Language Injection Settings Dialog: XML Tag

File | Settings | Project Settings | Language Injection - XML Tag

The dialog box opens when you click + in the Language Injection page, and choose **XML Tag** on the context menu, or select an entry and click 🔧.

Item	Description
Display Name	In this text box, type the name that will identify the injection configuration in the list. The text box is available when configuration is created (+), or configured (🔧).
Language	In this area, specify the language of the injection. If necessary, define the context to be automatically made up when the injection is opened in the editor. ▪ **ID** - from this drop-down list, select the ID of the language of the injection. ▪ **Prefix** - in this text box, specify the character string to be automatically added *before* the injection. ▪ **Suffix** - in this text box, specify the character string to be automatically added *after* the injection. The **Prefix** and **Suffix** text boxes are optional.
XML Tag	In this area, define the XML tag which indicates that the text enclosed in this tag should be treated as the selected language. ▪ **Local Name** - in this text box, specify the tag name without a namespace prefix. Use regular expressions to specify multiple tag names (`name1|name2`), case-insensitive names (`(?i)tagname` matches *tagname* as well as *TagName*), etc. Be sure not to enter any space characters as they would be significant for the match. ▪ **Namespace** - in this text box, specify the namespace URI of the XML tag. The field is optional.
Apply to all text fragments recursively	Select this check box to have the injection applied to embedded fragments recursively.

Item	Description
Advanced	In this area, specify additional settings to enable more fine-grained control over the injection process. - **Value Pattern** - in this text box, type a regular expression that determines the part of the XML text's value to inject the language into. By using the first capturing group of the pattern as the target for injection, you can configure the procedure to have the language injected only into values that match a certain pattern or into multiple parts that match the pattern. Examples: `[$#]\{(.*?)\}` matches the pattern used by the JSP/JSF Expression Language. `^javascript:(.*)` matches the `javascript` protocol that can be used in hyperlink-hrefs to execute JavaScript code. - **XPath Condition** - in this text box, specify an XPath expression to address the injection-target more precisely. The context in which the expression is evaluated is the surrounding XML tag. It is possible to use the XPath extension functions that are provided by the Jaxen XPath engine, e.g. `lower-case()`. Also, there are three additional functions that can be used to determine the current file's name, extension, and file type: `file-name()`, `file-ext()` and `file-type()`. Alternatively, a list of available functions can be retrieved through standard code completion. For performance reasons, it is recommended that you keep these expressions as simple as possible. Especially expressions that cause the whole document to be scanned, such as `//foo/bar` might cause performance problems with large files. - **Single File** - select this check box to specify an exact file to apply the injection to.

See Also

Procedures:

- Using Language Injections

Reference:

- Regular Expression Syntax Reference

External Links:

- IntelliLang
- IntelliLang Configuration

Web Resources:

- Developer Community