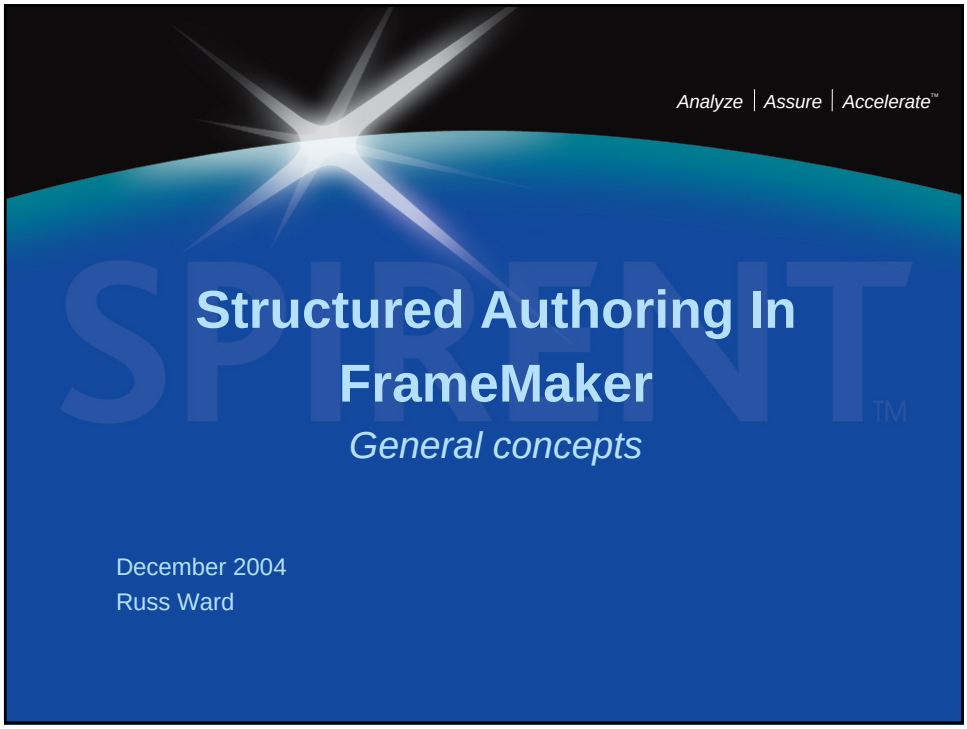



## Main Points

- Why structured authoring?
  - Technology is capable of performing work and improving efficiency in any technical writing process
  - To employ technology, the computer must first be able to recognize your content and its architecture the way that you do.
  - Structured authoring adds the necessary information (or *metadata*) to your content that makes technological intervention possible.

SPIRENT Communications Analyze | Assure | Accelerate™

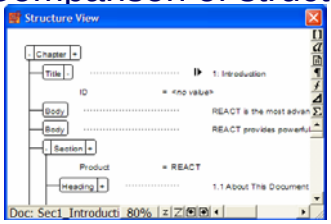
## Main Points (cont'd)

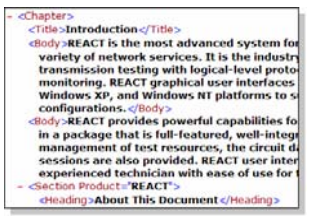
- Why FrameMaker as the structured authoring tool?
  - Most popular long-document authoring software available, and from a solid software company
  - Often is already the preferred authoring software, which minimizes migration requirements
  - Offers perhaps the friendliest structured environment on the market, with simplified structure development that does not require programming.
  - Powerful native tools to leverage structural markup for authoring conveniences and formatting consistency
  - Completely extensible via its API to perform customized tasks, including advanced single-sourcing activities that leverage structural markup.



## About elements

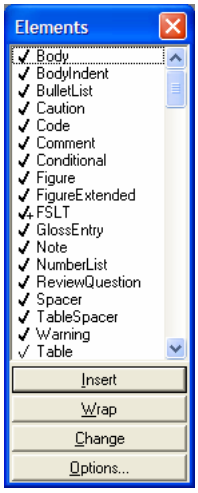
- A structured FrameMaker document follows the XML structure model (but is not XML).
- The fundamental components are *elements*, which wrap content and other elements to form the structural hierarchy.
- Comparison of structured Frame to "real" XML



## Authoring in structured Frame

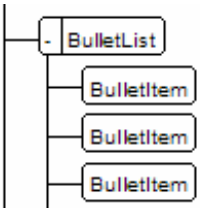
- In many ways similar to unstructured authoring, except much simpler once the concept of explicit architecture is mastered.
- In structured Frame, the *element catalog* is used to insert and format content, not the paragraph or character catalogs. The element definitions apply all formatting automatically.
- Element catalog-based authoring is much easier because your choices are automatically limited based on *context*. No more fishing through a giant list of formats for what you want.




