

SilkPage User Guide



SilkPage User Guide

Copyright © 2004, 2005, 2006 MarkupWare [<http://www.markupware.com/>]

Permission is granted to copy, distribute and/or modify SilkPage Documentation under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation. A copy of the license is available at <http://www.gnu.org/copyleft/fdl.html>

Table of Contents

Introduction	v
What is SilkPage?	v
Notation and Typographical Conventions	v
1. Installation and Setup	1
System Requirement	1
Prerequisite Software	1
Installation Tasks	1
SilkPage Distribution Download	1
Installing SilkPage Distribution	1
SilkPage Directory Structure	2
2. Usage	4
Site Creation	4
Create Site Scratch	4
Create Site Layout	6
Create Site Pages	8
Site Content Management	13
Main Elements	13
Examples	14
Site Publishing	14
Advanced Features	14
Site Customization	15
XML Feed Integration	15
RDF Integration	17
Auto Sitemap Generation	18
A. Global Configuration Files	19
catalog.xml	19
tidy.properties	19
B. Instance Configuration Files	20
build.properties	20
XSL and CSS Customization Parameters	20
DocBook Parameters	24
FTP Parameters	28
General Parameters	37
siteid.css	41
param.xsl	41
Search Field Parameters	41
Page Source Link Items Parameters	44
config.xsl	51
rdf.xsl	52
catalog.xml	52
C. Command Line Options	53
ant	54
D. Frequently Asked Questions	55
E. Resources	58
Latest Version of SilkPage	58
Prerequisite Software Resources	58
Specifications	58
Tutorials and Introductory Materials	59
Glossary	61

List of Examples

1.1. Installing SilkPage	2
2.1. Create Site Scratch	4
2.2. Site Layout	7
2.3. Page Head	9
2.4. Page Body	12
2.5. Linking Content Pages and Elements	14
2.6. Drafting your Website	14
2.7. Publishing your Website Online	14
2.8. RSS 1.0 Feed	16
2.9. Atom Feed	17

Introduction

This document contains useful information to assist you during the installation, customization and usage of SilkPage.

What is SilkPage?

SilkPage is an XML based Web publishing framework that has evolved from the Docbook Website. With strong foundations in XML, it is a publishing framework with a focus on web standards and usability. SilkPage provides a framework for publishing structured and standard compliant websites in XML. It is particularly suited to publish personal websites, though it is by no means limited to them.

Key Features

Content Management	XML based content management.
WaSP Compliant	Web Standards (WaSP) compliant XHTML output.
Accessibility	SilkPage is developed with accessibility and usability in mind.
Separate Presentation Model	Customization via XSL and CSS customization layers.

Notation and Typographical Conventions

There is a systematic usage of typography and highlighting in this document. Different styles are defined as follows:

Typography Styles

<code>element</code>	An XML element.
<code>attribute</code>	An XML attribute.
<code>filename</code>	Paths, directory and file names.
<code>application</code>	Programs and software application names.
<code>command synopsis</code>	Shell commands that are executed by the user
User Input	Text that the user has to type.
<code>Computer output</code>	Text displayed by the computer on the command line.

Chapter 1. Installation and Setup

System Requirement

- **CPU.** Intel® Pentium® II or higher processor
- **Memory.** 32 MB RAM
- **Disk Space.** 20 MB
- **Network.** Internet connection and a Web Browser

Prerequisite Software

Prior to installing SilkPage, you must first install the following application software and middleware. Detailed instructions for installing and using the below mentioned applications are beyond the scope of this document. However, you'll find appropriate instructions in the corresponding URL(s).

1. Download and Install Java (J2SE) [<http://java.sun.com/j2se>]
2. Download and Install Ant
3. Download and Install DocBook XSL Stylesheets (optional)
4. Download and Install DocBook Website (optional)
5. Download Latest Version of SilkPage

Installation Tasks

This section covers the SilkPage installation tasks. *VERSION* corresponds to the current version of the SilkPage software.

SilkPage Distribution Download

SilkPage comes in two bundles:

- `silkpage-VERSION.zip` only includes SilkPage distribution. If you choose this download file, you also need to download the required DocBook resources mentioned in the section called "Prerequisite Software".
- `silkpage-VERSION-plus.zip` in addition to SilkPage distribution, it also includes the required DocBook resources. It's recommended that you choose this download file as it includes all the required DocBook resources.

Installing SilkPage Distribution

To install SilkPage you only need to unpack the SilkPage distribution file. Unpacking the download file should result the directory structure shown in the section called "SilkPage Directory Structure". Once unpacked, you can test the SilkPage installation by running the `config` target as shown in Example 1.1, "Installing SilkPage".

Example 1.1. Installing SilkPage

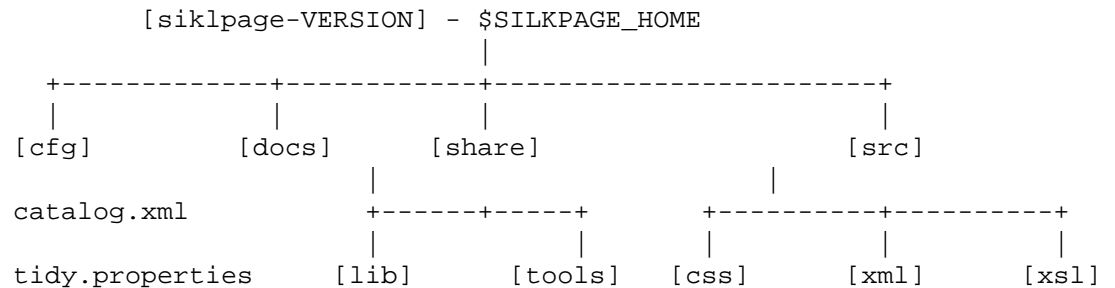
```
unzip silkpage-VERSION-plus.zip
cd silkpage-VERSION
ant -f src/xml/build/tasks.xml config
```

```
config:
 [echo] SilkPage (VERSION)
 [echo] user.name: jhe
 [echo] os.name: Linux
 [echo] temp.dir: /tmp
 [echo] DocBook Website installed ? true
 [echo] DocBook XSL installed ? true
```

Refer to Appendix C, *Command Line Options* for detailed information about SilkPage command line options. For troubleshooting, refer to Q & A D.3, “Installation”.

SilkPage Directory Structure

This section gives an overview of the SilkPage directories and the files they contain. Assuming the SilkPage software is installed at `silkpage-VERSION`, here are some of the most important directories:



SilkPage Directories

`SILK-PAGE_HOME` The installation directory of SilkPage software, referred to as `SILKPAGE_HOME` here after.

