

FrameMaker + SGML Quick Guide

Daniel K. Schneider, TECFA, University of Geneva

<http://tecfa.unige.ch/guides/xml/frame-sgml/>

Version 0.1 - nov 2, 2000,

Should be usable somewhat., but it certainly is rough draft quality

Summary

This is a very short guide to help you editing structured documents with FrameMaker + SGML (version 6). It is meant for a “tech savvy” audience. In addition you should skim through the FrameMaker+SGML book you get with the software and which is well done but too verbose. Note, that there are also companion documents for somewhat XML savvy users who want to work with their own structure definitions, see: <http://tecfa.unige.ch/guides/xml/frame-sgml/>

Introduction

FrameMaker + SGML (“F+S”) is a full featured SGML editor. It allows you to import SGML DTDs and text and to export the same. XML imports and exports also work (but with some restrictions). F+S is quite easy to learn at the user level, but very difficult if you want to define your own SGML application. So if you are alone with your F+S Box, plan a week to get really going unless you just want use Frame’s version of Docbook. If you simply want to edit with Docbook or lternatively use something your local SGML guru set up for you, you are in business after an hour or two and this document will help you just with that. For people who’d like to import XML DTDs into F+S, see my other quick guides.

F+S has its own element description language with which “SGML engineers” define EDDs (Element Definition Documents). EDDs define the structure of a document, most of its layout and more advanced things (like cross-reference elements, elements that act as index markers, etc.). When you author a document, you simply start with a blank template and then import definitions from an EDD files if the template does not have them already. The EDD also contains a reference to an SGML application, i.e. a short entry in some file somewhere on your system hat defines things like the character sets you use, (read/write rules form SGML/XML to F+S and the other way round), SGML configuration (important for XML fans) and such.

In other words, you can not just “use” FrameMaker+SGML besides DocBook: You either need some technical support in your site or you have to know how to register SGML applications for EDD files you can find elsewhere. Along with FrameMaker you will have an old version of the large DocBook schema and some lighter ones used in some demos. Both are difficult to find on all architectures.

What you should know already

You should have used (standard) FrameMaker before in order to profit from this little guide. However, if you have need used FM before but are very good in “grasping” the logic of interfaces, you can give this little guide a try anyhow.

Suggested Steps

Step 1: Locate Templates

Strangely enough, F+S template files for structured editing are very difficult to find. A template file is an empty FrameMaker file from which you start composing your document. In FM (unlike MS Word) there is no fundamental distinction between Style files and document files. A template file is just an empty file, but typically has already imported definitions from an EDD file. In order to open a FM file as template do Menu:File->New->Document. Then ***I wish you good luck*** (because you won't find much in your standard templates, unless some F+S knowledgeable person installed your site or machine). Here is what I found:

- ***Structured Templates***: See directory Templates/Structures: E.g. 1 report template, 2 outliner templates and slides. The report template uses modern layout style.
- ***Docbook***, the template file is called `template` and sits in the `sgml/docbook/app` directory. E.g. in our Unix installation it sits in: `/unige/frame_6.0/fmunit/ukenglish/sgml/docbook/app/template`
- In directory `/Samples/FMSGML` is a non editable EDD file (*why the hell?*), but you can open file `SGMLDevGuide.fm` file and export or reimport the EDD. You can do this trick with 2-3 other files found in the `/Tour` directory.

Exporting a EDD from a document (if you can't find the original) is done like this: Menu:File->Developer Tools->Export Element Catalog as EDD [*How the hell can one know what EDD has been imported in a template without exporting the EDD I don't know, one of many problems I have with F+S*]

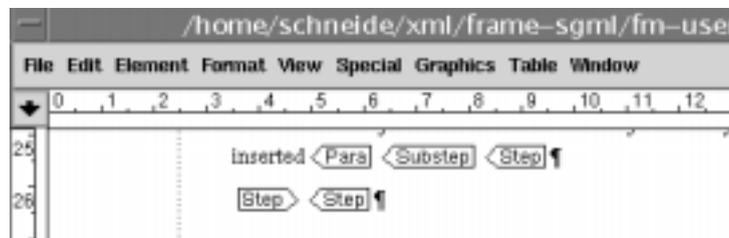
IMHO it is a bit embarrassing for Adobe that they only offer so few templates, unless they really think that their product should not be used without having a local SGML engineer who knows how to set up the environments. There are dozens of interesting DTDs in the World, but Adobe seems to ignore this situation.

