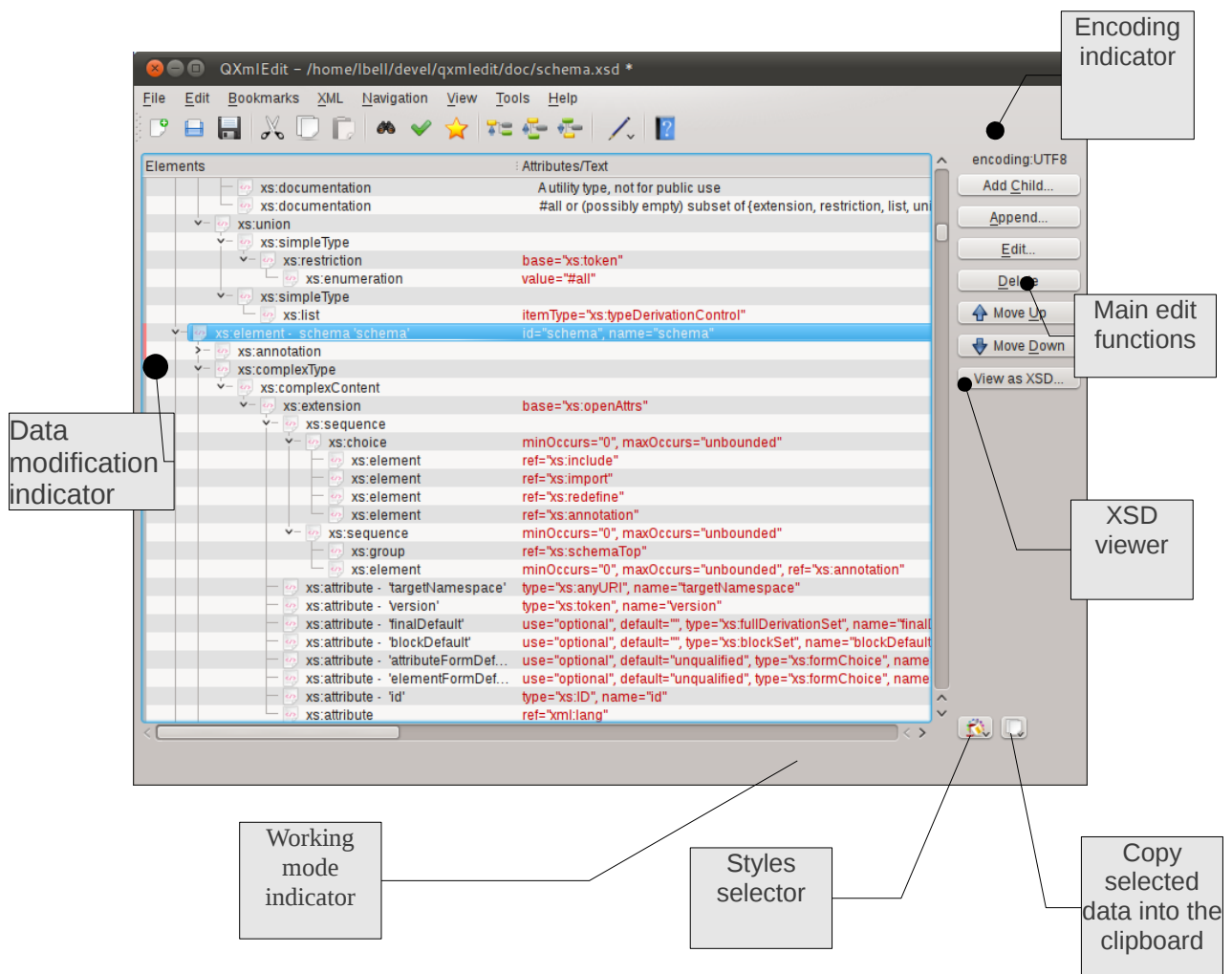
Main Functionality

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
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.**

Exploring Files

You can use the menu *File > Explore Structure* to load the only the structure of a XML File. No data will be loaded in memory, only elements definition, and you can't edit the data. When entering this operating mode, a small icon appears on the status bar (🔍). This mode is intended to explore a file that because of its dimensions, cannot be loaded in the normal mode.

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.