- `cfg/` The configuration directory of the SilkPage software.
 - `catalog.xml` used in order to perform Entity and URI resolution.
 - `tidy.properties` contains Tidy properties that are fed to Tidy when it's run by SilkPage to validate the XHTML output.
- `docs/` Contains SilkPage User Guide and documentation.
- `share/` Contains resource files used by SilkPage.

- `lib/` contains code libraries.
- `tools/` contains middleware and tools including DocBook Website and DocBook XSL stylesheets.

`src/` Contains SilkPage source files.

- `css/` contains a set of CSS stylesheets that are used to control the XHTML style and presentation.
- `xml/` contains Ant based build instructions used to generate your Website.
- `xsl/` contains a set of XSL stylesheets used to output XHTML.


```
sitemap.xml
glossary.xml

contact.xml
contact-send.php
contact-sent.xml
search.xml
feeds.xml
projects.xml
docs.xml
```

Scratch Website Directories

`SITE_HOME` is the root directory of the generated directory structure. This directory is referred to as `SITE_HOME` here after.

and your Website themes in terms of CSS and XSL stylesheets.

`build.properties` Contains build properties of your Website such as your Website URL and your website themes in terms of CSS and XSL stylesheets.

`build.xml` Contains build instructions used by Ant to build your Website. It contains the required targets that let you *preview*, *publish* and *upload* your Website.

`src/` The root directory for your Website content where all source files are kept.

- `css/`
 - `siteid.css` for your main Website CSS stylesheet to control your Website style.
- `xml/` The root directory for your Website XML content.
 - `en/` The root directory for your Website XML content in English. Of course, you could translate your Website content in other languages. For example, to add the French translation, you need to translate your XML content in French and put it under the `fr/` directory.
 - `index.xml` your Website main index page.
 - `layout.xml` describes the layout and structure of your Website as well as your Website navigation.
 - `sitemap.xml` contains instructions to generate your Website sitemap based on the `layout.xml`.
 - `glossary.xml` contains a collection of your web site terms and brief descriptions or definitions of those terms. Once you refer to a term in your site content, SilkPage transforms it to an HTML acronym. For example to refer to XML as an acronym, you write `acronym XML /acronym`. Also see `silkpage.acronyms.database.document` configuration parameter.
 - `search.xml`, `contact.xml`, `feeds.xml`, etc. also exist in this directory, serving as webpage examples.
- `xsl/`
 - `config.xsl` contains XSL based customization templates.

- `param.xsl` contains XSL based configuration parameters.
 - `rdf.xsl` is used as a driver stylesheet to generate RDF metadata for each XML source.
- `cfg/` The directory for build configuration files
- `catalog.xml`
- `img/` The directory graphical images of the website

Create Site Layout

After generating the scratch version of your Website, you need to layout your Website and define your required structure. The site layout feature provides a mapping between the XML source files and the generated HTML files. The XML source files can be stored in multiple directories. Note that, the physical location of the XML source files do not reflect the site logical hierarchy. The `dir` attribute of the `tocentry` element can be used to create any set of directory structure for the HTML output. As shown in the section called “Scratch Website Directory Structure”, there is a scratch layout file (`layout.xml`) generated for your Website that can be used as the basis to define the structure of your Website.

Main Elements

- `layout`: the root element of the site table of content.
- `config`: optional elements specifying global configuration information for the whole site. A `config` element is defined by a pair of `param` and `value`. This is a global element. That is it applies to every page of the Website.
- `copyright`: defines copyright information in the footer of each page. This is a global element. That is it applies to every page of the Website.
- `style`: specifies a CSS stylesheet used by all HTML pages of the site. This is a global element. That is it applies to every page of the Website.
- `toc`: the site layout top level element. It specifies the site home page.
- `tocentry`: the hierarchy of `tocentry` elements in a `toc` describes the logical navigation hierarchy of the Website. Each entry must identify the XML document (containing a `webpage`) that it represents and may identify the directory and filename where the file will appear in the HTML Website.
 - `page`: attribute specifies the XML source file used to generate the corresponding HTML file.
 - `filename`: specifies the HTML output filename. If a `filename` is not specified, `index.html` will be used.
 - `dir`: specifies the output directory where the HTML output is generated. if `dir` is not specified, the files will appear in the currently inherited directory (or in the root, if no directory is specified anywhere in the `tocentry` ancestry).
- `notoc`: pages identified with `notoc` elements instead of `tocentry` will be in the Website but will not appear in any hierarchy.

1. `index.html` generated from `index.xml` content.
2. `about/index.html` generated from `about/index.xml` content.
3. `about/colophon.html` generated from `about/colophon.xml` content.
4. `about/contact.html` generated from `about/contact.xml` content.
5. `docs/index.html` generated from `docs/index.xml` content.
6. `downloads/index.html` generated from `downloads/index.xml` content.
7. `sitemap.html` generated from `sitemap.xml`.

Create Site Pages

Once your Website structure and layout is defined, you need to create the corresponding pages in XML. That is your Website content. A SilkPage page is written using DocBook and Website elements. For each Web page there is a corresponding XML file. Each page consists primarily of three sections:

1. *Root*: must start with `webpage` and followed by any optional `config` elements.
2. *Head*: header information.
3. *Body*: page content marked up with DocBook and Website elements.

Create the Page Head

Main Elements

- `webpage`: specifies the page root element; its `id` must be unique in the scope of the site.
- `config`: specifies metadata information in the generated HTML output.
- `head`: contains the page head information:
 - `title`: specifies the title of the generated HTML page.
 - `summary`: used in the `title` attribute of the generated HTML page with the exception of the home page.
 - `keyword`: specifies metadata information for the page.

Examples

Example 2.3. Page Head

```
<?xml version="1.0">
<!DOCTYPE webpage ...>
<webpage id="home">
  <config param="rcsdate" value="$Date: 2006/04/17 16:26:47 $" />
  <head>
    <title>SilkPage</title>
    <summary>An XML based Publishing Framework for Web</summary>
    <keywords>Markup, web standards, W3C, XML, XSL, CSS, XHTML</keywords>
  </head>
</webpage>
```

In the above example, the header information of the SilkPage Website is specified as follows:

- `home` is the unique identifier of the home page.
- `rcsdate` specifies the last modification date of the page. The value of this parameter is set to `$Date: 2006/04/17 16:26:47 $` that is a CVS substitution keyword. Using this approach, SilkPage generates the last modification timestamp of the home page at the bottom of the XHTML output page.
- `title` specifies the page main title.
- `summary` provides a short description for the SilkPage framework.
- `keyword` provides some metadata information used by some search engines.