Step 2: Learn the location of the main tools

(1) The menu bar

The End-User Interface of F+S is quite well done. Before you start exploring, remember that you edit structured text. So ignore all the typical word processor

features. ***Do not do any formatting what so ever***, just enter information! If you must change formatting, go and learn how to edit the EDD file. If you don't have an EDD file, export one from your current File. So ignore the Format Menu unless you know what you do. Here are the most important things you need to know:

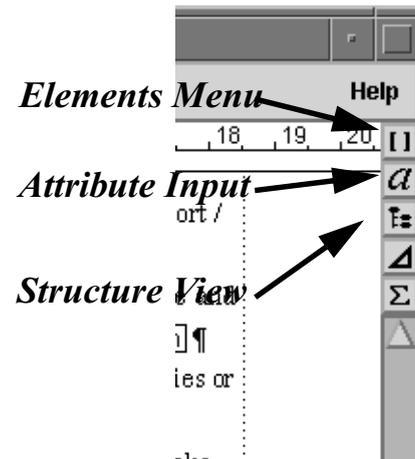


- With the File Menu you can open templates, files, but also import /export EDDs and access to Developer tools
- With the Element Menu, you can (a) validate your text, (b) split, merge and unwrap elements although mouse: right-click offers the same in context.
- With the View Menu you can define if you want to see element boundaries or not (and how)
- The Special Menu allows to insert graphics (anchored frames) and make Cross-references.

(2) The other authoring windows

There are three other Windows for SGML editing. All of them are important.

- The most important Window which always should be open is the Elements Menu. It will allow to insert an Element Tag when you author. It should be always open
- The Attribute Window will allow you to enter attribute values
- The Structure View shows you the structure of your document and your current position



(3) Be a power user

Make sure you get the full Menus

Menu: View->Menus->Complete

Note: configuring SGML related Menus and using some of these tools is explained below

Step 3: Open a Template or an example file

As I hinted before, this is a first hurdle, since you may just have bought F+S and have no help in your organization. Well in order to just play with a file and learn a little bit the interface, you can read the F+S Book (that came along with the software) and play around with subway doors. So there are three options:

- Open one of the four structured templates or the Docbook template (see above for their location)
- Install a SGML application in FrameMaker. See the companion document quick-fm-xml-guide.* for some help in this area. E.g. you can fiddle around with the Stepbystep EDD I wrote (see: <http://tecfa.unige.ch/guides/xml/frame-sgml/>)
- Open file /Tour/Doors.fm and play with it.

Step 4: Element boundaries View or Structure Window?

Know where you are

In structured editing it is important to know where you are, i.e. you must be aware within

which element in the XML/SGML hierarchy you are editing or inserting your mouse pointer. There are 2 principal ways of doing it explained below. Before you start playing with these options, make sure that you have either a Template or an example file opened. as explained above!

(1)Element boundaries View

In the View Menu of the main Menu bar you can define whether and how to display element boundaries. If you have a small screen (17" or smaller and especially if you work under Windoze) I suggest that you make visible elements:

Menu: View->Element Boundaries (as Tags)

This is mostly how I work (despite having a 21" monitor), unless I do proof-reading and minor editing, in which case I turn this view off.

(2)Use the structure Window

Open up the "Structure View" (click on the icon in the little bar to the right as explained above). This View will always show you where you are. Now if you want to insert an element in some place of hierarchy, just insert the cursor in front or after an element. In order to avoid scrolling you can also open/hide sub-trees by clicking on the +/- signs to the left of Element Names. The -/+ to right hides/displays Attributes. Note: under MS Win this window wants to stay in front and obscures the document window, no idea if this annoying behavior can be changed.

(3)Show Element Context

You can get some additional information about the your editing context by using: File->Developer Menu->Show Element Context. I don't use this.

Step 5:Further editing options

Here are 2 others settings I suggest

(1)Attribute insertion

When you insert a new element in you text, F+S will by default ask you to insert values for all attributes. If you think that you do not need this turn it off or ask it prompt you only for required values:

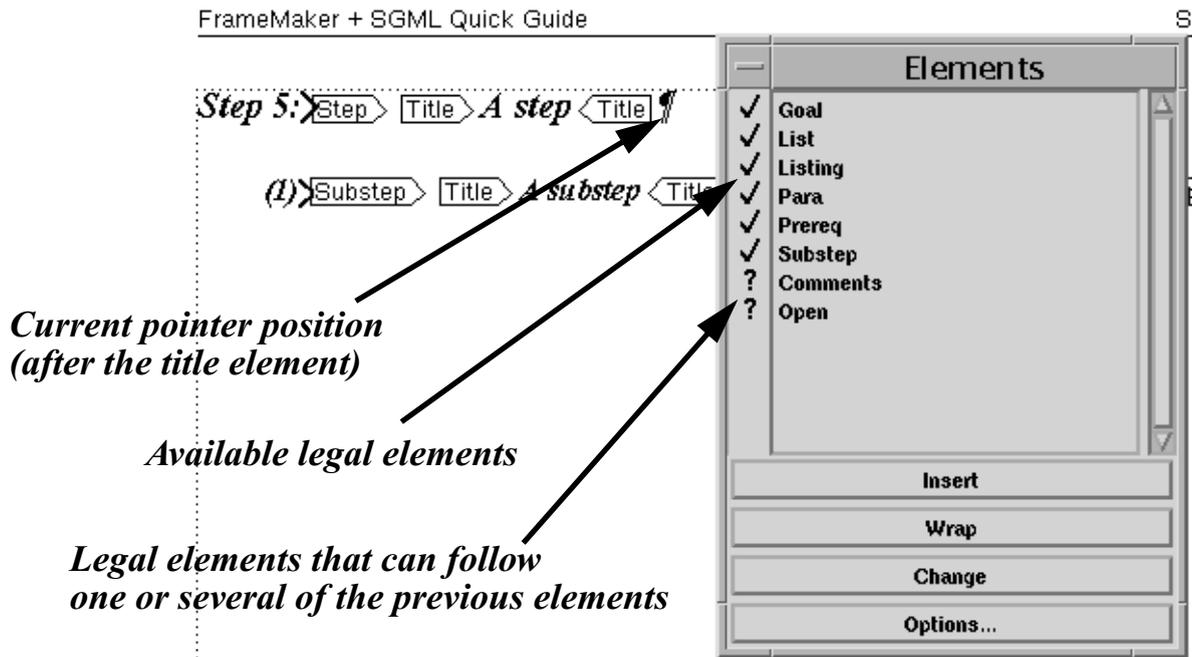
Menu: Element->New Element Options

Chose (IMHO) "Prompt for required elements", I strongly suggest this if you play around with my Stepbystep DTD. It asking you for ID Tags all the time is annoying

(2)Available elements in the Element Menu

You define what Elements should be available for authoring either in "Options" in the Element Menu or in Element->Set Available Elements ..." I suggest the following:

Valid Elements for Working Start to Finish



This will basically show the elements you are allowed to insert at a certain point. It will not offer you to insert elements that you already have inserted. See the figure.

Let's have a look at an example for those who know some XML or SGML. Here is the DTD rule for the Step Element (shown in both XML and EDD format):

XML:

```
<!ELEMENT Step (Title, Goal?, Prereq?, ( Para | List | Listing | Substep)*, Comments?, Open?) >
```

EDD: General rule: Title, Goal?, Prereq?, ((Para | List | Listing)* | Substep)*, Comments?, Open?

Since we already inserted a Title Element, It will not show in the options. Other existing elements not available within the Step will not be shown either if you configure the elements window as suggested.

Step 6: Write something

(1) Insertion

Insertion should be easy, provided that:

- you grasp the idea of structure editing
- that either turn on element boundaries viewing or know how to click into the structure view

(2) Changing an element

E.g. you wrote a paragraph as some element and now think it should be something else. If "something else" is a legal element in the same context do the following:

- select the whole element (in the text window glide the mouse over a boundary, in the structure window click on it)

- select an element in the Elements window and click on “Change” below in the same Window

Else, use some creative inserting, cutting, pasting, moving

(3) Merging and splitting

See either Menu: Element or context (mouse:right) menu. To split an element into 2 sisters, simply insert your mouse where you want to split and “Split”. To merge 2 brothers, glide your mouse over both and “Merge”.

(4) Unwrapping

It often happens by mistake that you have some text inside some element. If you want to “unwrap”, just select the element (as above) and click on unwrap in Element Menu or the Context (right-click) menu.

(5) Wrapping

Wrapping some text into some element is something that do frequently with so called “text range elements”. E.g. in my Stepbystep DTD I defined elements “Code”, “Strong”, “Notsure”, etc. to be used within paragraphs and list items.

(6) Moving

Select the element. Then you can either cut/paste it in the Text editing Window or drag it around in the Structure View. I prefer the further. However be aware that you need to use the “Cut” and not “Delete” button (on my Sun this is not the same).

Step 7:Validate

Structure text ought to be valid

For various reasons not explained here, your text should be valid, i.e. respect the rules for your type of structured document. If you are unhappy with these rules, write new ones or complain to your SGML guru.

