

Destination: Round Trip XML

Single Sourcing with Frame + XML

STC @ 50

Mary Ann Howell
maryann@hikaripub.com
Copyright 2003



What I'm not going to do

- ☞ Focus on XML: what it is, why we use it.
(That's a complete presentation on its own)
- ☞ Dwell on single-sourcing
(That's also another presentation)
- ☞ Philosophize: well, maybe not much — we've got a lot to cover in an hour and a half

What I am going to do

- ☞ Quick review of single source preparation
- ☞ Hands-on demo with a plain vanilla FM document
- ☞ Apply structure
- ☞ Convert it to XML
- ☞ Chunk the XML
- ☞ Store the XML chunks
- ☞ Reuse the chunks in a FM document
- ☞ Reuse the chunks on an active page on the web
- ☞ Comment on the process

What I hope you'll do

- ☞ Give me your feedback
- ☞ Share your experiences
- ☞ Recommend information sources, tools
- ☞ Make jokes

First of all...

Your content needs to be worth reusing.

Review: 7 Steps to Minimalist Writing

1. Analyze the users and their goals
2. Design the document to support the user
3. Write once and use many times
4. Write clearly and concisely
5. Edit
6. Test
7. Edit and test until it is ripped from your hands

Today's use case...

- ☞ We have existing documentation: the SlantAlpha user guide, an unstructured, plain vanilla FrameMaker document
- ☞ We want to convert the document to structured FrameMaker
- ☞ Convert the structured FrameMaker document into XML chunks for XML-based single sourcing
- ☞ Use the chunks for dynamic publishing on the web
- ☞ Use the chunks for insets in an automatically updated FM document

My background coming into the project

- ☞ Good background in FrameMaker
 - ☞ Four months XML training
 - ☞ Worked through the XML Cookbook tutorial included with FrameMaker 7
- Programs / Adobe / Program Files / Adobe7.0 / XMLCookbook

References I used

- ☞ FrameMaker 7 Structure Application Developer's Guide Online Manual (Adobe)
- ☞ XML Cookbook User Guide (Adobe)
- ☞ *The Complete Reference FrameMaker7* by Sarah S. O'Keefe and Sheila A. Loring
- ☞ *Developing SGML DTDs From Text to Model to Markup* by Eve Maler and Jeanne El Andaloussi
- ☞ Applications included in FM 7 program directory

XML overview: What it is

- ☞ People can read it
- ☞ It's generic, no one owns it
- ☞ Like HTML, it's a subset of SGML
- ☞ Unlike HTML, it's tagged by information type, rather than how it is formatted
- ☞ Has metadata, information about the information
- ☞ Uses hierarchal, nested organization

XML overview: Use it to:

- ☞ Share content across a company
- ☞ Save translation costs
- ☞ Document on any platform using any XML authoring tool
- ☞ Produce dynamic content:
 - Online or just-in-time printing
 - Only the information needed (screened by user profile: beginner, intermediate, advanced; security level)
- ☞ Publish in the appropriate format:
PDF, HTML, WML (Wireless markup language) PDB (portable database for Palm hand-helds)

XML overview: Structure

- ☞ Text is wrapped in tags <element>
- ☞ Elements can have attributes (metadata)
- ☞ Valid XML has a DTD or schema says "I'm XML", calls out what tags are legal in what order, and links to style sheets
- ☞ The resulting XML document says, "I'm XML", calls out its DTD or schema, and is the text-carrying or application document

XML DTD example

```
<?XML version="1.0" encoding="UTF-8"?>
<!--Top level containers-->

<!ELEMENT Book      (Front, Chapter+, Glossary?, Index) >
<!ATTLIST Book      Product
(SlantAlpha|DocDBase|Webspinner|Hikari_Chunker) #REQUIRED >
<!ELEMENT Front     ANY >
<!ELEMENT Chapter   (ChapterTitle, TopicsList, Overview,
                    Section+) >
<!ATTLIST Chapter   AuthorName CDATA #REQUIRED
                    Version
(1stDraft|2ndDraft|3rdDraft|Final) "1stDraft" >
```

XML document example

```
<?XML version="1.0" encoding="UTF-8"?>
<!DOCTYPE Chapter SYSTEM "slantalpha.dtd" >
<?XML-stYLESHEET href="App\slantalpha.css" type="text/css"?>
<Chapter AuthorName="Mary Ann Howell" Version="2ndDraft">
<Section>
    <Inset>
        <Head ID="sah0001" Product="SlantAlpha"
InfoType="Procedure" Subject="SlantAlpha installation intro" InfoLevel="Beginner">
            <?FM MARKER [Cross-Ref] 27398: Heading1:
Package contents?>Installing &SlantAlpha;</Head>
        <Para>To install SlantAlpha, follow the instructions
provided with
your PDA. </Para>
        <Para>For online information, see <HotLink>
            <Hypertext type="Hypertext"
text="message URL
http://www.palmos.com/products/install"/>http://www.palmos.com/products/install/</Hot
Link>
        </Para>
```

XML document example: read by an XML-compliant browser

Installing SlantAlpha

To install SlantAlpha, follow the instructions provided with your FDA.
For online information, see
<http://www.palmira.com/products/install/>

Steps to converting an unstructured FM document to XML

- ☛ Develop an EDD
- ☛ Apply structure to the FM document
- ☛ Set up the XML translation documents
FrameMaker needs to convert to valid XML
- ☛ Save the structured FM document as XML

Why use FM's structured authoring?