In addition to what's specified in the above example, SilkPage generates several other optional links in the XHTML output page as follows:

- **XML Source.** Links to the XML source of the page. The label of this link can be changed by setting the value of the configuration parameter `sources.xml`.
- **RDF metadata.** Links to the RDF metadata of the page. The label of this link can be changed by setting the value of the configuration parameter `sources.rdf`.
- **RSS Feed.** Links to the RSS feed of the page. The label of this link can be changed by setting the value of the configuration parameter `sources.rss10`.
- **Atom Feed.** Links to the Atom feed(s) of the page. The label of this link can be changed by setting the value of the configuration parameter `sources.atom`.
- **DOAP Source.** Links to the DOAP source, included in the XML source. The label of this link can be changed by setting the value of the configuration parameter `sources.doap`.
- **URFM.** Links to the URFM file included in the XML source. The label of this link can be changed by setting the value of the configuration parameter `sources.urfm`.
- **FOAF.** Links to the FOAF file included in the XML source. The label of this link can be changed by setting the value of the configuration parameter `sources.foaf`.

Create the Page Body

Main Elements

The page content is marked up using DocBook Website elements defined in `website.dtd`. Refer to the online documentation of Simplified DocBook DTD [<http://www.oasis-open.org/docbook/xml/simple/sdocbook/>] for available elements and their syntax.

Examples

Example 2.4. Page Body

```
<?xml version="1.0">
<!DOCTYPE webpage ...>
<webpage id="home">
  <config param="desc" value="The home index page"/>
  <config param="rcsdate" value="$Date: 2006/04/17 16:26:47 $" />
  <head>
    <title>SilkPage</title>
    <summary>An XML based Publishing Framework for Web</summary>
    <keywords>Markup, web standards, W3C, XML, XSL, CSS, XHTML</keywords>
  </head>
  <para>
    SilkPage is an XML based Web publishing framework that has evolved from
    the Docbook Website. With strong foundations in XML, it is a publishing
    framework with a focus on web standards and usability.
    SilkPage provides a framework for publishing structured and standrad
    compliant websites in XML. It is particularly suited to publish personal
    websites, though it is by no means limited to them.
  </para>
  <sidebar>
    <variablelist>
      <title>Key Features</title>
      <varlistentry>
        <term>Conent Management</term>
        <listitem>
          <para>
            XML based content management.
          </para>
        </listitem>
      </varlistentry>
      <varlistentry>
        <term>WaSP Compliant</term>
        <listitem>
          <para>
            Web Standards (WaSP) compliant XHTML output in addition to XML and
            PDF outputs.
          </para>
        </listitem>
      </varlistentry>
      <varlistentry>
        <term>Content Feeds Support</term>
        <listitem>
          <para>
            Integrated XML and RSS feed generation.
          </para>
        </listitem>
      </varlistentry>
      <varlistentry>
        <term>Separate Presentation Model</term>
        <listitem>
          <para>
            Customization via XSL and CSS customization layers.
          </para>
        </listitem>
      </varlistentry>
    </variablelist>
  </sidebar>
</webpage>
```

```
<varlistentry>
  <term>Cross Platform Support</term>
  <listitem>
    <para>
      Cross platform support via industry standards.
    </para>
  </listitem>
</varlistentry>
</variablelist>
</sidebar>
</webpage>
```

In the above example the `para` element is used to describe the SilkPage framework briefly. In addition, the `sidebar` element is used to list the key features of the SilkPage framework.

Site Content Management

SilkPage maintains metadata information about your site architecture, pages and their links. This information is kept in the `layout.xml` discussed in the section called “Create Site Layout”. Using this approach, SilkPage separates the site content from its logical structure. Site pages and documents are linked using `olink` that makes it possible to locate the target element in the site content and generate the site links automatically. `olinks` are resolved from a database of targets that is maintained by SilkPage. This solution reduces link maintenance of your Website significantly.

Main Elements

- `olink`: the main element for linking pages and elements.
 - `targetdoc` the `id` of the target webpage that must exist and its `id` must be unique.
 - `targetptr` the `id` attribute value of the target element within the target webpage. In order to link to the top of a given page, the value of the `targetptr` must be the same as the `targetdoc`.
- `ulink`: mainly used to directly link to external HTML pages and documents. Such links are not verified while processing, so they require manual maintainance in order to be valid URLs. Therefore, it's bad idea to use `ulink` to link directly to the generated XHTML pages; if you change the hierarchy or rename a page, the link will become stale. With `olink` this won't happen.
 - `url` the URL of the target HTML document.

Examples

Example 2.5. Linking Content Pages and Elements

```
<para>
SilkPage is an XML based Web publishing framework that has evolved from the
<ulink url="http://sourceforge.net/projects/docbook">Docbook Website</ulink>.
With strong foundations in XML, it is a publishing framework with a focus on
web standards and usability. It's mainly used to
publish personal and Websites. Refer to the SilkPage
<olink targetdoc="docs">User Guide</olink> for more information.
</para>
```

In the above example, a `ulink` is used to link to the DocBook Website. However, to link to the main index page of the documentation, a `olink` is used. Note that you don't have to do anything special to the page you're linking to. The only important thing is a valid entity id, in this case `docs`.

Site Publishing

SilkPage uses Ant to automate your site generation, validation and upload.

- Based on your XML content, SilkPage *generates* XHTML output.
- SilkPage uses Tidy [] to *validate* and eventually fix the generated XHTML.
- SilkPage facilitates the *upload* of your Website by using a FTP based file transfer. Assuming that you have used the *scratch* target like it was shown in Example 2.1, “Create Site Scratch”, you need to run SilkPage with the *preview* target to have the draft version of your Website generated.

Example 2.6. Drafting your Website

```
cd coolsite
ant preview
```

As shown in the above example, you can generate a draft version of your Website locally. This local version will be used to upload your Website when running SilkPage with the *publish* target. In order to publish your Website online, you need to run SilkPage with the *publish* target, as shown in Example 2.7, “Publishing your Website Online”.

Example 2.7. Publishing your Website Online

```
cd coolsite
ant publish
```

Refer to Appendix C, *Command Line Options* for detailed information about SilkPage command line options.

Advanced Features

As Web sites become increasingly sophisticated, they demand advanced requirements such as XML Feed integration, metadata rendering, etc. SilkPage is designed so that it's easy to customize to fit your special requirements. You can customize SilkPage at XSL and CSS layers. This section covers SilkPage customization and advanced features such as XML and Feed integration.

Site Customization

SilkPage allows stylesheet customization for a particular Website. That is customizing with XSL as well as CSS stylesheets. These customizations are integrated to SilkPage via *themes*. SilkPage provides two types of themes:

- *XSL Theme*: a set of XSL stylesheet grouped into a directory. This directory name is used by SilkPage to identify the desired XSL theme and use it when generating XHTML output.
- *CSS Theme*: a set of CSS stylesheet grouped into a directory. directory name is used by SilkPage to identify the desired CSS theme and use it when presenting XHTML output.

XSL Driven Customization Via XSL Themes

As indicated in the previous section, you can define and your own XSL theme and have SilkPage use it to generate your Website.

