Single-source Publishing with DocBook 4

Bob Stayton

Sagehill Enterprises

What is Single-source Publishing?

- Multiple outputs from single source file.
- Reusable content.
- Modular writing.
- Assembled documents.
- “Write once, publish many”.

Multiple outputs

- Print, PDF, HTML, slides, Help files.
- Different product versions.
- Different user levels.

Separate content from formatting

- XML file has no formatting.
- Separate stylesheets supply formatting.
- Match formats to element names

Why Single-source?

- Reduce redundant writing.
- Writers can specialize.
- Update in a single place.
- Users get more accurate information.
- Fewer words to translate.

What is DocBook?

- XML for technical documentation.
- Related stylesheets and tools.
- Started in 1991 as SGML, now XML.
- OASIS standard since 1998.

Like HTML, except:

- A lot more tags.
- Tags identify document parts.
- All tags must be closed (or empty).
- No style information.

Why use DocBook?

- Designed for technical documentation.
- Content kept separate from format.
- Quality check through validation.
- Open to computer processing.

More reasons

- Choose your publishing tools.
- Cross platform.
- Automate your processing.

Standard publishing features

- Front matter
- Graphics
- Tables
- Glossaries
- Bibliographies
- Indexes

Technical publishing

- Nested section levels.
- Numbered figures, tables, examples.
- Tasks and procedures.
- Code synopses.
- Code examples.
- Running headers and footers.

Free DocBook stylesheets

- Stylesheets for multiple outputs.
- Many advanced features.
- Customizable.
- Open source project on SourceForge.
- Active support community.

DocBook output formats

- HTML
- XHTML
- XSL-FO
- PDF and PostScript
- HTML Help
- JavaHelp
- man pages
- TeX
- RTF

What’s the downside?

- XML learning curve.
- Investment in setup.
- Technical staff needed.

DocBook is not …

- Microsoft Word
- FrameMaker
- Quark Xpress
- InDesign

Best for …

- Multiple output formats.
- Multiple releases over time.
- Large documentation sets.
- Batch processing environment.
- Shared authoring.

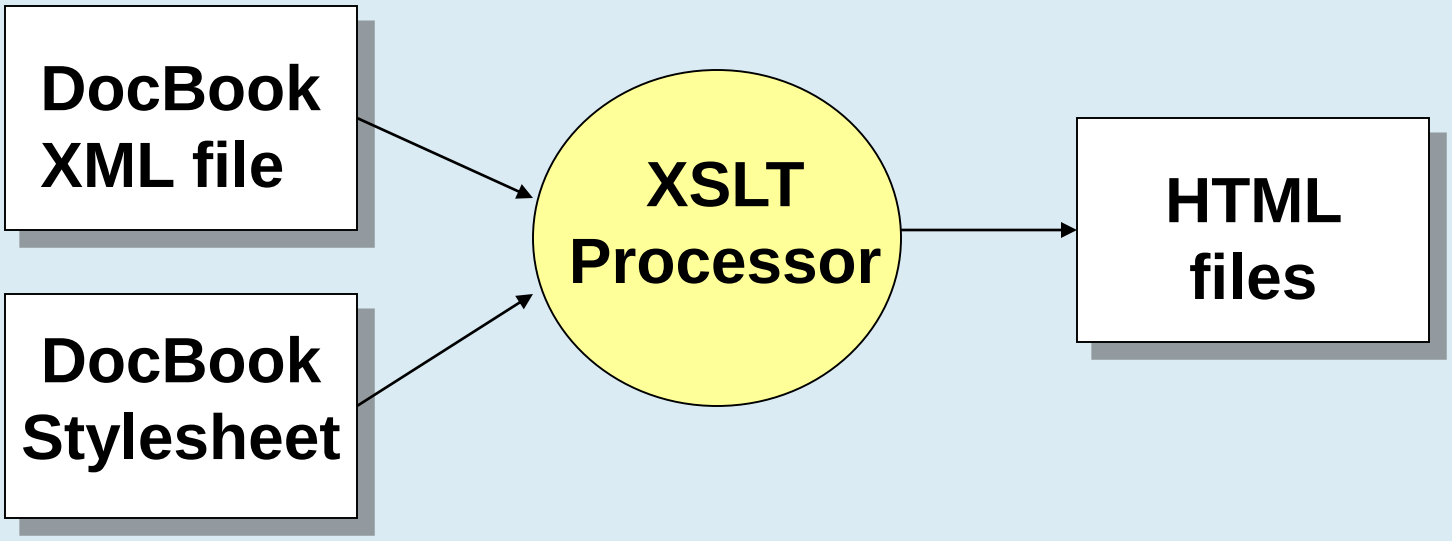
Who uses DocBook?

