

# Adobe<sup>®</sup> FrameMaker 6.0



Using XMLcss Plug-in



# Contents

<b>Using the XMLcss plug-in for FrameMaker+SGML</b>	Using the XMLcss plug-in to create an XML style sheet	3
	Read/write rules	4
	Generating EDD based CSS information for HTML files	5
	Configuration Parameters	6



# Using the XMLcss plug-in for FrameMaker®+SGML

**T**he FrameMaker®+SGML XMLcss plug-in lets you generate a cascading style sheet (CSS) for structured documents based on Element Definition Document (EDD) formatting. When you export a structured document to XML using FrameMaker+SGML, both an XML document and a CSS document are produced. The information in the CSS document is based on the formatting information in the paragraph catalog and in the font catalog. If the element names in your FrameMaker+SGML document do not correspond with the paragraph tags and the character tag used to format these elements, the formatting may be inadequate.

The XMLcss plug-in examines the actual formatting used in the FrameMaker+SGML document and produces a CSS that ties element names to the actual formatting of these elements. Context-based formatting rules are also handled correctly.

## Using the XMLcss plug-in to create an XML style sheet

The XMLcss plug-in does the following:

- Generates a .css file containing formatting instructions related to the actual formatting of the document
- Lets you generate a .css file for the document or book you are exporting
- Lets you change the contents of an XML file by wrapping the XML comments in HTML tags, extending image and equation elements with an "src" attribute, deciding whether to scale all images and font sizes, and applying the standard HTML namespace to image and table elements

### Configuring FrameMaker+SGML to use the XMLcss plug-in

In order to take advantage of the XMLcss plug-in for SGML applications, you must set the "UseAPIClient" setting to xml-export. To do this, you must modify the sgmlapps.fm file.

If you are already using a non-standard API-client for SGML read/write, you might want to make a copy of the entire application. Call the new copy something like xml-<old name>. Set the new application to use xml-export as the API client via the "UseAPIClient" setting.

#### To configure FrameMaker+SGML to use the XMLcss plug-in:

- 1 Choose File >Developer Tools > Edit SGML Application File.
- 2 Click an insertion point and click Structure View button (  ). In Structure View, click to the right of Version and press Return to create a new application.
- 3 Type the application name, such as XMLcss Client, and press Return.

4 Click the Element Catalog button (I1). In the Element Catalog, double-click UseAPIClient, and then do one of the following:

- (Windows) Type XML-css
- (Macintosh) Type XMLcss
- (UNIX) Type ext-xml

5 Use the read-write rules feature, XMLWriteRules, to point to a document with the xml-write rules. A good starting point for the xml-write rules file is a copy of the sgml read/write rules file.

6 When finished, save and close sgmlapps.fm.

**To use the XMLcss plug-in:**

1 Open a document and verify that it uses the SGML application associated with the xml-export APIClient in the sgmlapps.fm file. To do this, choose File > Set SGML Application. Choose the name of the XML client you configured from the pull-down menu, and then click Set.

2 Choose File > Save As, add the .xml file extension to the filename, choose XML from the Save as type pop-up menu, and click Save. Choose the name of the XML client you configured from the Use SGML Application pop-up menu and click Continue.

3 Start your browser and open the XML file you saved.

*Note: For viewing XML files, you must use any XML-compliant Web browser. Microsoft Internet Explorer 4 and Netscape Navigator 4.5 are not XML compliant.*

4 Set your browser to display using Unicode so that quotation marks and other special characters will be displayed properly.

5 Depending on your browser you may want to map all FrameMaker+SGML image elements to img elements. This is done by adding a number of lines such as the following to the read/write rules file:

```
element "Img" is fm graphic element "xx";
```

Note that you can have a separate xml-write rules file for your SGML application. In addition, you might want to export formats of images to either GIF or JPG.

## Read/write rules

The XML client is strictly browser-independent. You can therefore configure through the read/write rules by inserting a comment in the read/write document. Note that the configuration parameters must be a comment to avoid interference with the standard read/write file parser. Note also that a standard FrameMaker return must be used to divide the lines.

To produce an XML and a CSS file that can be viewed in an XML-capable browser, you don't have to enter any special rules. All preferences default to the setting that will work with standard XML browsers.

If you need to customize the export functionality, you have to enter information in the read/write rules file. The following is an example of a setting that scales all image sizes by a factor of 2.5 and all font-sizes by a factor of 1.6:

```
/*  
#CS-EXTENDED-READ-WRITE-START  
FontFactor 1.6  
ImageFactor 2.5  
#CS-EXTENDED-READ-WRITE-END  
*/
```

The following is an example of exporting XML files without generating a CSS file:

```
/*  
#CS-EXTENDED-READ-WRITE-START  
ExtendedCSSMode None  
#CS-EXTENDED-READ-WRITE-END  
*/
```

## Generating EDD based CSS information for HTML files

You can use the XMLcss to generate an EDD based CSS for an HTML file. To do this you should create a read/write rules file.

**1** Re-map all SGML tags to HTML by inserting rules, such as:

```
element "UL" is fm element "List";  
element "LI" is fm element "Item";
```

You may need to unwrap some elements, using a rule such as:

```
fm element "TableHead" unwrap;
```

**2** Configure XMLcss to add extra HTML information. The following is an example of a set of preferences that configures XMLcss to generate HTML/CSS that can be displayed using Netscape 4.5 or later.

```
/*  
#CS-EXTENDED-READ-WRITE-START  
WrapXMLinHTML Yes  
FontFactor 1.1  
ImageFactor 1.8  
Images_and_Tables_Use_HTML_NS No  
#CS-EXTENDED-READ-WRITE-END  
*/
```

See “Configuration Parameters” on page 6 for a more complete description of configuration parameters.

## Configuration Parameters

The following table provides a description of configuration parameters:

Parameter	Values	Meaning	Default
FontFactor	Decimal number	Used to change font sizes in the generated document. The right way to construct a web version of an SGML version is to construct a web-specific EDD. If you don't have one at hand, you can enlarge font sizes by using this parameter.	1.1
ImageFactor	Decimal number	Used to control the size of produced images	1.8
WrapXMLinHTML	Yes   No	Everything that is not Yes is a No. If Yes, then the XML is embedded in HTML tags. This can be used to generate an HTML file with a CSS document based on the EDD formatting. (See "Generating EDD based CSS information for HTML files" on page 5.)	No
Images_and_Tables_Use_HTML_NS	Yes   No	Everything that is not Yes is a No. If Yes, then image and table elements will be assigned the HTML name space. This is used to notify browsers (such as IE5) that the elements should be formatted specially. Example:  <HTML:TABLE> . . . </HTML:TABLE>	Yes
ExtendedCSSMode	PerBook   None   Standard	Determines if a css file should be generated and whether or not the css file should be extended.  If the value is None, no extended css file is generated and the default CSS file is deleted.  If the value is Standard, the built-in css generator will be used to generate the css file.  If the value is PerBook, the XMLcss client will generate an extended css file for the book. If a document is being exported, one extended css file is generated for the document.	PerBook