1. Write your desired XSL stylesheets and put them in a directory. The name of this directory is used to identify your newly created XSL theme.
2. install it by copying the directory to `SILKPAGE_HOME/src/xsl` . Note that `SILKPAGE_HOME` is the install directory of the SilkPage software.
3. set the value of the configuration parameter: `silkpage.adapt.xsl.theme` to your newly created theme.

CSS Driven Customization Via CSS Themes

Like XSL themes, a new CSS theme can be defined and used as follows:

1. Write your desired CSS stylesheets and put them in a directory. The name of this directory is used to identify your newly created CSS theme.
2. install it by copying the directory to `SILKPAGE_HOME/src/css`. Note that `SILKPAGE_HOME` is the install directory of the SilkPage software.
3. set the value of the configuration parameter: `silkpage.adapt.css.theme` to your newly created theme.

XML Feed Integration

A *feed* is a wrapper for pieces of regularly and sequentially-updated content such as news headlines, articles, weblog posts, and so on. Atom and RSS are the two flavours of XML feeds. They are both supported by SilkPage. Once integrated into your site, a news reader application such as Bloglines, RSSOwl, or even FireFox can be used to automatically check your list of feeds. This enables your visitors to know what's new on your site.

RSS Feed Integration

RSS is an application of XML, that provides an open method of syndicating and aggregating Web content. Using RSS files, you can create a data feed that supplies headlines, links, and article summaries from your Website.

Example 2.8. RSS 1.0 Feed

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://purl.org/rss/1.0/"
        xmlns:dc="http://purl.org/dc/elements/1.1/">

<channel rdf:about="http://www.markupware.com/">
  <title>MarkupWare Projects</title>
  <link>http://www.markupware.com/projects/</link>
  <description>MarkupWare project portfolio</description>
  <dc:language>en-us</dc:language>
  <dc:rights>Copyright &#169; MarkupWare</dc:rights>

  <items>
    <rdf:Seq>
      <rdf:li rdf:resource="http://www.markupware.com/projects#silkpage"/>
    </rdf:Seq>
  </items>
</channel>

<item rdf:about="http://www.markupware.com/projects#silkpage">
  <title>SilkPage Project</title>
  <link>http://silkpage.markupware.com</link>
  <dc:date>2004-09-04</dc:date>
  <description>
    SilkPage is an XML based Web publishing framework that has evolved from
    the Docbook Website. With strong foundations in XML, it is a publishing
    framework with a focus on web standards and usability. SilkPage provides
    a framework for publishing structured and standard compliant websites in
    XML. It is particularly suited to publish personal websites, though it
    is by no means limited to them.
  </description>
</item>

</rdf:RDF>
```

The above example, defines a RSS 1.0 compliant feed for the project section of the MarkupWare Website. Based on the above source, SilkPage generates proper XHTML output page with a link to the RSS feed. The label of this link can be changed by setting the value of the configuration parameter `sources.rss10`.

Atom Feed Integration

RSS is an application of XML, that provides an open method of syndicating and aggregating Web content. Using RSS files, you can create a data feed that supplies headlines, links, and article summaries from your Website.

Example 2.9. Atom Feed

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>MarkupWare Projects</title>
  <subtitle>MarkupWare project portfolio</subtitle>
  <link rel="alternate" type="text/html"
        href="http://www.markupware.com/projects/" />
  <link rel="self" type="application/atom+xml"
        href="http://www.markupware.com/feeds/projects.xml" />
  <id>http://www.markupware.com/feeds/projects.xml</id>
  <updated>2006-01-26T11:26:54+01:00</atom:updated>
  <author>
    <atom:name>MarkupWare Team</atom:name>
  </atom:author>
  <dc:language>en-us</dc:language>
  <dc:rights>Copyright &#169; MarkupWare</dc:rights>

  <entry>
    <title type="text">SilkPage</title>
    <id>http://www.markupware.com/projects#silkpage</id>
    <link rel="alternate" type="text/html"
          href="http://silkpage.markupware.com/" />
    <published>2004-12-20T20:19:41+01:00</published>
    <updated>2005-07-15T23:34:34+01:00</updated>
    <content type="text">
      SilkPage is an XML based Web publishing framework that has evolved from
      the Docbook Website. With strong foundations in XML, it is a publishing
      framework with a focus on web standards and usability. SilkPage provides
      a framework for publishing structured and standard compliant websites in
      XML. It is particularly suited to publish personal websites, though it
      is by no means limited to them.
    </content>
  </entry>
</feed>
```

The above example, defines an Atom compliant feed for the project section of the MarkupWare Website. Based on the above source, SilkPage generates proper XHTML output page with a link to the Atom feed. The label of this link can be changed by setting the value of the configuration parameter `sources.atom`.

RDF Integration

RDF is the World-Wide Web, specification, data (RDF). A specification being developed in 2000 by the W3C as a foundation for processing meta-data regarding resources on the Internet, including the World-Wide Web. SilkPage generates RDF (along with a link pointed to the generated RDF) for each page of your Website based on the XML content of the page. The label of the RDF link can be changed by setting the value of the configuration parameter `sources.rdf`.

Auto Sitemap Generation

Sitemap is an important help for your Website visitors to find their way around your site. SilkPage generates and maintains your Website sitemap. A `config` element should be added to the `layout.xml` in order to have the sitemap generated (See Example 2.2, “Site Layout”). SilkPage maintains and provides a link to your Website sitemap.

Note

A `tocentry` or `notoc` must be added to the `layout.xml`, pointing to `sitemap.xml` contained in your Website source directory.

Appendix A. Global Configuration Files

This section describe each of the global configuration parameter used in SilkPage. This information is most useful when configuring SilkPage globally after the installation.

SILKPAGE_HOME/cfg/catalog.xml

It is a catalog of public resources such as DTDs or entities that are referenced in an XML document. It's typically used to make web references to resources point to a locally cached copy of the resource. SilkPage uses this catalog to perform Entity and URI resolution. Please refer to the Catalogs Specification for more information about the usage and syntax of `cfg/catalog.xml`.

SILKPAGE_HOME/cfg/tidy.properties

SilkPage uses Tidy to validate and cleanup the XHTML output. `tidy.properties` is located in the `cfg/` directory and it is the Tidy configuration file to control its parameters. To find out about the Tidy parameters and options that can be defined in `tidy.properties`, please refer to Tidy Website.

Appendix B. Instance Configuration Files

This section describe each of the configuration parameter used by SilkPage to publish an instance of your website. This information is most useful when publishing a particular instance of a Website.

SITE_HOME/build.properties

`SITE_HOME /build.properties` contains main configuration of an instance website.