- Sun
- IBM
- Microsoft
- Hewlett Packard
- Symantec
- Red Hat, SuSE, Mandrakesoft
- Linux Documentation Project

What do you need?

- DocBook Document Type Definition (DTD)
- Writing tools.
- XSL stylesheets.
- Processing tools.

Processing DocBook



Why a DTD?

- Defines element names and usage rules.
- Used to validate documents.
- Improves reliability.
- Special characters with names:
  - `&trade;` instead of `&#x2122;` for ™

DocBook DTD

- OASIS maintains DocBook DTDs.
- Current version is 4.5.
- 400 elements.
- Customizable: easy to subset or extend.

Writing tools

- Notepad or other text editor.
- XMLMind's XMLEditor (free).
- Oxygen XML.
- Just Systems' XMetal.
- Syntext's Serna (free).
- Arbortext Editor.

Writing DocBook

- XML declaration
- DOCTYPE declaration
- Root element

DocBook document

```
<?xml version="1.0"?>
<!DOCTYPE  book
   PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
   "http://www.oasis-open.org/xml/4.5/docbookx.dtd">
<book>
<title>Using a mouse</title>
<para>A mouse is used for blah blah.</para>
<chapter>
<title>Mouse buttons</title>
<para>A mouse has one, two, or three buttons.</para>
</chapter>
</book>
```

XMetal demo

- Formatted editing view.
- Element list based on context.
- Select attributes.
- Show or hide tags.
- Validate.

Three classes of elements

- Hierarchy elements
- Block elements
- Inline elements

Hierarchy elements

- Hierarchy elements establish divisions and hierarchy of content
- Hierarchy elements are containers for other elements.
- May contain other hierarchy elements and block elements.

Book hierarchy elements

```
<set>
    <book>
        <part> (optional)
            <preface>
            <chapter>
            <reference>
            <appendix>
            <glossary>
            <index>
```

Section hierarchy

```
<chapter>
    <sect1>
        <sect2>
            <sect3>
                <sect4>
  OR                    <sect5>
<chapter>
    <section>
        <section>
            <section>
                <section>
                    <section>
```

Article – alternative to book

- Less than a book.
- Like chapter, but not numbered.
- Good for:
  - White paper.
  - Short HOWTO.

Block elements

- Paragraphs, lists, notes, examples.
- Block elements delineated by line breaks before and after in output.
- Lines wrap within (with exceptions).
- May contain text, inline elements, and other block elements (e.g., nested lists)

Block examples

Paragraph	`<para>Text</para>`

Bullet list	`<itemizedlist>`
	`  <listitem>`
	`  <para>First</para>`
	`  </listitem>`
	`</itemizedlist>`

Numbered list	`<orderedlist>`
	`  <listitem>`
	`  <para>Item 1</para>`
	`  </listitem>`
	`</orderedlist>`

More block elements

- **<procedure>**, containing **<step>+**
- **<variablelist>**, like HTML DL list
- **<note>, <caution>, <important>, <tip>, <warning>**
- **<synopsis>, <cmdsynopsis>, <funcsynopsis>**, for syntax.
- **<qandaset>** for FAQs
- **<msgset>** for error messages

Text display elements

- Preserve whitespace.
- Optional line numbering attribute.
- **`<programlisting>`**
  - Monospace font.
- **`<screen>`**
  - Monospace font.
- **`<literallayout>`**
  - not monspace font by default

Program listing example

```
<programlisting>#!/usr/bin/perl
use LWP::UserAgent;
use XML::RSS;
my ($file, $url) = @_;
my $ua = LWP::UserAgent-&gt;new;
&amp;FetchSS($ua);
sub FetchSS {
    return($_-&gt;GetSS)
}
</programlisting>
```

CDATA to avoid escaping

```
<programlisting><![CDATA[#!/usr/bin/perl
use LWP::UserAgent;
use XML::RSS;
my ($file, $url) = @_;
my $ua = LWP::UserAgent->new;
&FetchSS($ua);
sub FetchSS {
    return($_->GetSS)
}
]]></programlisting>
```

Formal displays

- Elements
  - figure
  - example
  - table
  - equation
- Numbered and titled.
- Each can contain appropriate content.