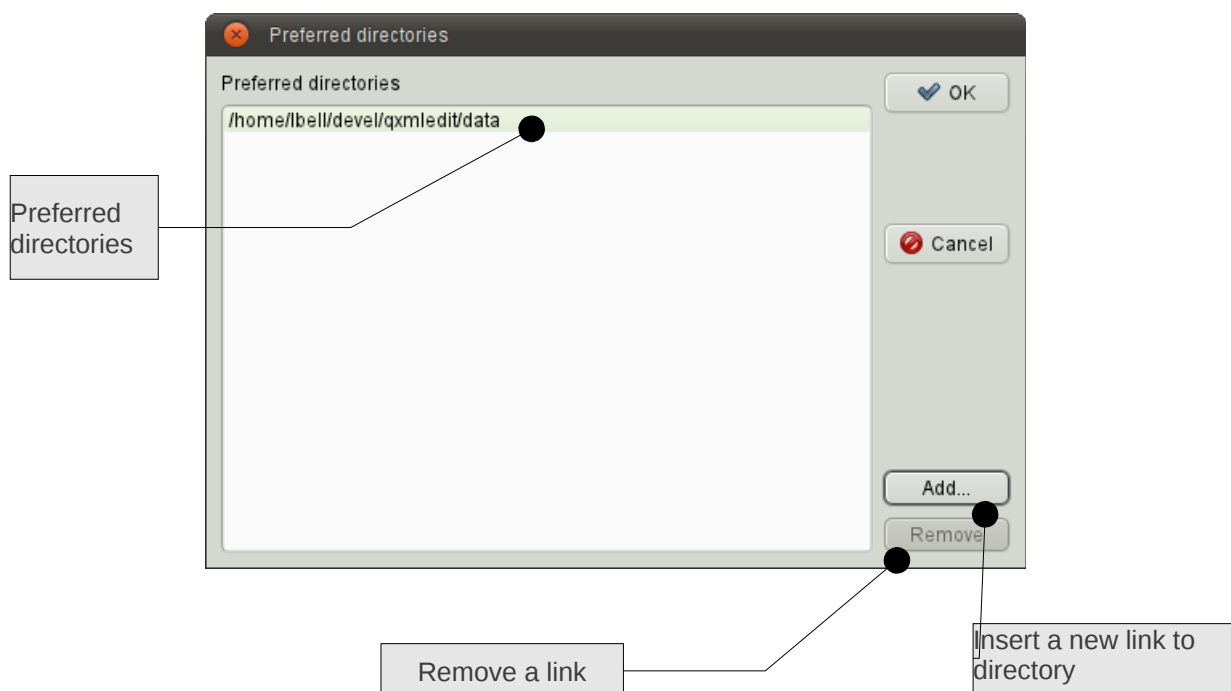To add the directory of the current open file to the preferred ones:

- 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**

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**
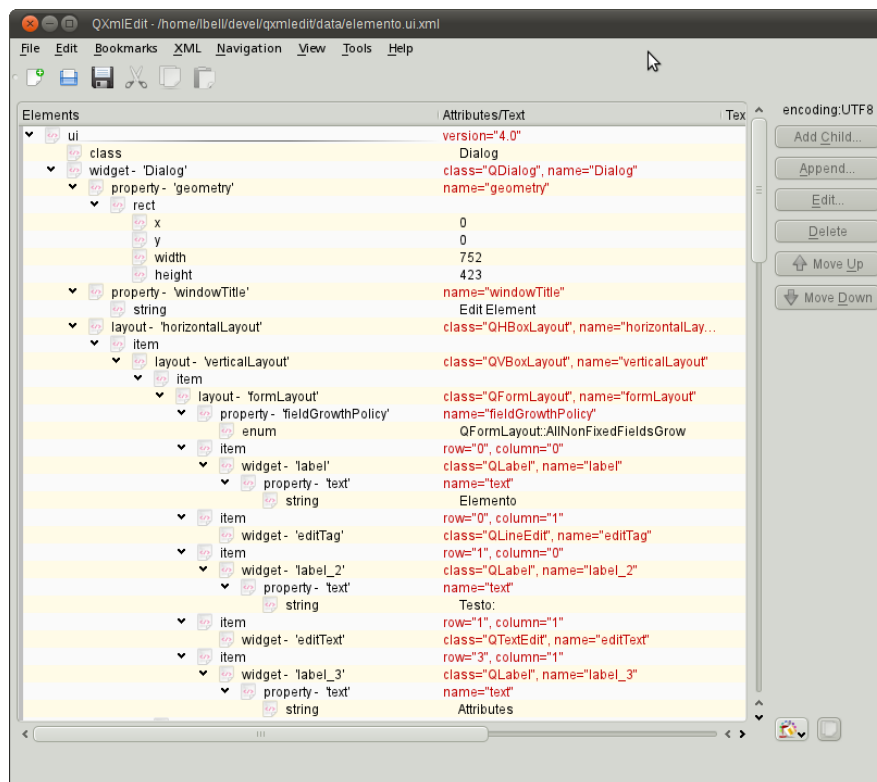
To show the attribute length near to their name:

- 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**

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 **Edit > Configure** menu.