XSL and CSS Customization Parameters

This section covers the customization and configuration parameters at the XSL and CSS layers. SilkPage uses *SilkPage XSL Themes* to process and generate your website. You may either use the default XSL themes or plug in yours. Likewise, *SilkPage CSS Themes* are used in order to present your website. Again, you may either use default SilkPage CSS Themes or define and use yours.

Name

`silkpage.adapt.xsl.main` -- XSL configuration parameters and customization templates

```
silkpage.adapt.xsl.main=  
${user.xsl.dir}/config.xsl
```

Description

Specifies the main XSL configuration parameters and customization templates used when generating your website. Refer to `config.xsl` and `param.xsl` to learn about the configuration parameters and templates of this file.

Name

`silkpage.adapt.xsl.rdf` -- XSL configuration parameters related to RDF output

```
silkpage.adapt.xsl.rdf=  
${user.xsl.dir}/rdf.xsl
```

Description

Specifies the main XSL configuration parameters used when generating your RDF metadata for your website. Refer to `rdf.xsl` to learn about the configuration parameters of this file.

Name

`silkpage.adapt.css.theme` -- Specifies the CSS Theme used to generate your website. SilkPage comes with some pre-defined CSS Themes that you can select. However, you can write your own CSS Theme and install it under `SILKPAGE_HOME/src/css` . Note that `SILKPAGE_HOME` is the install directory of the SilkPage software. Currently, SilkPage includes the following CSS themes.

`silkpage.adapt.css.theme=CSS Theme`

CSS Themes

- gimp
- mozilla
- silkpage
- xwtk
- markupware
- phenix

Name

`silkpage.adapt.xsl.theme` -- Specifies the XSL Theme used to generate your website. SilkPage comes with some pre-defined XSL Themes that you can select. However, you can write your own XSL Theme and install it under `SILKPAGE_HOME/src/xsl`. Note that `SILKPAGE_HOME` is the install directory of the SilkPage software. Currently, SilkPage includes the following XSL themes.

```
silkpage.adapt.xsl.theme=XSL Theme
```

XSL Themes

- gimp
- mozilla
- silkpage
- xwtk
- markupware
- ala

DocBook Parameters

This section covers configuration parameters related to DocBook resources. That is the installation directories of the required DocBook resources.

Name

docbook.xsl.home -- Docbook XSL Stylesheets installation directory.

docbook.xsl.home=fullpath

Description

Specifies the installation directory of the DocBook XSL Stylesheets. This should be a fully qualified path.

Name

docbook.website.home -- Docbook Website DTD and Stylesheets installation directory.

docbook.website.home=fullpath

Description

Specifies the installation directory of the DocBook Website DTD and Stylesheets. This should be a fully qualified path.

Name

`silkpage.website.database.document` -- An internal database for resolving links.

`silkpage.website.database.document=fullfilename`

Description

Specifies the fully qualified name to an auto-generated database file used by SilkPage for resolving links. This should be a fully qualified path. On Windows platforms, the path MUST start with a '/.

Name

`silkpage.acronyms.database.document` -- The global glossary file for resolving terms and acronyms.

`silkpage.acronyms.database.document=fullfilename`

Description

Specifies the fully qualified name to the site glossary file. This file may contain a collection of terms and brief descriptions or definitions of those terms. Once you refer to a term in your site content, SilkPage transforms it to an HTML acronym.

FTP Parameters

This section covers the configuration parameters used by the FTP task. This task is used when running SilkPage with the *publish* target. That is when publishing your Website online.

Name

ftp.server -- Specifying ftp server address

ftp.server=FTP Hosts erver

Description

The address of the remote ftp server where you Website is located.

Name

ftp.port -- Specifying FTP Port

ftp.port=number

Description

The port number of the remote ftp server. Defaults to port 21.

Name

ftp.remotedir -- Specifying where your website will be located the ftp server

ftp.remotedir=path

Description

Remote directory on the ftp server. That is your home directory on the Web server.

Name

ftp.userid -- The login id to use on the ftp server.

ftp.userid=userid

Description

Specifies the login id to use on the ftp server.

Name

ftp.password -- The login password to use on the ftp server.

ftp.password=password

Description

Specifies the login password to use on the ftp server.

Name

ftp.depends -- Transfers only new or changed files if set to "yes". Defaults to "no".

ftp.depends=yes/no

Description

If specified transfers only new or changed files if set to "yes". Defaults to "no".

Name

ftp.passive -- Specifies passive-mode ("yes") transfers. Defaults to "no".

ftp.passive=yes/no

Description

Specifies passive-mode ("yes") transfers. Defaults to "no".

Name

ftp.binary -- Specifies binary-mode ("yes") or text-mode ("no") transfers. Defaults to "yes".

ftp.binary=yes/no

Description

Specifies binary-mode ("yes") or text-mode ("no") transfers. Defaults to "yes".

Name

`ftp.localdir` -- The local directory where XHTML files are located. By default, it's set to `#{dist.dir}`. It's recommended not to change the value of this parameter.

```
ftp.localdir=fullpath
```

Description

The local directory where XHTML files are located. By default, it's set to `#{dist.dir}`. It's recommended not to change the value of this parameter.

General Parameters

This section covers the general configuration parameters that are used when running SilkPage with various targets.

Name

`default.encoding` -- The encoding used by various tasks. Defaults to *UTF-8*. Please refer to Character Sets and Encoding [<http://www.iana.org/assignments/character-sets>] to learn more about available encoding.

default.encoding=encoding

Description

The encoding used by various Ant tasks.

Name

default.lang -- The ISO language code used to locate the XML source files.

default.lang=ISO Language Code

Description

The ISO language code used to locate the XML source files. Defaults to *en*. Please refer to Code for the representation of names of languages [<http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt>] to learn more about available ISO 639 language codes.

Name

default.temp.dir -- The temporary directory used for performing the build tasks.

```
default.temp.dir=fullpath
```

Description

The temporary directory used for performing the build tasks. Defaults to the default temp. directory of your local host.

Name

default.url -- Your Website URL.

```
default.url=URL
```

Description

Your Website URL. Defaults to the *site.id* that you provide when running SilkPage with *scratch* target.

SITE_HOME/src/css/siteid.css

When running SilkPage with the *scratch* target, it generates (among other things) a default CSS stylesheet for your Website. The name of the CSS stylesheet is based on the *siteid* argument that you provide when running SilkPage with the *scratch* target. This stylesheet contains one `import` declaration that is an instruction to import some other default style sheets from the *SilkPage CSS Themes*. You may modify the `import` declaration and instruct it to import your own CSS stylesheets. You can even add your CSS stylesheets globally by defining your own CSS Theme and instruct SilkPage to use your CSS themes instead of its default. Refer to the section called “XSL and CSS Customization Parameters” to learn how you can customize SilkPage by adding your CSS themes.