Graphics

- **<mediaobject>**
  - Contains one or more **<imageobject>**
    - Each contains one **<imagedata>**
- Separate object for each output format.
- Select by role attribute.
- Replaces simpler **<graphic>** element

mediaobject example

```
<mediaobject>
  <imageobject role="html">
   <imagedata fileref="mouse.png"/>
  </imageobject>
  <imageobject role="fo">
   <imagedata fileref="mouse.pdf"/>
  </imageobject>
</mediaobject>
```

Inline elements

- Commands, filenames, user input.
- Inline elements appear within a line of text.
- No implied line break before or after.
- May contain text, other inline elements.
- Dozens of inline elements.

Cross reference with xref

```
<chapter id=Mouse>
<title>Using a Mouse</title>
...
See <xref linkend="Mouse"/>.
```

- Output:

```
See <a href=#Mouse>Chapter 3, Using
  a Mouse</a>.
```

Cross reference with link

```
<chapter id="Mouse">
<title>Using a Mouse</title>
...
Use your
<link linkend="Mouse">mouse</link>
for ...
```

- Output:

```
Use your <a href=#Mouse>mouse</a>
  for ...
```

External cross references

- **`<olink>`** uses two attributes:
  - **`targetdoc`** points to another document
  - **`targetptr`** points to an **`id`** in it.
- Requires setting up target database.
- Creates dependencies between documents.

Glossaries

- A **`<glossary>`** contains <glossentry>s
- A **`<glossterm>`** can link to a **`<glossentry>`**.
- Can do automatic linking based on word match.
- Can generate a glossary from an external master collection.

