

1

What's new in DocFrame version 2

What's in this chapter

- ◆ Formatting template
 - ◆ EDD
 - ◆ Structured application
 - ◆ XSL stylesheets
 - ◆ Sample files
-

This chapter explains new and updated features in DocFrame version 2. Users of versions 1 and 1.1 should read this chapter for important information about changes to DocFrame components.

If you did not use a previous version of DocFrame, you can skip this chapter.

Formatting template

The formatting template contains changes to paragraph tags and conditional tag names.

Paragraph tags

There are two new paragraph tags in version 2: `v_version` and `r_revision`. The `v_version` tag has autonumbering that contains the word “Version”; the `r_revision` tag has autonumbering that contains the word “Revised.” These tags are associated with the `Version` and `RevisionDate` elements, respectively.

In version 1, the words “Version” and “Revised” were generated by prefix rules for the two elements. By using autonumbering in associated paragraph tags instead, the EDD is more language independent—translators update the paragraph tag autonumbering instead of modifying the EDD. The actual version number and revision date, supplied by book attributes, are now EDD suffix rules.

Also, text formatted by the footnote paragraph tags (`f_footnote` and `tf_table footnote`) is now smaller.

Condition tags

To avoid problems with XSL condition processing, spaces in condition tag names been removed. Tags named `online only` and `print only` in previous versions are now called `onlineonly` and `printonly`.

Indicators for the condition tags have been changed as well.

EDD

The EDD for version 2 includes changes to formatting rules and attributes.

Formatting rules

The formatting rules for the `Version` and `RevisionDate` elements have changed. Those elements are now associated with the `v_version` and `r_revision` paragraph tags.

FileName attribute for chapter- and section-level elements

The `FileName` attribute has been added to all top-level elements except `UserGuide` and to all of the section-level elements (including the Introduction sections). This attribute lets you assign specific file names in the HTML output.

The `FileName` attribute is optional. When XML files are converted to HTML, the XSL transformations base file names on that attribute value. If you specify no value, the file name is generated automatically.

For details on file naming in the XSL transformations, see “File naming” on page 5.

Id attribute

In `DocFrame` version 1, the `Title` element had an `Id` attribute; cross-references to `Title` elements are based on that attribute. In `DocFrame` 2, the `Id` attribute has been moved to the parent elements of `Title` (for example, `Chapter` and `Section`) to facilitate cross-references across books saved as XML—a new feature in `FrameMaker` 7.1.



If you import the EDD for DocFrame 2 into files based on DocFrame version 1, all cross-references to Title elements will be broken—the Id attribute is not valid for the Title element in the EDD for version 2. Evaluate whether you need the features provided by DocFrame 2 before you import the structure into existing files and are forced to rebuild the cross-references.

Structured application

The structured application for DocFrame version 2 is called **DocFrame2**. The new name ensures that version 1 applications—particularly those that you modified to match your own structure based on DocFrame version 1—aren't overwritten during installation of version 2 files.

The template file, DTD, and read/write rules contain some changes.

Template

The template file associated with the structured application (docframetemplate.fm) has been updated to reflect the changes to the formatting template and to the EDD.

DTD

The DTD has been changed to reflect the changes to the structure of the EDD. The DTD also includes a new custom entity definition for nonbreaking hyphens.

Read/write rules

A new read/write rule was added to handle nonbreaking hyphens.

Application definitions file

There is a separate application definitions file (structapps.fm) for versions 7.0 and 7.1 of FrameMaker. The version 7.1 file includes configuration information for cross-book linking in XML output.

XSL stylesheets

The XSL stylesheets for DocFrame version 2 are significantly different from those in previous versions.