SITE_HOME/src/xsl/param.xsl

`SITE_HOME / src / xsl / param . xsl` is the main XSL configuration parameter file used by SilkPage when processing your XML sources.

Search Field Parameters

SilkPage can generate a Google [<http://www.google.com>] based search field for your Website. This section covers this search field and its associated parameters.

Name

pages.search-box -- Enables/Disables the generation of the search field for your Website. Defaults to 1, that is enabled.

```
<xsl:param name="pages.search-box" select="1"/>
```

Description

Specifies

Name

pages.search.input-label -- Specifies the label text that is displayed right before the search field.

```
<xsl:param name="pages.search.input-label" select="''"/>
```

Description

Specifies the label text that is displayed right before the search field. Defaults to an empty label.

Name

pages.search.submit-label -- Specifies the label text that is displayed as the search submit button label.

```
<xsl:param name="pages.search.submit-label" select="Search"/>
```

Description

Specifies the label text that is displayed as the search submit button label. Defaults to *Search* label.

Page Source Link Items Parameters

If instructed, SilkPage generates link items for XML source and RDF metadata for each generated Webpage and link items for Atom feeds when specified, in XHTML format. This section covers the configuration parameters for these link items.

Name

sources.rdf -- Specifies the text for the RDF metadata link item. Defaults to "RDF" label.

```
<xsl:param name="sources.rdf" select="RDF"/>
```

Description

Specifies the text for the RDF metadata link item. Defaults to "RDF" label.

Note

This parameter also controls the generation of this link item. If no value is specified for this parameter, the link won't be displayed.

Name

sources.rss10 -- Specifies the text for the RSS feed link item.

```
<xsl:param name="sources.rss10" select="RSS"/>
```

Description

Specifies the text for the RSS feed link item. Defaults to "RSS" label.

Note

This parameter also controls the generation of this link item. If no value is specified for this parameter, the link won't be displayed.

Name

sources.xml -- Specifies the text for the XML source link item.

```
<xsl:param name="sources.xml" select="XML"/>
```

Description

Specifies the text for the XML source link item. Defaults to "XML" label.

Note

This parameter also controls the generation of this link item. If no value is specified for this parameter, the link won't be displayed.

Name

sources.urfm -- Specifies the text for the URFM source link item.

```
<xsl:param name="sources.urfm" select="URFM"/>
```

Description

Specifies the text for the URFM source link item. Defaults to "URFM" label.

Note

This parameter also controls the generation of this link item. If no value is specified for this parameter, the link won't be displayed.

Name

sources.doap -- Specifies the text for the DOAP source link item.

```
<xsl:param name="sources.doap" select="DOAP"/>
```

Description

Specifies the text for the DOAP source link item. Defaults to "DOAP" label.

Note

This parameter also controls the generation of this link item. If no value is specified for this parameter, the link won't be displayed.

Name

sources.foaf -- Specifies the text for the FOAF source link item.

```
<xsl:param name="sources.foaf" select="FOAF"/>
```

Description

Specifies the text for the FOAF source link item. Defaults to "FOAF" label.

Note

This parameter also controls the generation of this link item. If no value is specified for this parameter, the link won't be displayed.

Name

sources.atom -- Specifies the text for the Atom feed link item.

```
<xsl:param name="sources.atom" select="Atom"/>
```

Description

Specifies the text for the Atom source link item. Defaults to "Atom" label.

Note

This parameter also controls the generation of this link item. If no value is specified for this parameter, the link won't be displayed.

SITE_HOME/src/xsl/config.xsl

SITE_HOME /src/xsl/config.xsl is the main XSL customization templates file used by SilkPage when processing your XML sources.

Name

user.search-box -- Defines an XSL template containing your Website specific information that is used by Google when searching your Website.

```
<xsl:template name="user.search-box"> ... </xsl:template>
```

Description

Defines an XSL template containing your Website specific information that is used by Google when searching your Website. By default, the URL of your Website specified in default.url is used in this template.

SITE_HOME/src/xsl/rdf.xml

SITE_HOME /src/xsl/rdf .xml is used as a driver stylesheet to generate RDF metadata for each XML source.

SITE_HOME/cfg/catalog.xml

It is a catalog of public resources such as DTDs or entities that are referenced in an XML document. It's typically used to make web references to resources point to a locally cached copy of the resource. SilkPage uses this catalog to perform Entity and URI resolution. Please refer to the Catalogs Specification for more information about the usage and syntax of cfg/ catalog.xml.

Appendix C. Command Line Options

This section covers the complete list of command line targets and options of SilkPage used to publish websites. SilkPage command is based on Ant. Before running SilkPage targets, make sure that you have installed the Prerequisite software listed in the section called “Prerequisite Software”. Moreover, refer to the section called “`build.properties`” in order to learn about parameters' configuration that may effect the command line targets.

Name

ant -- Processes SilkPage command line targets options used for generating and publishing websites.

ant [config] [preview] [publish] [scratch]

Targets

scratch

uses the pre-defined SilkPage templates to generate the first draft of your website in XML format. This target must be executed in the `SILKPAGE_HOME` directory and takes two arguments:

1. `-f src/xml/build/tasks.xml`
2. `-Dsite.id=identifier`

Once executed, it generates the directory structure shown in the section called “Scratch Website Directory Structure” that can be used as a start point for your website development.

Full command line:

```
ant -f src/xml/build/tasks.xml -Dsite.id=identifier scratch
```

preview

After executing the the section called “*scratch*” target, you need to change directory to `SITE_HOME` and execute the preview target in order to generate and install a local copy of your website in XHTML format. The local website is installed relative to the `temp.dir` directory. If not specified, the local host *default temp dir* is used.

Full command line:

```
ant preview
```

publish

generates a local copy of your website and publishes it online by uploading the generated files. Before executing this target, make sure that you have set the configuration parameters of the section called “FTP Parameters”. This target must be executed in the `SITE_HOME` directory and takes two arguments that are the login and password of the FTP server of your website:

1. `-Dftp.userid=login`
2. `-Dftp.password=password`

Full command line:

```
ant -Dftp.userid=login -Dftp.password=password publish
```

config

prints configuration information about the SilkPage and its prerequisite software mentioned in the section called “Prerequisite Software”. This target must be executed in `SILKPAGE_HOME` directory and takes the following argument:

```
-f src/xml/build/tasks.xml
```

Full command line:

```
ant -f src/xml/build/tasks.xml config
```

Appendix D. Frequently Asked Questions

This FAQ is provided with the hopes to be helpful for users using SilkPage. The information is updated as the questions and answers becomes appearant to the SilkPage team.

D.1. General Information

What is SilkPage

Refer to the section called “What is SilkPage?” section.