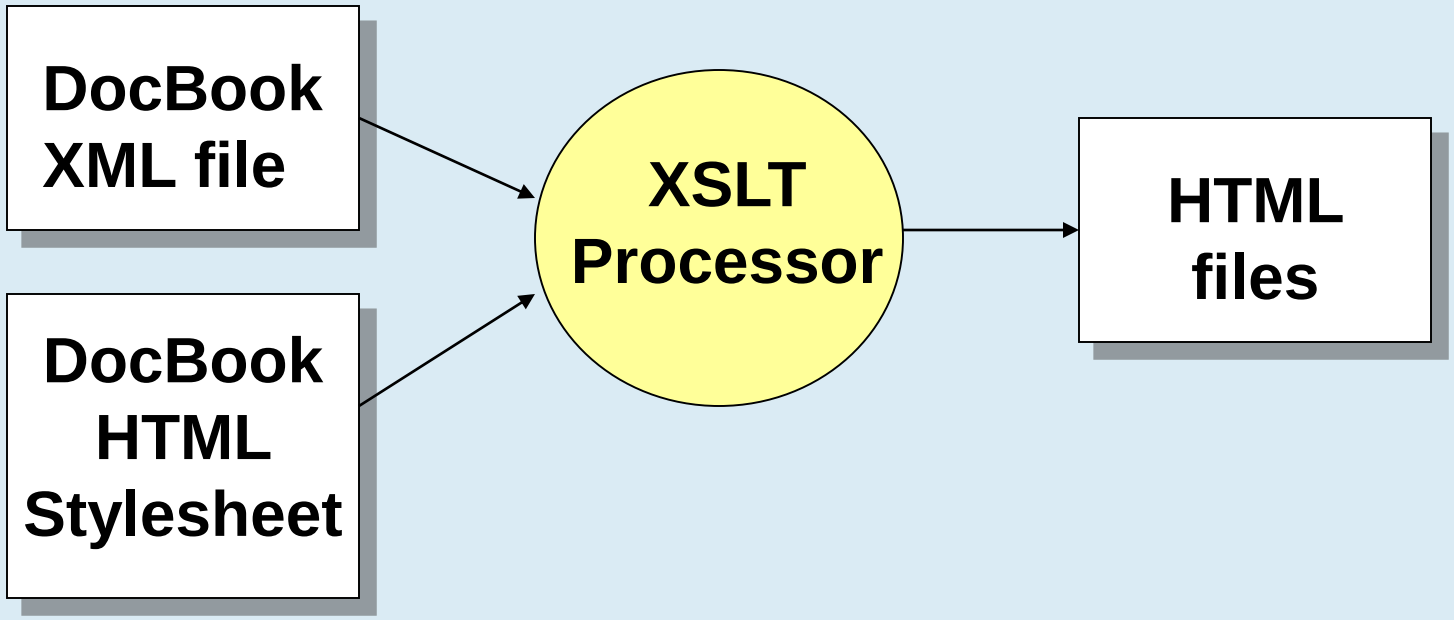
Bibliographies

- A **<bibliography>** contains **<biblioentry>**s.
- A **<xref>** can link to a **<biblioentry>**.
- Numbered [2] or abbrev [Brody2002] link style.
- Can generate a bibliography from an external master collection.

Indexes

- Insert **`<indexterm>`**s in document.
- Add empty **`<index/>`** element at end.
- Stylesheet automatically generates the index.
- HTML shows section titles.
- FO shows page numbers.

Getting to HTML



DocBook XSL stylesheets

- Written in XML using XSL namespace.
- Stylesheet variant for each output form.
- Designed for customization.
- Adapt to corporate style.