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:

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:
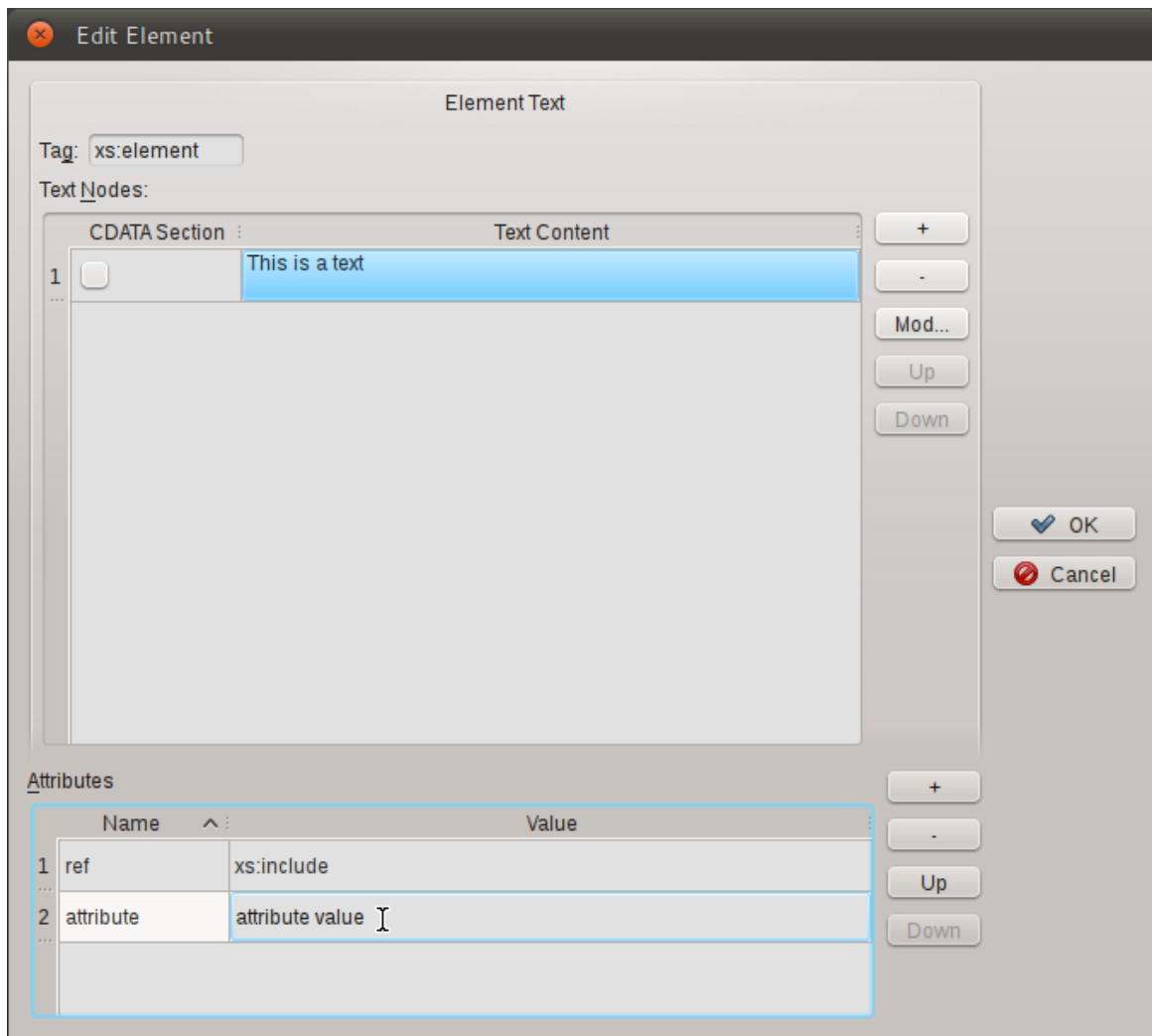
In the panel you can insert the following information:

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:
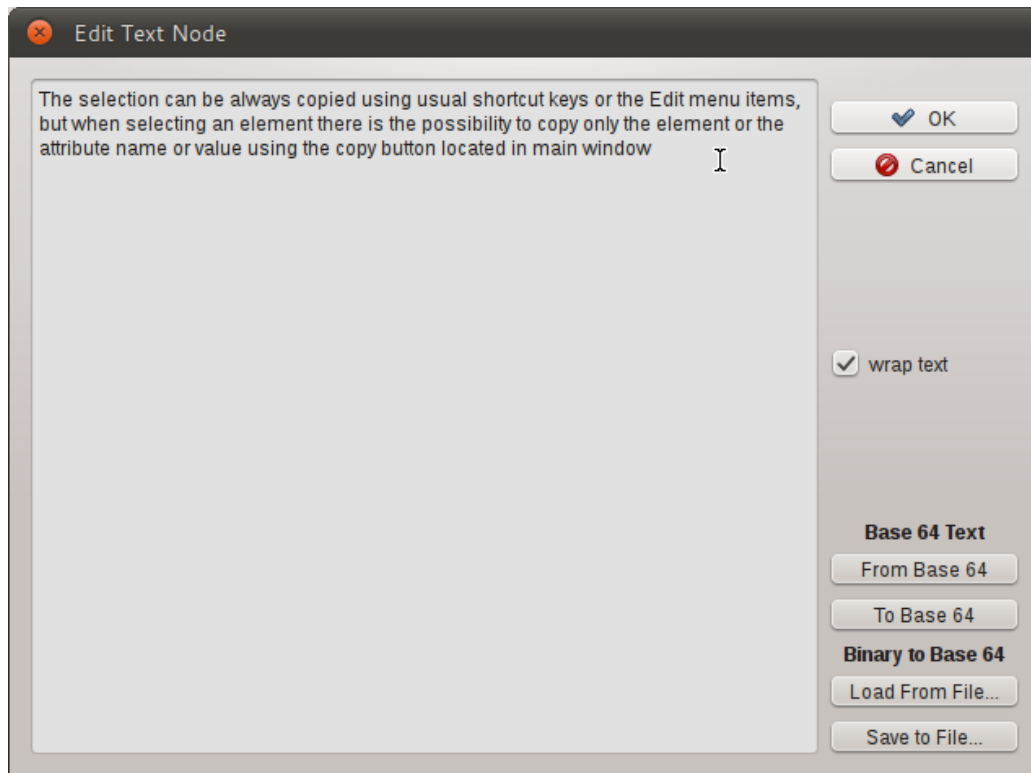
- Activate the **wrap text** check box.

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:
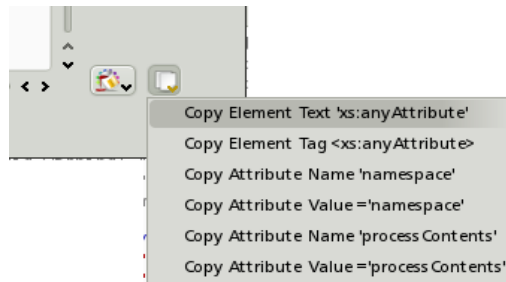
to copy the selected element tag:

- 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.

Validating a XML Document

A 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 a 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
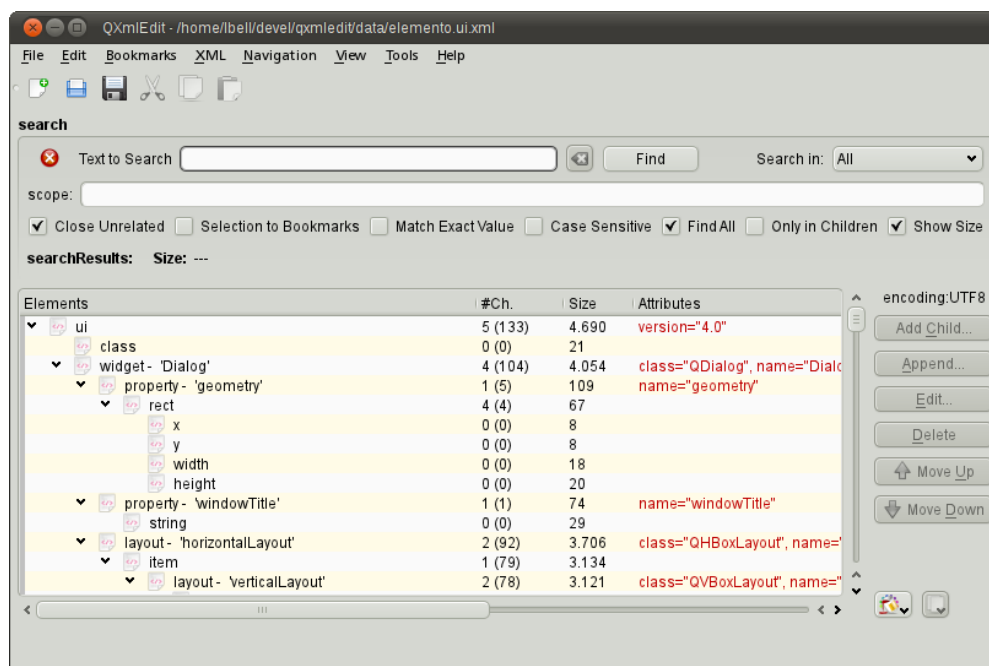
In the panel you can do the following operations:

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 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.