(1) The validation feature

Menu: Element->Validate

Will you show invalid parts (step by step like a spelling checker). With the above operations you can fix things. Mostly you just need to insert missing elements or missing parent elements.

(2) Look at the structure window

This is an alternative method. Elements in wrong positions etc. are shown in red.

Step 8: Learn some shortcuts

Shortcuts: I suggest that you look at the Quick Reference Card. Some shortcuts come really handy, in particular if you do not wish to turn on the “View Element Boundaries View”. If you know well your structure, you can guess where you are just by looking the elements menu (which should be always open). Here are 2 the most useful ones (learn some more):

- “After the next element”: Meta-down (unix) or Alt-Ctrl-down (Win)
- “Start of next element’s contents: Meta-right (unix) or Alt-Ctrl-right (Win)

Note: Under Unix a lot of people use the small subset of Emacs with standard FrameMaker. You will not find these for structure editing, e.g. No M-C-f :(

Step 9:[Optional]: Installing SMGL applications

Real users have SGML applications

An SGML application will allow you to manage EDD files, but also to insure that a Template can find the right read/write rules, predefined entities, uses the right character set etc. You can set the default SGML application in the Menu: File->Set SGML Application. End users don’t need to do this. They either open a Template File or Import Elements from a EDD into an Empty New File. The EDD usually has an SGML defined. So if you play with my examples below, don’t start changing names before you understand what you do.

Ok, the rest now just deals with a practical example, adapt to other situations :)

(1) Get the files

Grab all the files from <http://tecfa.unige.ch/guides/xml/frame-sgml/Stepbystep/>. If you are on a slow connection you can take less, i.e. you just need Stepbystep-edd.fm, Stepbystep-sgml.dcl, Stepbystep.dtd, Stepbystep.rules. In addition I suggest that you take the source of this very document you are reading:

(2) Open the SGML application file

Menu: File->Developer Tools->Edit SGML Application File

This will automatically open your central “SGML applications” configuration file. It is called sgmlapps.fm (probably). It might be a good idea to first make a COPY of this file since it is read on startup (see below where you can find it).

Under Windows you can find this file in the SGML installation directory: **smglapps.fm** [probably]. Warning: If you are 2 people using F+S on the same machine, coordinate a bit here

Under Unix, the procedure is a bit more complicated when you do it the first time, because each user can have his own SGML application file (as it should be). Here are the steps:

Menu: File->Developer Tools->Edit SGML Application File

It will complain that you can't write (at least with our site thing) because you are dealing with a system-wide installation. So you have to save the file to your own "fminit" directory in something like the following way (roughly). It must go to:

```
~/fminit/<your_language_version>/sgml/sgmlapps.fm
E.g.
  /home/schneide/fminit/ukenglish/sgml/sgmlapps.fm
  /home/schneide/fminit/usenglish/sgml/sgmlapps.fm
```

If the xxx/sgml directory does not exist you have to make it of course. Before you do so figure out what language version of F+S you use. A good test to know what language version you have got is to create a fminit/xxx/Templates directory, put a file there, then click on "New" in FrameMaker main menu and see if the file is there.

```
E.g.
  home/schneide/fminit/usenglish/Templates/manuel.fm
```

(3)Install the SGML application

This is an important step. If you don't do it right, you will suffer later! You can kill or leave the existing definition for DocBook. Since sgmlapps.fm and only this file will hold all your future SGML applications I suggest that you leave it. Ok, now let's edit this file:

Insert your mouse after "Application Definition Version". This will show a SOLE SGML "Application" element in the Element menu. Select SGMLApplication and click on insert Then add a few definitions. IMHO the following ones are good enough for starters. Replace ALL the values (text after "(:)") in the following example with your preferences:

```
Application name:Stepbystep
Read/write rules:XXX/Stepbystep.rules
XML character encoding: ISO Latin1
SGML character encoding: ISO Latin1
DOCTYPE:Stepbystep
DTD:XXX/Stepbystep.dtd
SGML declaration:XXX/Stepbystep-sgmldecl
```

Note: On a PC use PC style for this filename like c:\Home\mysgml\XXX\

(4)[Optional] Install the template file

Copy file Stepbystep-template.fm to your Framemaker installation (somewhere in a directory called Templates. Then when you start a new file with File->New->Document you will hopefully see your new template. If you do not want to do this, you can always click yourself to the place where you put it.

(5)Use an EDD or a template file

Now, to write your own Stepbystep document you can either:

- Open a new document with the Template file
- Open the EDD file first (no need to edit it!), just open it. Then open a blank custom document and import EDD elements: In this blank document you do: Menu: File->Import Element Definitions, save it

Enjoy :)