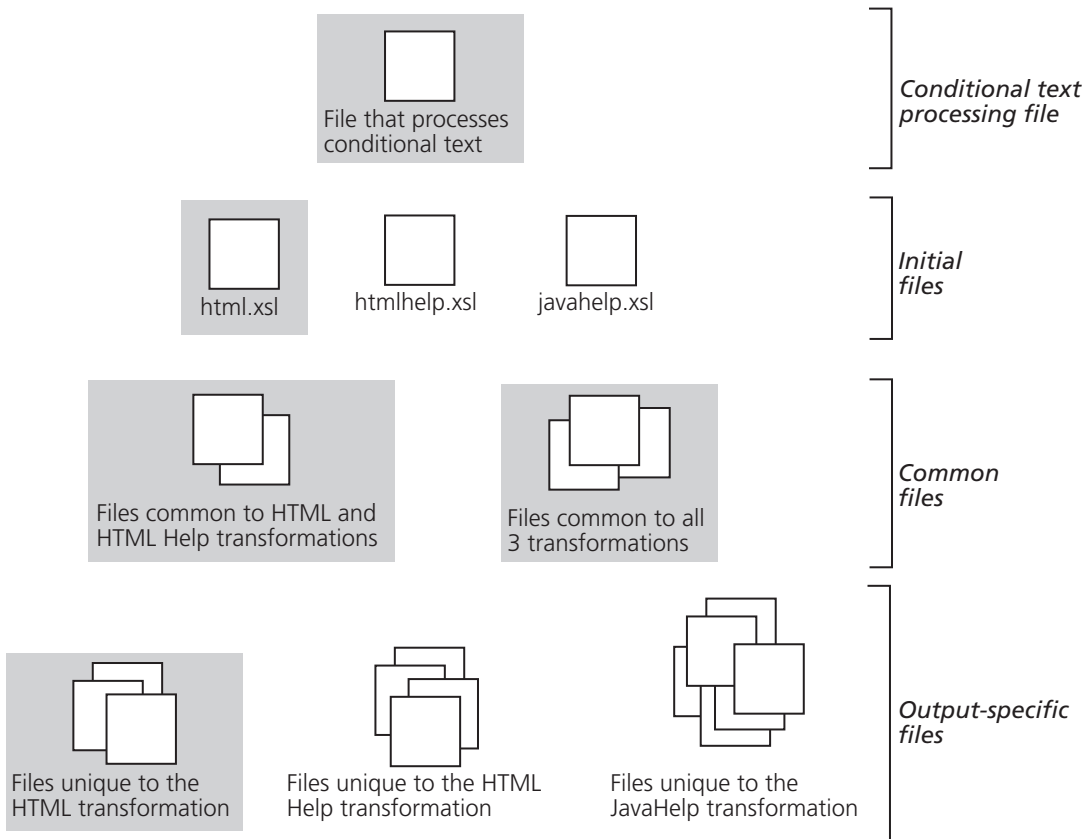
JavaHelp

The DocFrame environment now includes XSL files to transform XML into JavaHelp 2.

Modular design

Version 2 stylesheets are modular. Each type of output—HTML, HTML Help, and JavaHelp—has a main stylesheet (html.xsl for HTML, for example). The main stylesheet uses several other stylesheets (for example, the xrefs.xsl stylesheet that handles cross-reference processing). A few of the stylesheets are used by all three output formats.

The shading in Figure 2 shows the files used during the transformation to HTML.



See “Overview of the DocFrame version 2 transformation files” on page 74 for descriptions of the XSL files in version 2.

Processing of conditional text

FrameMaker 7.1 preserves conditional text in files saved as XML output. A new XSL file, `process_conditions.xsl`, processes the conditions in the XML files.

File naming

In DocFrame version 1, output file names reflected the position of the chapter or section. For instance, a file name of `UG1-3.html` referred to the first chapter, third section.

In DocFrame version 2, output file names are generated automatically using XSL's `generate-id()` function. The resulting file names are arbitrary; most are like `d0e3431.html`. Using `generate-id()` guarantees that you will get a single, unique ID for a given element during any one transformation. The value generated may change from one transformation pass to the next.

DocFrame version 2 also allows you to specify the file names you want. To accomplish this, you use the new `FileName` attribute, which is available for chapter- and section-level elements. If a file name is specified for this attribute, that value is used instead of the default generated file name.

When you name files, keep in mind the following rules:

- ◆ If you use the `FileName` attribute, you must ensure that each file name you provide is *unique*. If you do not, file-name collisions may occur, which will cause problems in your transformed output.
- ◆ By default, the XSL transformation uses the `TitlePage` element to create the splash (entry) page, named “index.” Do not name another file “index” unless you have renamed the `TitlePage` element. If you name another file “index” without renaming the entry page, the entry page is overwritten.
- ◆ The `FileName` attribute should not include the file name extension. The extension is added by the `pages.xsl` file and is defined as a variable in the output's main XSL file (`html.xsl`, `htmlhelp.xsl`, or `javahelp.xsl`).

Variable changes

Because of the file naming changes (discussed in “File naming”), the root-name and index_pos variables are no longer needed and have been removed.

A number of global variables have been added to simplify customization of the navigation bar. The next_label, prev_label, ix_label, and toc_label variables define the default text displayed in the navigation bar as Next, Previous, Index, and Contents, respectively. You can customize these variable definitions if you need to localize the stylesheets. You can also replace the text with image references.

The new file_ext variable controls the file extension for generated files. The default is .html.

Splash pages

A splash page is generated for HTML, HTML Help, and JavaHelp. This processing is found in the TitlePage transformation template. The table of contents processing no longer includes the version number and revision date; those have been moved to the splash page.

Sample files

The sample files in the SampleFiles directory reflect the changes to the EDD and the formatting template.