

# Xml Editor Add-In

## Version 0.1

by Matt Ward, [mrward@users.sourceforge.net](mailto:mrward@users.sourceforge.net)

## Contents

1 Introduction.....	1
2 User Interface.....	2
3 Deployment .....	6
4 ToDo.....	7
6 Issues.....	8

## 1 Introduction

This add-in provides extended xml support for SharpDevelop. Features include:

- Element, attribute and namespace auto-completion for configured schemas.
- Xml folding.
- Xml validation against a schema.

## 2 User Interface

### 2.1 Namespace Auto-Completion



On typing the '=' character after the xmlns attribute a list of the configured namespaces will be shown.

### 2.2 Element Auto-Completion



### 2.3 Attribute Auto-Completion



On typing a space character after an element name a list of attributes for that element will be shown.

### 2.4 Auto-Completion Keyboard Shortcut

Pressing Ctrl + Space will bring up the auto-completion list view.

This feature is currently limited since the cursor needs to be exactly placed, for example after '<' character, in order for the list to be displayed.

## 2.5 Xml Validation against a Schema

To validate an xml document, first associate a namespace with it via the xmlns attribute, then either right click and select "Validate" or click the Tools->"Validation Xml" menu option or use the shortcut Ctrl+Shift+V.

Validation information will be shown in the output window. Errors will shown in the output window and the task list. Clicking an entry in the task list will move the cursor to the position of the error in the xml.

```
Xml1.xml*
<?xml version="1.0"?>
<project xmlns="http://nant.sf.net/release/0.85-rc1/nant.xsd">
</project>
```

Output

Xml

----- Validating xml -----  
The required attribute 'name' is missing. An error occurred at , (2, 2).  
Validation: failed

```
Xml1.xml*
<?xml version="1.0"?>
<project xmlns="http://nant.sf.net/release/0.85-rc1/nant.xsd">
</project>
```

Task List

!	Line	Description	File
✖	2	The required attribute 'name' is missing. An error occurred at , (2, 2).	Xml1.xml

## 2.6 Folding

The editor will show folds down the left hand side if the document is well formed xml. Elements and comments can be folded if they spread over more than one line.

A folded element will display <element name>.

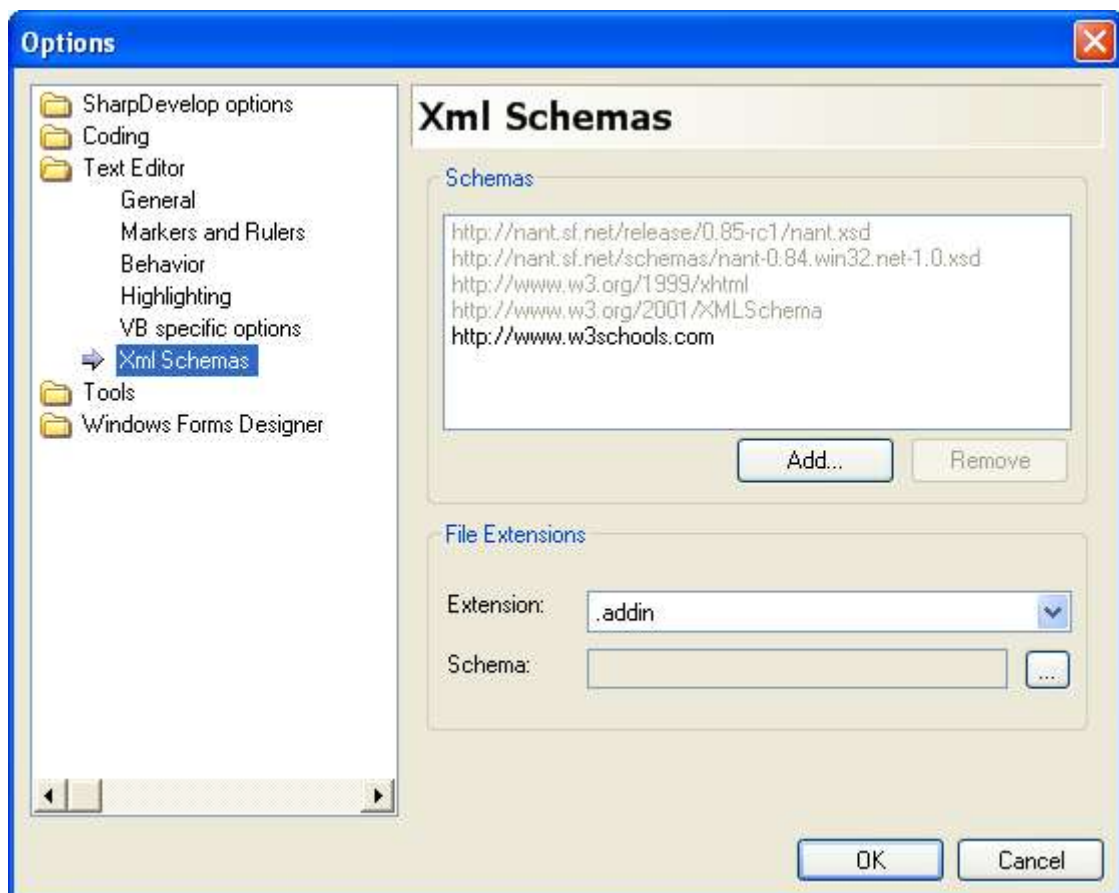
A folded comment will show the first line of comment text inside comment tags.



```
Xml1.xml*
<?xml version="1.0"?>
<project xmlns="http://nant.sf.net/release/0.85-rc1/nant.xsd"
  <!-- Foo build project for Foo Test -->
  <target name="build">
    <mkdir dir = "${addindir}"/>
    <csc>
    <copy>
  </target>
</project>
```

## 2.7 Configuring Schemas

The schemas that SharpDevelop is aware of can be viewed via an options dialog found at Tools->Options->Text Editor->Xml Schemas.