Matched items
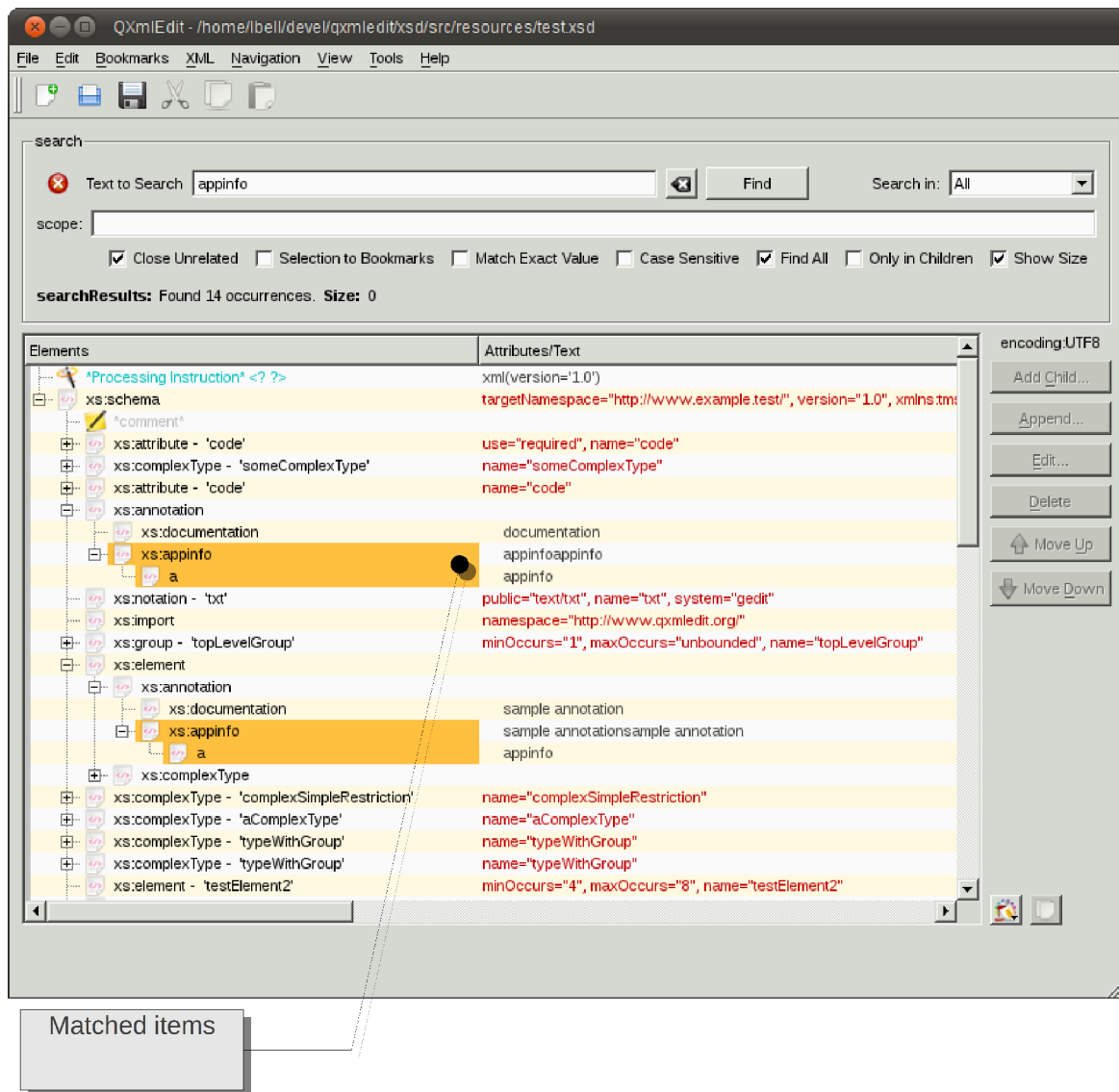
Comparing Files

The purpose of the comparison as implemented is to give a compact view of changes, for that reason the output is not something you expect from a source merge program (for that, you can use a specialized merge program), but something to immediately recognize what attribute values are changed, if any.

To compare a file against the current:

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

In the panel that opens you have the following options:

*   Show the old values instead of the new ones in the difference view.
*   Compare elements only (**Ignore non elements** option)
*   Compare the XML structure only (**Ignore text**), ideal to have a bird view on the structure.
*   Copy the comparison result to the clipboard.

- Retry the comparison, after changing some option.