- ☞ Frees authors from all formatting
- ☞ Enforces your style guide—no one-step bullets or procedures; no headings without paragraphs
- ☞ Consistency
- ☞ Can't leave out key sections
- ☞ Ease of use — can move stuff around and navigate
- ☞ Free addition to FM7, used to be big \$\$

Setting up structured authoring in FrameMaker

- ☞ Plan the architecture
- ☞ Use FrameMaker in the Structured mode:
File / Preferences / General / Structured FrameMaker
- ☞ Write an Element Definition Document (EDD) while testing it with your document
- ☞ Make your test document the structure template
- ☞ Apply the template to unstructured documents

Examine the architecture

Book

Front matter

Table of Contents

Chapter

Overview

Sections

Headings

Introduction information

Procedure

Examples

Related topics links

Extra information- in depth

(Paragraphs, bullets, steps tables, graphics, cross refs (look at para and character tag catalogs))

Glossary

Term

definition

Index

Develop an EDD

☛ I started with the ChapterEDD.fm from:

Program Files / FrameMaker7.0 / XML Cookbook /
Completed folder

☛ I also referred to:

Program Files / FrameMaker7.0 / Samples / fmsgml /
SGMLDevGuide.edd

EDD example

EDD Version is 7.0
Structured Application: StartAlpha

Top level containers

Element (Container): Book
Valid as the highest-level element.
General rule: Front, Chapter, Glossary, Index

Attribute list

| | | |
|-----------------------------------------------------------|--------|----------|
| 1. Name: Product | Choice | Required |
| Choices: StartAlpha, DocDBase, Webspinner, Hikari_Crawler | | |

Element (Container): Front
Valid as the highest-level element.
General rule: <ANY>

Element (Container): Chapter
Valid as the highest-level element.
General rule: ChapterTitle, TopicList, Overview, Strategy, Section+

Attribute list

| | | |
|----------------------------------------------|--------|----------|
| 1. Name: AuthorName | String | Required |
| 2. Name: Version | Choice | Optional |
| Choices: 1stDraft, 2ndDraft, 3rdDraft, Final | | |
| Default: 1stDraft | | |

Automatic insertions
Automatically insert child: ChapterTitle

Applying structure

- ☞ Import the structure: With your EDD open, from your plain vanilla FM document select: File / Import / Element Definitions
- ☞ From your plain vanilla FM document select: File / Structure Tools / Generate Conversion Table
- ☞ Edit the table to match your EDD
- ☞ With the conversion table also open, from your plain vanilla document select: File / Utilities / Structure Current Document

Conversion
table
example

| Wrap this object or objects | In this element | With this qualifier |
|-----------------------------|-----------------|---------------------|
| P:chapter title | ChapterTitle | |
| P:topic head | TopicHead | |
| P:topic para | TopicPara | |
| P:Body | Para | |
| P:body | Body | |
| P:body indent | Para | |
| P:bullet | Bullet | unordered |
| P:bullet sub | Bullet | unordered |
| P:caption | Callout | |
| P:cell heading | TableHeading | |
| P:heading1 | Head | |
| P:heading1 top | Head | |
| P:heading2 | Head | |

Apply element tags

- ☞ As per directions in your FrameMaker user guide
- ☞ Or work through FM 7 Classroom in a Book examples
- ☞ Or follow instructions in the XML Cookbook user guide
- ☞ Or follow instructions in The Complete Reference FrameMaker 7

Tagged FM example

Section **Unset** **Head** Installing SlantAlpha **Head** ¶

Para To install SlantAlpha, follow the instructions provided with your PDA. **Para** ¶

Para For online information, see **HotLink** **HyperText** <http://www.palmos.com/products/install/> **HotLink** **Para** ¶

1) **Procedure** **Step** Unzip the slantalpha3.5.zip file on your computer. **Step** ¶

2) **Step** Connect the HotSync® cable to your PDA. **Step** ¶

3) **Step** Follow the user guide instructions for installing add-on applications supplied with your PDA and install ¶

- } **Bullet** SlantAlpha.pro **Bullet** ¶

Set up the translation documents for FM roundtrip XML

1. The source application file, which points to the documents below: structapps.fm
2. A blank template: slantalphaTemplate.fm
3. An FM-to-XML translation document: slantalphaReadWrite.fm
4. Style sheets for XML formatting XSLT and or CSS: slantalpha.css, slantalpha.xsl
5. An XML DTD: slantalpha.dtd