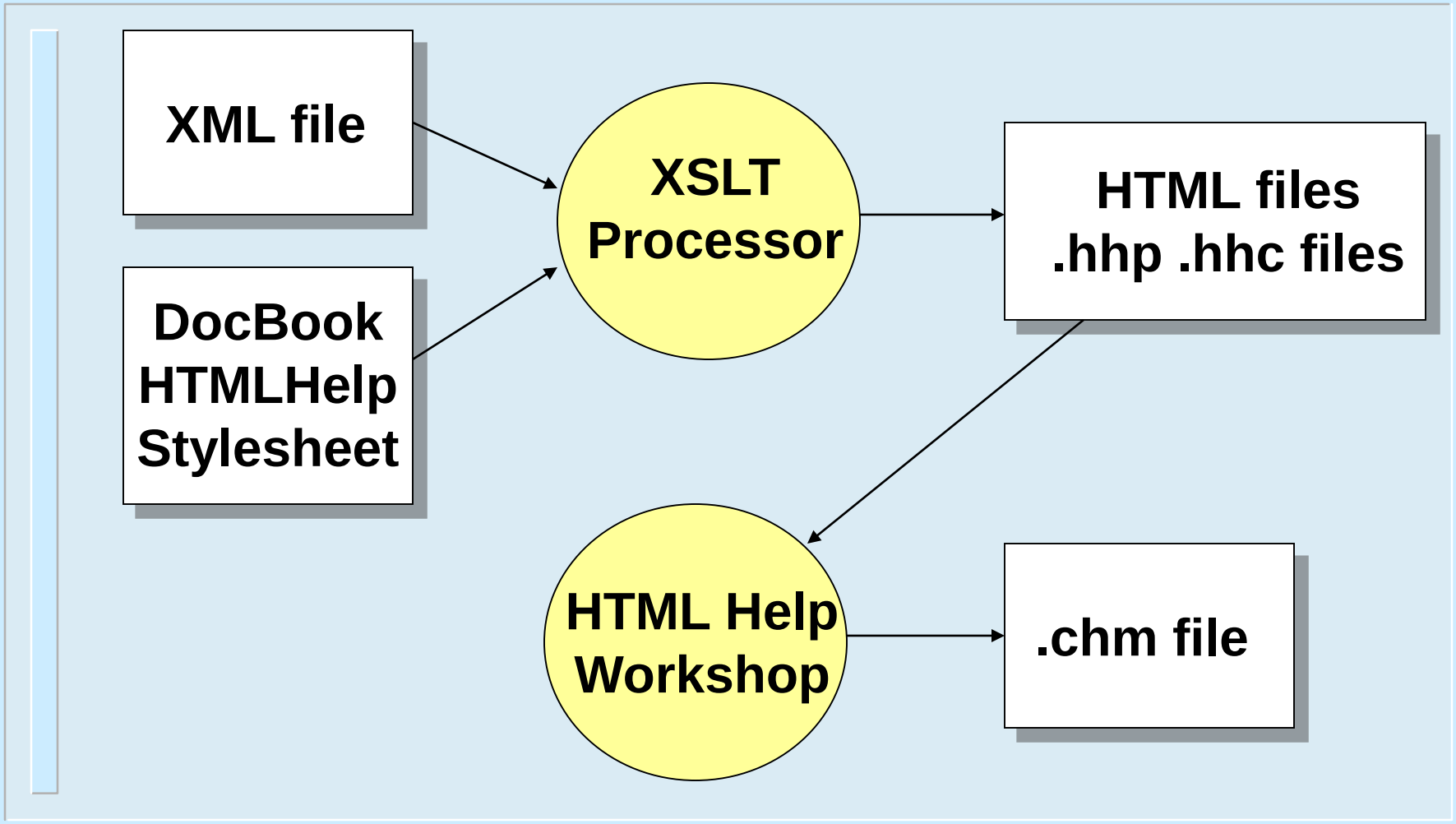
XSLT processors

- xsltproc from Gnome’s xmlsoft.org
- Saxon 6 from SourceForge
- Xalan Java from Apache XML project