In the following screen shot you can note that "id" attribute of "keywords" element is different and that "data2" element has been deleted.
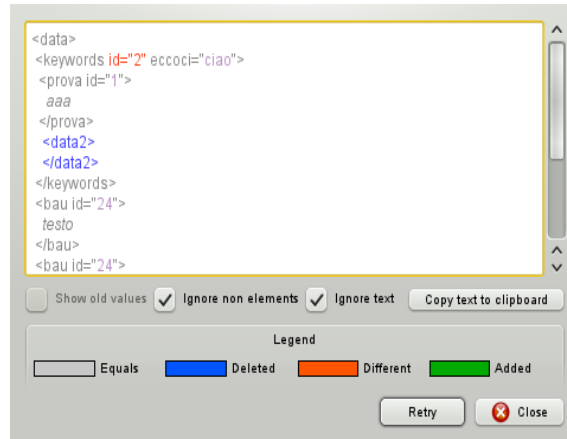
Working With Snippets

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 a 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.

Snippet category

Snippet to insert
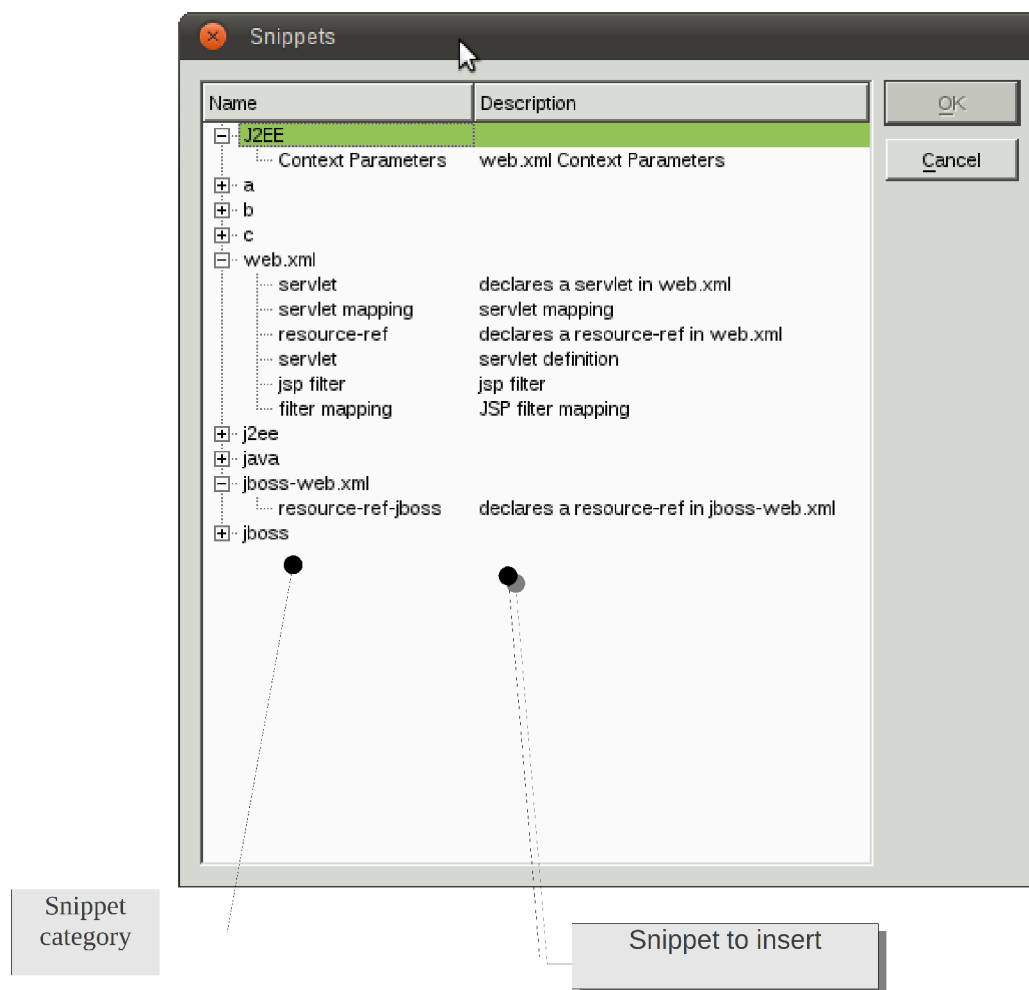
To edit the snippets and their categories:

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

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:

Search box

Navigation bar

XML Schema

NAVIGATION

▼ ELEMENTS
all
annotation
any
anyAttribute
appinfo
attribute
attributeGroup
choice
complexContent
complexType
documentation
element
field
group
import
include
key
keyref
notation
redefine
schema
selector
sequence
simpleContent
unique

▼ TYPES
all
allNNI
annotated
anyType
attribute
attributeGroup
attributeGroupRef
blockSet
complexR...tionType
complexType
derivationSet
element
explicitGroup
extensionType
formChoice
fullDerivationSet
group
groupRef

E | Reference to xs:group

E | Reference to xs:attributeGroup

T | formChoice
simple

Simple type restriction
Enumeration:
qualified
unqualified

T | reducedDerivationControl
simple

Simple type restriction
Enumeration:
extension
restriction

T | derivationSet
simple

Union

T | simple

Simple type restriction
Enumeration:
#all

T | simple

Shows or hides the
navigation bar

Graphical view of XSD
types and elements

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.