SPIRENT Communications Analyze | Assure | Accelerate™

## Authoring in structure Frame (cont'd)

- A constant awareness of document architecture is important. In structured authoring, you are explicitly specifying that architecture to the computer, and in turn it is using it to perform work.
- All pieces of content are "owned" by something, such as how a bullet list "owns" its bullet items. In structured authoring, this knowledge of ownership is the key to success.




SPIRENT Communications Analyze | Assure | Accelerate™

## General conveniences of structured authoring

- Enhanced navigation with the structure view
- Quick selection of logical components
- Fast movement of content by dragging and copy/paste
- Exposure of otherwise hidden components, such as table components
- Elements can be easily changed, wrapped, and unwrapped



## "Smart formatting" with the EDD - Examples

- Application of different "note" formats, and even dynamic altering of those formats, based on where the note appears.
- Automatic selection of heading formats based on the level of section nesting. No more incorrect heading hierarchy.
- Automatic selection of chapter title format based on document type, as defined by the *highest-level element*.

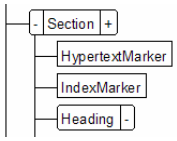
## Guided authoring - Examples

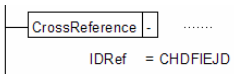
- Predetermined chapter and section hierarchies
- Preset list structures (no more accidental bullets in a numbered list)
- Enforced anchor paragraphs for graphics and tables
- Rigid whole-document structures as applicable, such as for a cover page



## Other authoring benefits

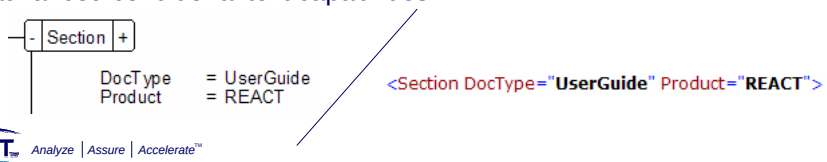
- Enhanced control over FrameMaker gizmos such as markers and cross-references, because these objects are now wrapped in structural elements and exposed in the structure view
- Easier table manipulation through the structure view
- Easier and more intuitive cross-referencing, because links are made through structural metadata, not elusive markers
- Simple management of overlapping character formats, because each character format is represented by an element



## Attributes

- Attributes are a separate feature of structural markup in an XML content model, in addition to elements.
- Attributes are owned by elements, and any element can have any attributes in accordance with the desired structure definition.
- Attributes, as the name implies, generally add descriptive data about an element.
- Attributes are “pure” metadata, in that they simply provide a means of applying additional information to your content, without necessarily altering the element structure or hierarchy.
- Because attributes “tell about” elements, they are the perfect place to store information about conditional output, in pursuit of advanced conditional text capabilities.




## Advanced single-sourcing - Overview

- The addition of structure already makes native single-sourcing features easier, due to the delimiting qualities of the *markup*.
- Once markup is available, though, advanced single-sourcing capabilities become immediately possible, because the metadata provides the roadmap necessary for a computer to manage the content.
- Native FrameMaker is notoriously weak on markup-based single-sourcing features. However, inexpensive and robust third-party plugins are available to fill the gap.

SPIRENT Communications Analyze | Assure | Accelerate™

## Markup & metadata – Buzzword roundup

- In the context of documentation structure, the terms markup and metadata are effectively synonymous.
- Markup and/or metadata are the features of the structure at add the additional information about your content, including element names, attribute values, and component relationships.
- Markup and/or metadata can sometimes seem like amorphous buzzword terms, but they are extremely important and have real meaning once you have applied them in a real-life situation.
- The markup and/or metadata of a structured Frame document are conceptually identical to an XML document, which is why XML-related technologies can be applied to Frame content via API plugins.



## Two types of single-sourcing

- "Conditional text" approach – Represented natively in FrameMaker by the conditional text feature
- Granular content reuse – Represented natively in FrameMaker with text insets (and to some degree, variables)

## "Conditional text" approach to SS

- Characterized by putting the content for all outputs in a single contextual text flow, and tagging conditional content with some nature of metadata. Prior to delivering or publishing, non-applicable conditional output is simply removed by some method.
- The conditional text approach is a virtually irreplaceable component in any single-sourced workflow. If done right, it quickly becomes a mainstay.
- FrameMaker conditional text pioneered this approach in an unstructured, WYSIWYG authoring environment.
- Native FrameMaker conditional text is good for unstructured work, but the advanced potential introduced with structural metadata makes it obsolete.



## Granular reuse approach to SS

- Characterized by the reuse of content chunks at will, anywhere needed
- The order of content in the deliverable is independent of the order of content in the "repository." In other words, you have a set of pieces that you reassemble as desired.
- Differs from conditional text, in that conditional text "takes away" the parts you don't want, whereas a granular reuse approach adds the parts you do want, wherever you want
- FrameMaker text insets follow the granular reuse concept, and helped pave the way for advancements with the methodology.
- Cutting edge, XML-based single-sourcing technologies generally follow this model, such as XSLT applications.
- The FrameSLT plugin allows this level of reuse and transformation within Frame, by applying XPath/XSLT technologies in the GUI.