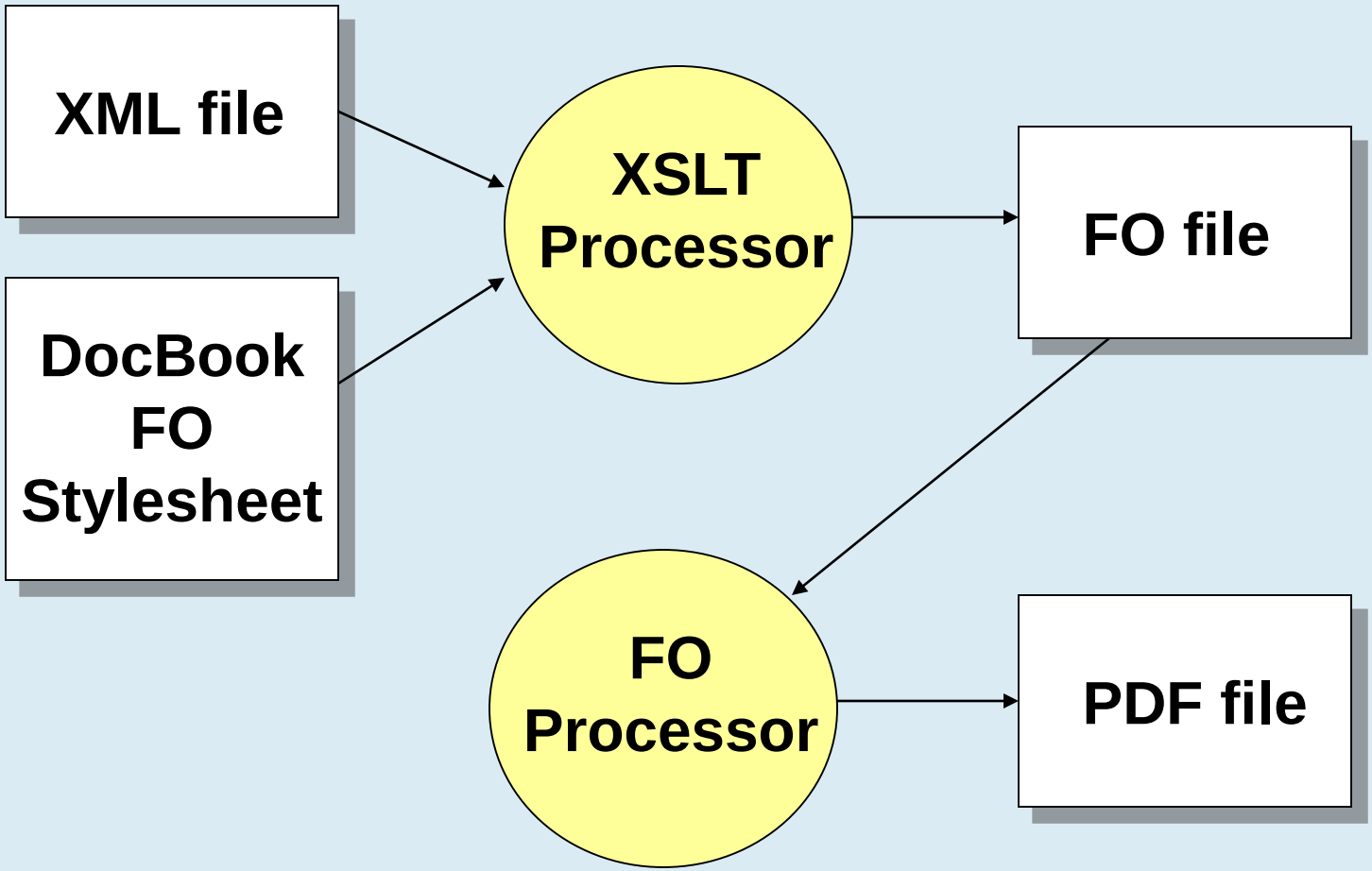
HTML options

- HTML or XHTML
- Single file or multiple “chunks”.
- Use CSS classes for styling.
- Post process with
  - HTML Help Workshop
  - Java Help indexer