For whom is SilkPage intended

SilkPage is for everyone who believes in Open Standards, XML and quality. The only pre-requisite to use SilkPage is some basic knowledge of XML, to be able to encode your website content. While the SilkPage software is well documented, it's (cuurently) based on a command line executions. If you can't sand command line executions, then SilkPage is not for you.

I don't know anything about XML, can I still use SilkPage

No, to use SilkPage, you need to provide your Website content in XML format. Therefore, you need to know the XML (very) basics. Refer to A Technical Introduction to XML [<http://nwalsh.com/docs/articles/xml>] to get started.

What are the differeces between SilkPage and DocBook Website

SilkPage is based on DocBook Website. Both applications share almost the identical features, such as Website publishing. However, SilkPage provides some extra features:

- Generates Web Standards (WaSP) compliant XHTML output
- Provides a better Customization via Extensible Stylesheet Language Family (XSL) [<http://w3.org/Style/XSL>] and CSS customization layers.
- Prvides a better support for XML feeds such as RSS and RDF.
- Automates the build and website publising by using ant tasks.

On what is SilkPage based

SilkPage is mainly based on DocBook Website.

What is the relation between SilkPage releases and DocBook Website

Piror to every release, SilkPage team takes the current version of DocBook Website releases, makes modifications (if necessary), and prepares its own release. The DocBook Website output may have a different look/feel than SilkPage output. Moreover, SilkPage provides extra features not present in DocBook Website.

D.2. Usage and Customization

How do I create my first Website

1. the section called “Create Site Scratch” Create a brand new site from scratch
2. the section called “Create Site Layout” Structure your Website by organizing your XML content

3. the section called “Create Site Pages” Create XML source pages and content

How do I migrate my website from DocBook Website to SilkPage

SilkPage is based on DocBook Website. The only thing you need to do to port your website to SilkPage is to copy the content of your website under `SILKPAGE_HOME/src/en`. Note that `en` is the two letter ISO code for languages. You can use the language of your choice. Refer to `default.lang` for valid language codes.

How can I change the link labels of the footer items generated by SilkPage

Refer to the section called “Page Source Link Items Parameters” to learn how you can control this feature.

SilkPage generates a Google based search field for my website, how can I turn it off

Refer to the section called “Search Field Parameters” to learn how you can control this feature.

What is RSS

RSS is a format for syndicating news and the content of news-like sites, including major news sites like Wired, news-oriented community sites like Slashdot, and personal weblogs. Refer to All About RSS [<http://www.faganfinder.com/search/rss.shtml>] for a complete description.

What are all RSS formats and versions ? Which one should I choose

There are 0.90 and 0.91 by Netscape [<http://my.netscape.com/publish/formats/rss-spec-0.91.html>], 1.0 by RSS-DEV [<http://groups.yahoo.com/group/rss-dev>] and 0.9x and 2.0 by UserLand Software [<http://www.userland.com/>]. Refer to the What's RSS Article [<http://www.xml.com/lpt/a/2002/12/18/dive-into-xml.html>] to find out more.

D.3. Installation

What is required to run SilkPage

Refer to the section called “System Requirement”.

How do I install the SilkPage software

It's simple, for all platforms you can download the .zip archive file and unpack it on the hard disk using your favorite tool. See the section called “Installing SilkPage Distribution” for more information.

On what platforms SilkPage can be installed and run

Any operating system supporting Java (J2SE), Standard Edition. However, it should be mentioned that we use GNU/Linux as our reference operating system.

I can't even run SilkPage with the `config` target to test my installation, what's wrong

The most common hiccup is when `JAVA_HOME` or/and `ANT_HOME` are not set. Make sure that you have done the following before installing and running SilkPage:

1. Download the J2SE release (version 1.4 or later) from: <http://java.sun.com/j2se>
2. Install the J2SE according to the instructions included with the release.
3. Set an environment variable `JAVA_HOME` to the pathname of the directory into which you installed the J2SE release.
4. Download the ant release (version 1.6.1 or later) from: <http://ant.apache.org>
5. Install Ant according to the instructions included with the release.

6. Set an environment variable `ANT_HOME` to the pathname of the directory into which you installed the ANT release.

In addition, make sure that the section called “DocBook Parameters” are set correctly.

Appendix E. Resources

This appendix provides the resource references mentioned explicitly in this document. In addition, a sampling of resources for additional information about XML, XSL and CSS is provided.

Latest Version of SilkPage

The most recent version of SilkPage Software and its guide (that is this document) can be found at:

SilkPage Software
[\[http://www.silkpage.org\]](#) SilkPage Core distribution.

SilkPage Document-
ation
[\[http://www.silkpage.org\]](#) The most up-to-date version of this document.

SilkPage Templates
[\[http://www.silkpage.org\]](#) SilkPage Templates distribution.

Prerequisite Software Resources

Apache Ant
[\[http://ant.apache.org\]](#) Apache Ant is a Java-based build tool. In theory, it is kind of like Make, but without Make's wrinkles.

Java SE
[\[http://java.sun.com/javase\]](#) Java Platform, Standard Edition (Java SE, formerly known as J2SE) offers a complete environment for application development and deployment on desktops and servers.

DocBook XSL Stylesheets
[\[http://docbook.sourceforge.net/release/xsl/current/html/index.html\]](#) The DocBook XSL stylesheets are a set of stylesheets for use with an XSLT engine (such as xsltproc or Saxon) for transforming DocBook XML documents into other DocBook output formats, such as HTML, PDF, Microsoft HTML Help, and man pages.

SAXON, the XSLT and XQuery Processor
[\[http://saxon.sourceforge.net\]](#) Saxon is an Open Source Java XSLT and XQuery processor created by Michael Kay.

Apache XML Commons Resolver package
[\[http://commons.apache.org/xmlresolver/\]](#) An XML Entity and URI Resolver donated by Norman Walsh to the Apache Software Foundation.

JTidy
[\[http://jtidy.sourceforge.net\]](#) JTidy is a Java port of HTML Tidy, a syntax checker and pretty printer for HTML.

Specifications

Cascading Style Sheets, Specifies level 1 of the Cascading Style Sheet mechanism (CSS1). Covers all the basic properties and simple selectors which work reliably in most modern browsers.

Level 1

<http://www.w3.org/TR/CSS1>

Cascading Style Sheets, Level 2 (Revision 1) Specifies level 2 (Revision 1) of the Cascading Style Sheet mechanism (CSS2.1). Released after CSS1, adds a whole new set of powerful features such as selectors, visibility properties, media types and positioning.

<http://www.w3.org/TR/CSS2/>

XHTML 1.0 The specification for XML which behaves just like HTML in all browsers. Follows a stricter rule set, and thus, is easier to error-check using common HTML validators.

<http://www.w3.org/TR/xhtml1/>

XML 1.0 The W3C technical recommendation that defines XML 1.0.