## Sourcerer

- Sourcerer is a complete replacement for native conditional text, using attribute metadata instead of condition tags to identify conditions
- Sourcerer is owned by Advantica, Inc. in Carlisle, PA. Rumor has it that it will be released shortly at a low cost, perhaps even for free. For more information, visit www.advantica.biz/sourcerer.
- Once put into use, Sourcerer becomes a fundamental component of the documentation process.



## Sourcerer features

- Produces output by removing the undesired conditional output, much like native conditional text
- Can color “conditional text” like native conditional text
- Validates the hierarchy of assigned attribute conditions using a set of standard rules
- Provides a set of general shortcuts for working in the structured environment.

## Sourcerer advantages

- Can manage virtually unlimited and overlapping conditions, both for coloring and producing output.
- Uses the concept of preset schemes to produce output, color, and validate, ensuring consistent content management.
- Allows you to “lock” conditions in the structural metadata for easy viewing and management. In addition, the use of structural metadata for conditions prepares the content for migration out of FrameMaker, should it ever become necessary.
- Can create a duplicate book for the output
- Allows entire chapters to be conditionalized.



## Other notes on Sourcerer

- Provides for multiple, overlapping conditions, which allows the generation of many unique outputs from one source. For example, it would become possible to output different user guides and online help from a single book.
- Unlimited in-text commenting capability, for authors or reviewers
- Ability to have temporary or tentative content in books that you don’t want in the published output
- Interacts well with WebWorks, producing output books directly in the WebWorks project area. This eliminates the need to be concerned with conditions within WebWorks.

## Introduction to FrameSLT

- FrameSLT is an XSLT-emulator that operates on structured FrameMaker documents.
- FrameSLT is based on W3C XML-related technologies.
- FrameSLT is owned by West Street Consulting www.weststreetconsulting.com. (Owned by Russ Ward)



## Advantages of FrameSLT

- Allows unlimited, granular content reuse without the need to leave FrameMaker.
- Provides transformation capabilities, meaning that content repurposed in the deliverable does not need to have the same structure and/or appearance as it did in the “repository.”
- Adds XPath capabilities to structured Frame, an important query language for navigating an XML-model structure tree.

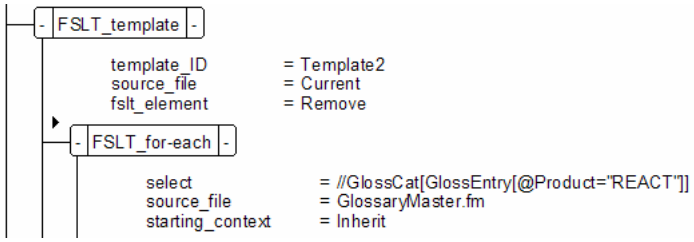
## How FrameSLT works

- FrameSLT recognizes certain “transformation” elements, which you place in your structured documents as the EDD allows. During processing, it processes these elements according to the instructions contained as attribute metadata.
- Normally, a transformation element provides instructions on:
  - What document or book to search for some kind of content
  - What specific content should be retrieved from that document or book, and then how it should be brought into the source document (nature of the transformation)

SPIRENT Communications Analyze | Assure | Accelerate™

## How FrameSLT works (cont’d)

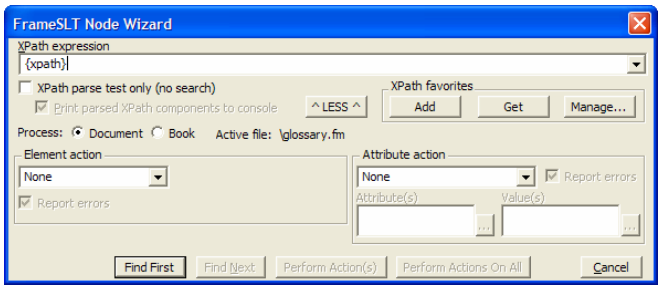
- The transformation process loosely follows the methodology of an XSLT processor, customized for use within FrameMaker.




SPIRENT Communications Analyze | Assure | Accelerate™

## Node Wizard

- FrameSLT also includes the Node Wizard, an XPath-based query and structure management tool. The Node Wizard is a must-have, especially during a conversion from unstructured to structured.






## Reasons to use FrameSLT

- FrameSLT will allow you to use FrameMaker documents and books like content databases.
- FrameSLT is the key to merging external XML content with existing FrameMaker content. This process makes it possible to automatically generate documents and books by piecing together existing XML content.
- Its usage promote education on W3C standards and technologies, which is important for a writer's professional growth.