HTML Help output



Getting to print



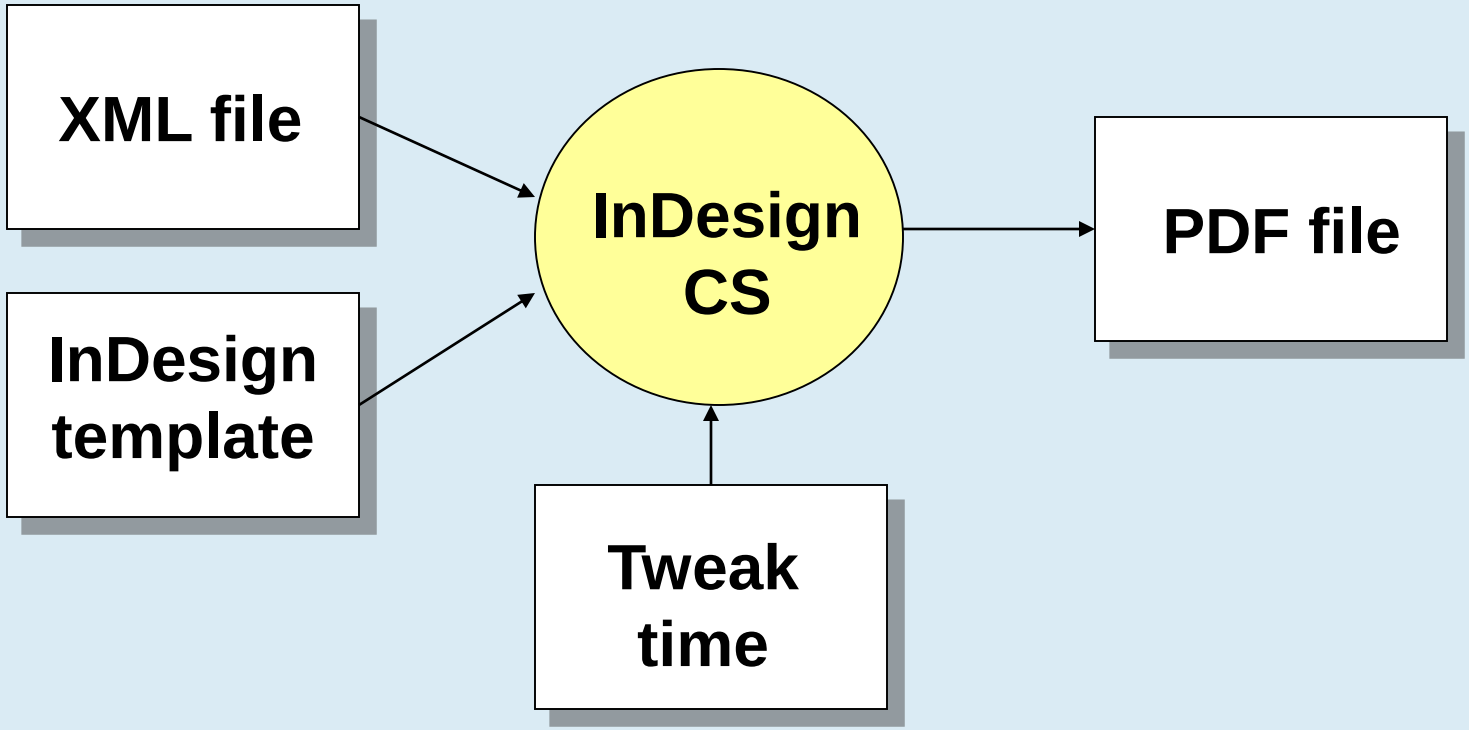
Leading XSL-FO processors

- XEP from RenderX.
- XSL Formatter from Antenna House
- FOP from Apache XML Project

Automated print production

- Stylesheet flows content onto pages.
- Automatic TOC and index.
- Automatic page breaking.
- Writer does not do formatting.

Tweaking for print



Customizable stylesheets

- Modular design (~100 files)
- Reusable named templates.
- Empty, user-defined templates.
- Parameterized (~220 parameters)
- Title page tools.
- Templates for generated text.

Stock DocBook

Chapter 4. Basics

Introduction

Basics include general information about how to:

- Start your Virtual product.
- Work with the product.
- Monitor your product's activities.
- Access more information.

Check the version number

You can check the version number of your product on your computer. Use the version number to help you find more information about your product on the Virtual Web site.

Procedure 4.1. To check the version number

1. Start your product.
2. Click Help and Support.
3. On the Help menu, click About <your product name>.
4. In the About dialog box, select your product name.

Start the application

After installation, the application automatically protects any computer on which it is installed. You do not have to start the program to be protected.

Procedure 4.2. To start the application

- Do one of the following:
  - On the Windows taskbar, click Start > Programs > Virtual Internet Security > Virtual AntiVirus > the application.
  - On the Windows XP taskbar, click Start > All Programs > Virtual Internet Security > the application > the application.
  - On the Windows taskbar, click Start > Programs > Virtual SystemWorks > the application > the application.
  - On the Windows XP taskbar, click Start > More Programs > Virtual SystemWorks > the application > the application.
  - On the Windows taskbar, click Start > Programs > the application > the application 2004.

18

Stock DocBook

Chapter 4. Basics

Introduction

Basics include general information about how to:

- Start your Virtual product.
- Work with the product.
- Monitor your product's activities.
- Access more information.

Check the version number

You can check the version number of your product on your computer. Use the version number to help you find more information about your product on the Virtual Web site.

Procedure 4.1. To check the version number

1. Start your product.
2. Click Help and Support.
3. On the Help menu, click About <your product name>.
4. In the About dialog box, select your product name.

Start the application

After installation, the application automatically protects any computer on which it is installed. You do not have to start the program to be protected.

Procedure 4.2. To start the application

- Do one of the following:
  - On the Windows taskbar, click Start > Programs > Virtual Internet Security > Virtual AntiVirus > the application.
  - On the Windows XP taskbar, click Start > All Programs > Virtual Internet Security > the application > the application.
  - On the Windows taskbar, click Start > Programs > Virtual SystemWorks > the application > the application.
  - On the Windows XP taskbar, click Start > More Programs > Virtual SystemWorks > the application > the application.
  - On the Windows taskbar, click Start > Programs > the application > the application 2004.

18

Customized

Basics 4

Introduction

Basics include general information about how to:

- Start your Virtual product.
- Work with the product.
- Monitor your product's activities.
- Access more information.

Check the version number

You can check the version number of your product on your computer. Use the version number to help you find more information about your product on the Virtual Web site.

To check the version number

1. Start your product.
2. Click **Help and Support**.
3. On the Help menu, click **About <your product name>**.
4. In the About dialog box, select your product name.

Start the application

After installation, the application automatically protects any computer on which it is installed. You do not have to start the program to be protected.

To start the application

- Do one of the following:

Installing XSLT processor

- xsltproc
  - Download libxml2 and libxslt C source.
  - Configure and make.
  - Windows binaries available.
- Saxon and Xalan
  - Need Java runtime environment.
  - Download and add to CLASSPATH.
  - DocBook extensions available.