Update the source application file: to point to the FM-to-XML support files

stuctapps.fm

Application Definition Version 7.0

| | |
|--------------------------------|-----------------------------------|
| Application name: | SlantAlpha |
| DTD: | App/slantalpha.dtd |
| Read/write rules: | App/slantalpha/Read/WriteRules.fm |
| Template: | App/slantalphaTemplate.fm |
| CSS2 Preferences: | |
| Generate CSS2: | Disable |
| Add FM CSS Attribute To XML: | Disable |
| Retain Stylesheet Information: | Disable |
| XML Stylesheet | |
| Type: | css |
| URI: | App/slantalpha.css |
| Namespace: | Enable |
| DOCTYPE: | Book Chapter Section |

Produce a blank template

Delete all the text from your structured FM example
and save the blank file as:

slantalphaTemplate.fm

(or whatever you have listed in your structapps.fm)

Make a Read/Write rules document (Translates between the EDD and the DTD)

I started with the ChapterReadWriteRules.fm
document located:

Program Files / FrameMaker7.0 / XML Cookbook /
Completed folder

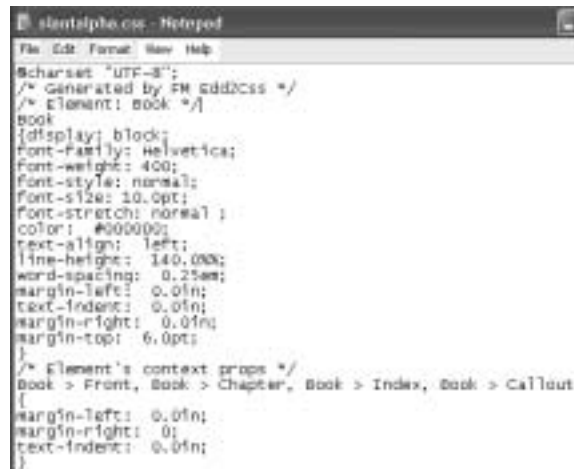
Read/Write rules example

```
fm version is "7.0";  
/*  
 * Include all ISO entity mapping rules.  
*/  
#include "isoall.rw"  
element "NoteTable"  
{  
    is fm table element "NoteTable";  
    attribute "Column_Num" is fm property  
columns;  
}  
element "RelatedTopicsTable"  
{  
    is fm table element "RelatedTopicsTable";  
    attribute "Column_Num" is fm property  
columns;
```

Produce CSS stylesheets

- ☛ Read and follow the instructions in the XMLcss.pdf located:
Programs / Adobe / FrameMaker7.0 / OnlineManuals
- ☛ From your structured FM example, select:
File / Structured Tools / Generate CSS2

CSS2 stylesheet example



```
slant1pbe.css - Notepad
File Edit Format View Help
@charset "UTF-8";
/* Generated by FM gdd2css */
/* Element: Book */
Book
{
display: block;
font-family: Helvetica;
font-weight: 400;
font-style: normal;
font-size: 10.0pt;
font-stretch: normal;
color: #000000;
text-align: left;
line-height: 140.000%;
word-spacing: 0.25em;
margin-left: 0.0in;
text-indent: 0.0in;
margin-right: 0.0in;
margin-top: 6.0pt;
}
/* Element's context props */
Book > Front, Book > Chapter, Book > Index, Book > Callout
{
margin-left: 0.0in;
margin-right: 0;
text-indent: 0.0in;
}
```


Transform the FM EDD
into an XML DTD

Convert the EDD to a DTD:

File / Structure Tools / Save As DTD

Save the structured FM document
as XML

File / Save As / (select XML from Save as type)

Open your XML file

- ☛ Use any XML editor or XML compliant browser
- ☛ Then start tweaking all your translation files to correct weirdnesses

Web publishing

If your use case includes posting to the web, you've got more to do...

- ☛ Your users will probably not be using the XML-compliant browser of your choice.
- ☛ You'll need to transform your XML to HTML with XSALT and an XSLT Processor, such as SAXON.

XSLT example

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<!-- <xsl:template match="Chapter" --> -->
<!-- <xsl:template match="Inset" --> -->
<html>
<head></head>
<body style="padding-left:5%; display:block; font-family:Arial; font-style:normal;">
<xsl:apply-templates/>
</body>
</html>
</xsl:template>
<xsl:template match="ChapterTitle">
<p style="color:blue; font-size:x-large;">
<xsl:apply-templates/>
</p>
</xsl:template>
```

Wow, what a snap, eh?

Comments on process—What's up my sleeves

- ☛ I married a top-notch software engineer to help with my XSLT and to get the web database together
- ☛ We are still setting up the XML-HTML transformation for our website
- ☛ My use case can use FM7 to generate DTD, XML, and link to XML chunks
- ☛ But — the XML chunks linked to by the FM7 document cannot be updated from the FM inset dialog box
- ☛ FrameMaker final role?: publishing for print, PDF, online help
- ☛ Recommend considering at the least, structured authoring in FM
- ☛ Wait for it — Any grandiose scheme is likely to be outdated before it is finally hatched
- ☛ Watch for third-party tools that will help you get where *you* want to go

Hikari Publishing
Fine distillation of information
www.hikaripub.com