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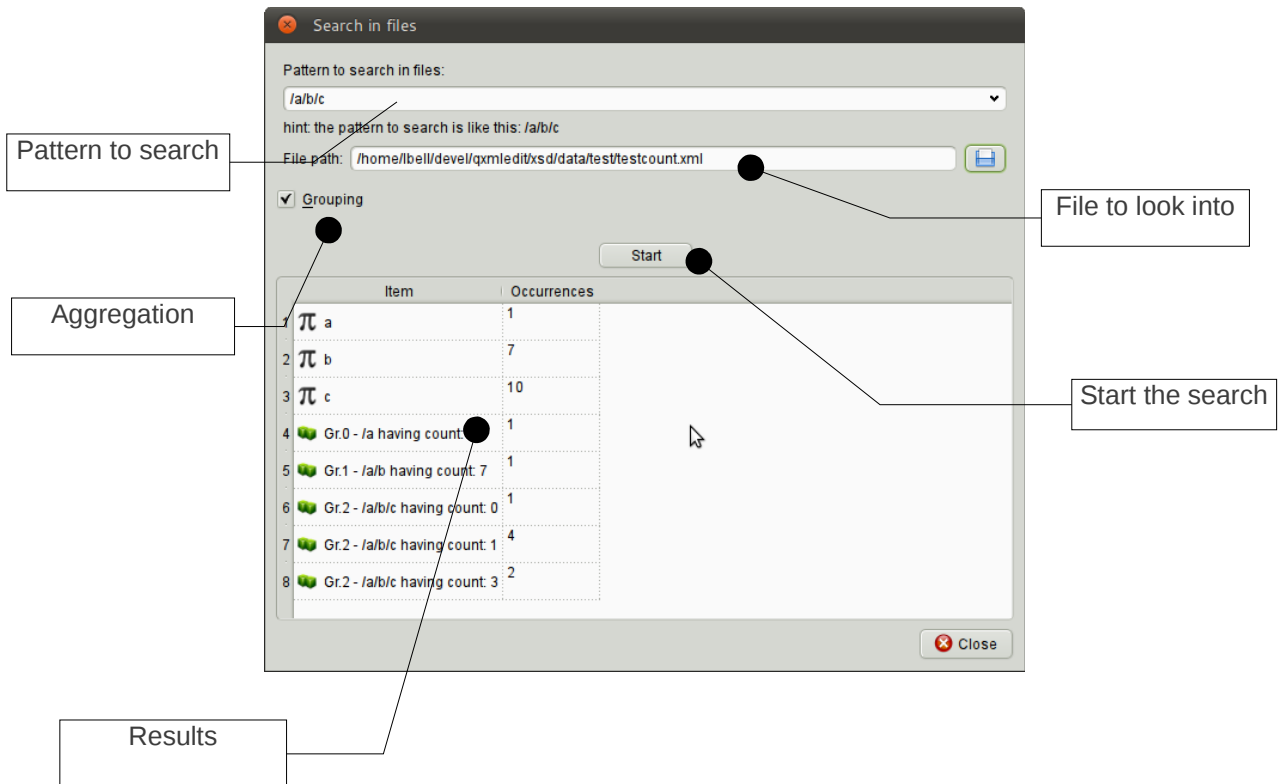
In this window you can insert the following informations:

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**.
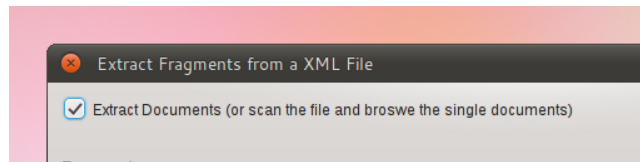
Decide if you want to extract information or browse it

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.
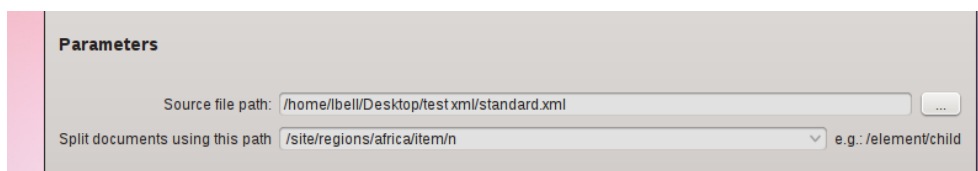
Choose the file and how to fragment it