<http://www.w3.org/TR/REC-xml/>

The Atom Syndication Format The specification for Atom, an XML-based Web content and metadata syndication format.

<http://calypso.org/atom.php>

FOAF Vocabulary Specification This document is produced as part of the RDFWeb FOAF project, to provide authoritative documentation of the contents, status and purpose of the RDF/XML vocabulary and document formats known informally as 'FOAF'.

<http://hobbes.com/foaf/>

RDF Site Summary (RSS) 1.0 This documents describes RSS, the web syndication format as an RDF vocabulary.

<http://www.rss.org/>

RDF/XML Syntax Specification This document defines an XML syntax for RDF called RDF/XML in terms of Namespaces in XML, the XML Information Set and XML Base.

<http://www.w3.org/RDF/formats/>

Namespaces in XML This document defines namespaces in XML.

<http://www.w3.org/TR/REC-xml-names/>

XML Information Set This specification provides a set of definitions for use in other specifications that need to refer to the information in an XML document.

<http://www.w3.org/TR/REC-xml-info/>

XSL Transformations (XSLT) Version 1.0 This specification defines the syntax and semantics of XSLT, which is a language for transforming XML documents into other XML documents.

<http://www.w3.org/TR/xslt/>

XML Catalogs Specification OASIS XML Catalogs V 1.1 Specification.

<http://www.oasis-open.org/xml-catalogs/>

Tutorials and Introductory Materials

CSS Reference Pages CSS articles, references and books. (by Eric Meyer)

<http://www.nczonline.com/>

- Zen Garden A demonstration of what can be accomplished visually through CSS-based design.
[\[http://www.sagepub.com\]](http://www.sagepub.com)
- A Technical A close look at the ins-and-outs of XML. (by Norman Walsh)
Introduction
to XML
[\[http://www.xml.com\]](http://www.xml.com)
- DocBook It's the definitive guide to using the DocBook XSL stylesheets. (by Bob Stayton)
XSL: The
Complete
Guide
(Second Edi-
tion)
[\[http://www.docbook.org\]](http://www.docbook.org)
- Atom Syndic- This document focuses on The Atom Standard produced by the IETF AtomPub
ation Format Working Group.
- Introduction
[\[http://www.xml.org\]](http://www.xml.org)
- Describe Edd Dumbill continues the development of a vocabulary for describing open source
open source software projects, presenting a schema for the new vocabulary
projects with [http://usefulinc.com/doap] and example project descriptions.
XML
[\[http://www.docbook.org\]](http://www.docbook.org)
- RSS Tutorial This tutorial explains the features and benefits of a Web format called RSS, and gives
[\[http://www.rss1.com\]](http://www.rss1.com) a brief technical overview of it.
- What Is RDF In this document, Tim Bray and Dan Brickley explain why RDF and metadata are es-
[\[http://www.w3.org/2001/22/rdf/\]](http://www.w3.org/2001/22/rdf/) sential to processing information on the Web.

Glossary

Terms

T I D Y [http://www.w3.org/People/Raggett/tidy]	TIDY is a free utility from W3C to fix HTML/XHTML mistakes automatically and tidy up sloppy editing into nicely layed out markup See Also HyperText Markup Language (XHTML) [http://w3.org/MarkUp/] .
DocBook Website [http://books.oorenix.com/docbook/]	The Website doctype and stylesheets are for making websites (like this one). If you generate a website using Website, consider including the button above in your site (so that your visitors will know you built it using DocBook).
DocBook XSL Stylesheets [http://books.oorenix.com/docbook/]	These are XSL stylesheets for the DocBook DTD and its derivatives (Simplified DocBook, etc.). If you use them to generate PDF/print documents or Web/HTML content, consider including the button above somewhere in your documents (so that your reader will know the documents were built using DocBook). See Also Extensible Stylesheet Language Family (XSL) [http://w3.org/Style/XSL/].
Web Standards [http://www.webstandards.org]	Founded in 1998, The Web Standards Project (WaSP) fights for standards that reduce the cost and complexity of development while increasing the accessibility and long-term viability of any site published on the Web. We work with browser companies, authoring tool makers, and our peers to deliver the true power of standards to this medium.
Apache Ant [http://ant.apache.org]	Ant is a Java-based build tool. In theory, it is kind of like Make, but without Make's wrinkles.
Java (J2SE) [http://java.sun.com/j2se]	Java 2 Standard Edition.
Extensible Markup Language (XML) [http://w3.org/XML/]	Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.
Extensible Stylesheet Language Family (XSL) [http://w3.org/Style/XSL]	XSL is a family of recommendations for defining XML document transformation and presentation.
Cascading Style Sheets [http://w3.org/Style/CSS]	Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colors, spacing) to Web documents.
HyperText Markup Language (XHTML) [http://w3.org/MarkUp/]	XHTML 2 is a general-purpose markup language designed for representing documents for a wide range of purposes across the World Wide Web. To this end it does not attempt to be all things to all people, supplying every possible markup idiom, but to supply a generally useful set of elements.
Resource Description Framework (RDF)	World-Wide Web, specification, data (RDF) A specification being developed in 2000 by the W3C as a foundation for processing meta-data regarding resources on the Internet, including the World-Wide Web. See Also Really Simple Syndication (RSS) .
Really Simple Syndication (RSS)	Really Simple Syndication (RSS) is a lightweight XML format designed for sharing headlines and other Web content.

	See Also Resource Description Framework (RDF) .
World Wide Web (W3C) [http://www.w3.org]	The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential.
Uniform Resource Locator (URL)	Uniform Resource Locator (URL) specified by W3C. See Also World Wide Web (W3C) [http://www.w3.org] .
FireFox	The Mozilla [http://www.mozilla.org] based Web Browser. See Also Web Browser .
Web Browser	A program used to view HTML documents. Some examples include: FireFox [http://www.mozilla.org/products/firefox], Mozilla [http://www.mozilla.org/products/mozilla1.x], Opera [http://www.opera.com], Safari [http://www.apple.com/safari] See Also HyperText Markup Language (XHTML) [http://w3.org/MarkUp/] .

Colophon

This document was prepared using DocBook XSL Stylesheets [<http://docbook.sourceforge.net/projects/xsl/>]. The main font is helvetica, arial and sans-serif. The DocBook XSL was refined and customized by Sina K. Heshmati [<http://www.khakbaz.com/sina/>] which in turn, was derived from the style by Norman Walsh [<http://nwalsh.com/people/ndw/>]. In addition, the visual markup is controlled by a single CSS stylesheet prepared by Javad K. Heshmati [<http://www.khakbaz.com/javad/>]. The final version of this document was produced by running a single Makefile on a Dell Latitude running the Ubuntu [<http://www.ubuntu.com/>] distribution of the GNU/Linux [<http://www.gnu.org/>] operating system.