SharpMedia Templating Engine

© 2008Žiga Osolin – SharpMedia project, document version: DRAFT

Table Of Contents

[About 2](#_Toc200384984)

[Input 3](#_Toc200384985)

[Source File 3](#_Toc200384986)

[Templating File 3](#_Toc200384987)

[Project File 3](#_Toc200384988)

# About

SharpMedia Templating Engine is a Developer tool that adds support for templates. Templates are higher level, allowing things that would be available by C++'s preprocessor and templates (and probably a bit more). A template is used to generate C# (or other language's) source files as a pre-built step using a template file and XML description (template injection) file(s).

# Input

Templating engine uses a seperate project, templating and source file.

## Source File

The source file contains code (in our case C#, but there are no restrictions on usage) with some special commands that insert text based on templates. The templating engine understands the following commands:

* **@TemplateParameter** – inserts the template parameter (as string) to location;
* **//#Command** - issues a specific command. List of all commands is given below.

Enclosing commands are provided using special begin and end tags, such as:

* //#if - //#endif
* //#ifdef - //#endif

We use **@** and **//#** syntax to not break »runtime« file analysis (no or limited errors in input files).

## Templating File

The templating file is a casual XML file. The file consists of template sets and templates.

A template set can be global or local. All global template sets are available for inheritance (even by local template sets) and their names must be globally unique. Local template sets can be considered as arrays, providing scoping to templates. Their name is used only for location and may not need to be globally unique.

A template alone represents a name-value pair. The only value type currently supported is string, but in future we may allow object based value.

**<TemplateSet Name=GlobalName Extends=SomeOtherSet >  
 <Template Name=SomeName Value=SomeValue/>  
 <TemplateSet Name=SomeScopedVar>  
 <Template Name=FirstInArray Value=1/>  
 <Template Name=SecondInArray Value=2/>  
 </TemplateSet>  
</TemplateSet>**

## Project File

Project files references all source files and template files. Besides that, it also contains all source/template pairs that must be generated. It is also an XML based file with the following sections:

* Sources – constains all sources as **<Source Name=Name Path=Value/>** tags
* Templates – constains all templates as **<Template Path=Value/>** tags
* Mappings – constains **<Mapping Source=SourceName Template=TemplateName OutputPath=OutputPath/>** tags