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.
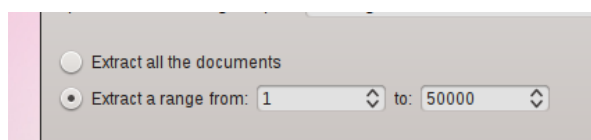
Limit the extracted fragments number

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.
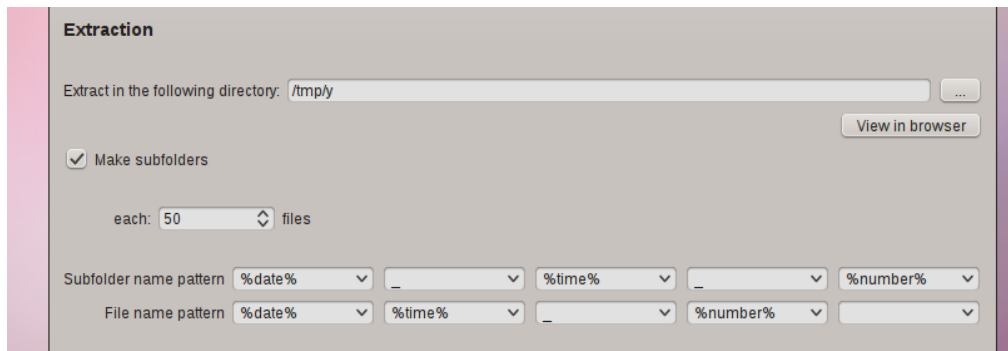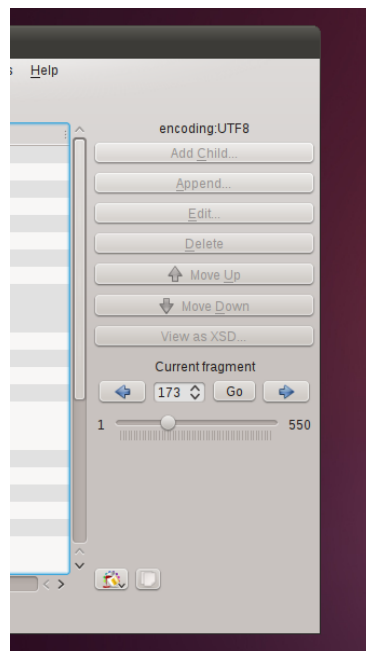

Decide the output folder and the naming

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.

Illustrazione 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*
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.
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

Information	Description
	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
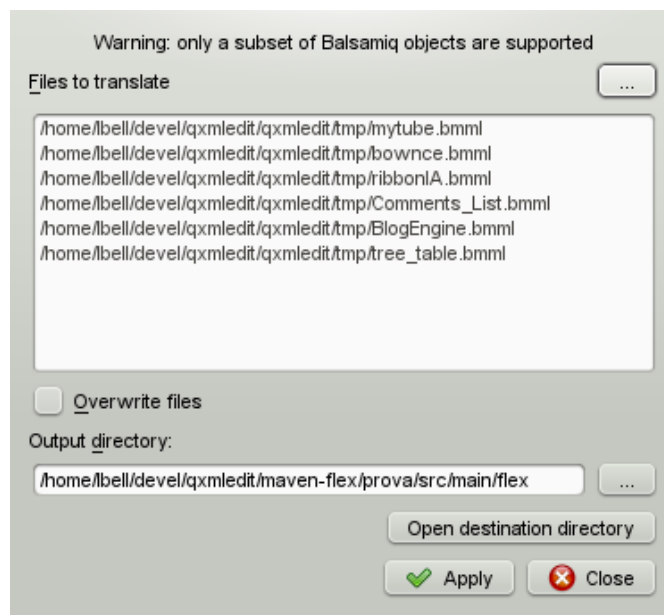
Appendix

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	• "keywords" • "styles" • "ids"

Element "styles"

This element is simply a collection of "style" elements

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" • "family": font family if different than 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" 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 "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 version="1.0" encoding="UTF-8"?>
<!-- this is a sample QXmlEdit style file -->
<style name="Bluish style" description="this is a simple style in blu">

<keywords>
 <keyword keyword="resource" idStyle="1"/>
 <keyword keyword="widget" idStyle="1"/>
 <keyword keyword="class" idStyle="1"/>
 <keyword keyword="layout" idStyle="2"/>
 <keyword keyword="window" idStyle="5"/>
 <keyword keyword="property" idStyle="3"/>
</keywords>

<styles>
 <style id="1" color="2080FF" />
 <style id="2" color="FF00FF"  family="Lucida" size="10" bold=""
italic="true" />
 <style id="3" color="C0C0FF"  family="Verdana" size="14" bold="true"
italic="" />
 <style id="5" color="0000FF" italic="true" />
</styles>

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