In the "Schemas" group box, schemas that have been added by the user are highlighted in black text, system schemas, those that are installed with SharpDevelop (none currently), are displayed in grey text.

User added schemas are copied to the folder "C:\Documents and Settings\[username]\Application Data\ICSharpCode\SharpDevelop\schemas".

System schemas are stored in the folder "C:\Program Files\sharpdevelop\data\schemas". They cannot be removed via the options dialog since they exist in "C:\Program Files", assumed to be read-only, and so they are displayed in grey text.

Only user added schemas can be added or removed. Clicking the "Add" button opens up the "Open file" dialog allowing the user to browse to an .xsd file and add it.

The "File Extensions" group box shows default associations between an xml file extension and a schema. For example, ".build" files could be associated with a nant schema, then whenever a .build file is opened it automatically gets auto-completion for that nant schema without the need to specify a "xmlns" attribute. Any xmlns attribute specified in the xml will override the default schema associated with the file extension. To associate a schema with an xml file extension click the browse button and select a schema from the list of configured schemas.

## 3 Deployment

### 3.1 Binaries

Assuming you installed SharpDevelop to "C:\Program Files\SharpDevelop", create the folder:

```
"C:\Program Files\SharpDevelop\AddIns\AddIns\DisplayBinding\XmlEditor"
```

Copy the contents (including subfolders) of the zip file's "bin" folder to the folder just created.

Run SharpDevelop.

### 3.2 Source Code

Source code should be copied to "SharpDevelop\src\AddIns\DisplayBindings\XmlEditor" so the folder structure is:

```
SharpDevelop\src\AddIns\DisplayBindings\XmlEditor  
SharpDevelop\src\AddIns\DisplayBindings\XmlEditor\AddIn  
SharpDevelop\src\AddIns\DisplayBindings\XmlEditor\Src  
SharpDevelop\src\AddIns\DisplayBindings\XmlEditor\Doc
```

### 3.3 Building

Open up the XmlEditor.cmbx in SharpDevelop and build the code. The PostBuildTasks.bat file should copy various files to the SharpDevelop\AddIns\AddIns\DisplayBindings\XmlEditor folder.

Alternatively use Nant 0.84 and run the XmlEditor.build file.

## 4 ToDo

### 4.1 Medium Priority

Compulsory attributes added? User selects an element from autocomplete, compulsory attributes added automatically.

Attribute value intellisense.

Support for included schemas via schemaLocation attribute.

Attributes from extension elements parents.

More intelligent intellisense. Handle choices/sequences/min max occurs.

Duplicate attributes.

More tests for missing type checks in XmlSchemaCompletionData.FindElement and FindChildElement.

An extension element that is itself derived from an extension element.

Restriction element's base type that has a 'required' attribute.

No 'type' attribute in xs:element intellisense.

Mixed namespace support. Handle namespaced element inside an element from a different namespace.

More intelligent Ctrl+Space handling (e.g. Ctrl+Space in the middle of an element name, in an attribute name, etc.)

Intellisense for attributes inside an element written like:

```
<foo  
    bar="aa"  
>
```

## 6 Issues

### 6.1 File name extension changes

How to handle the user changing the filename extension to a non-xml extension.

Currently if you rename an xml file, currently on display, to say a .cs file you lose any .cs code completion, since you are not using the standard text editor, until the file is closed and reopened. There is a similar issue if you rename a .cs to an xml file.

Options:

- 1) Disallow this.
- 2) Do nothing.
- 3) Change highlighter only.
- 4) Create a SharpDevelopTextEditor control inside the XmlView.
- 5) Create the correct view for the content.
- 6) Merge the xml editor with the SharpDevelopTextEditor.
- 7) Eclipse style - user chooses which editor to use.
- 8) Derive from SharpDevelopTextEditor and override processing when the file is xml.

Should there only be one editor for text? Architecture change to handle switching between editors automatically when the filename changes?

### 6.2 Xml validation with no schema specified

Xml validation on a file which has a schema associated it by default, but none explicitly specified via an xmlns attribute. When you run validate it is not using any schema so it is always valid.

### 6.3 Code duplication

Code duplication - There is a big overlap with the default text editor and the text editor library classes.

The CodeCompletionWindow class has been duplicated and tweaked:

- Now changes its width based on the size of the text being displayed.
- Only tab and enter now trigger the insertion of any selected item in the auto-completion list.

The SharpDevelopTextAreaControl and the TextEditorDisplayBinding share a lot of similar code to the XmlEditorControl and the XmlView respectively.

## 6.4 Xhtml1-Strict

The .Net XmlSchema class is not able to load the xhtml1-strict.xsd because this schema redefines the "xml" namespace via the string `xmlns:xml="http://www.w3.org/XML/1998/namespace"` at the top of the file. According to the w3 spec this is valid, but the XmlSchema class throws an exception. Removing this definition allows it to load.