Using xsltproc

- HTML output:

```
xsltproc \
   --output  myfile.html \
   /usr/share/docbook/html/docbook.xsl \
   myfile.xml
```

- FO output:

```
xsltproc \
   --output myfile.fo \
   /usr/share/docbook/fo/docbook.xsl \
   myfile.xml
```

Using Saxon

- HTML output:

```
java  -cp "/usr/java/saxon.jar:\
  /docbook-xsl/extensions/saxon65.jar" \
  com.icl.saxon.StyleSheet \
  -o  myfile.html \
  myfile.xml \
  /usr/share/docbook/html/docbook.xsl
```

Generating PDF with FOP

- Convert a .fo file on Unix or Linux:

```
fop.sh -fo myfile.fo -pdf myfile.pdf
```

- Convert a .fo file on Windows:

```
fop.bat -fo myfile.fo -pdf myfile.pdf
```

- Convert an XML source file:

```
fop.sh -xsl /docbook-xsl/fo/docbook.xsl \
-xml myfile.xml -pdf myfile.pdf
```

File not found …

- Most common beginner’s problem.
- Many components working together.
- Common problems:
  - Path to DTD in file’s DOCTYPE.
  - Missing XML character entities.
  - Java CLASSPATH missing something.

XML catalog

- Maps generic addresses to specific locations on local machine.
- Portable setup; just edit catalog file when move something.
- Adds flexibility, and complexity.
- SGML catalogs are less capable.

Using parameters

- Named variables in stylesheet.
- Examples:
  - Specify CSS stylesheet name.
  - Turn on section numbering.
  - Shade verbatim output.
- Each processor has its own command syntax.

Parameter example

```
xsltproc  --output myfile.html  \
--stringparam html.stylesheet style.css \
--stringparam  shade.verbatim  1 \
html/docbook.xsl  myfile.xml
```

- Output

```
<link rel="stylesheet" href="style.css"
  type="text/css">

<table border="0" bgcolor="#E0E0E0">
  <tr><td><pre class="literallayout">
```

Customization driver file

- New stylesheet that combines:
  - Standard DocBook XSL templates
  - Your customizations.
- Kept separate, so updates are easy.
- Used in place of DocBook stylesheet.

Driver file content

- Standard XSL stylesheet elements.
- Pull in stock DocBook with:

```
<xsl:import
  href="path/to/docbook.xsl"/>
```

- Parameter settings.
- Replacement templates.
- New generated text.

Customization example

```
<?xml version='1.0'>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/
  Transform"  version="1.0">

<xsl:import href="html/docbook.xsl"/>
<xsl:param name="shade.verbatim"
           select="1"/>
<xsl:template ... </xsl:template>

</xsl:stylesheet>
```

Special features

- Profiling.
- Modular documentation.
- Cross referencing.
- Localization.

Profiling

- *aka* Conditional text.
- Multiple versions from same file.
- Marked with profiling attributes:
  - **`os, userlevel, condition`**, etc.
- Use a profiling stylesheet.
- Select conditions at runtime:

```
profile.os="linux"
```

Modular documentation

- Separate large documents into multiple XML files.
- System entity != valid document.
  - No DOCTYPE allowed.
  - xrefs won’t resolve.
- Use Xinclude instead.
  - Modules can be valid documents.
  - Use olink to link between them.

Cross referencing

- Author or generate hotlink text.
- Define cross reference “styles”.
- Link within or between documents
- Between documents:
  - Within a concurrent set.
  - Reference to previously published doc.
  - Reference to remote content.

Olinking

- Create olink database from documents.
- XSL processor reads the database.
- Can generate link text.
- Can link to base URI per document.
- Creates dependencies between docs.

Localization

- Unicode and other encodings
- Generated text in 44 languages
- Index collation
- Mixed language documents

DocBook community

- Mailing lists
  - **docbook-apps** for processing questions.
  - **docbook** for DTD questions.
- DocBook SourceForge project
  - Stylesheet CVS files
  - Bug reports and feature requests.
- DocBook Wiki
  - User contributions.

Questions?

References:

- *DocBook: The Definitive Guide*
  - https://tdg.docbook.org/tdg/4.5/docbook.html
- *DocBook XSL: The Complete Guide*
  - http://www.sagehill.net/docbookxsl/