ChannelPoint Merlot 1.0
User's Guide

5825 Mark Dabling Blvd.
Colorado Springs, CO 80919
Phone 719-260-1232 • Fax 719-482-2999
P4220-00

Contents

Preface

About This Book

ChannelPoint Merlot 1.0™ is a Java-based, open source tool, built by ChannelPoint, Inc., for creating and editing Extensible Markup Language (XML) files and library files of reusable XML. An open source project is defined as free source code made available to the development community at large. The rationale is that a broad group of developers produces a better, more bug-free product in the end.

The ChannelPoint Merlot 1.0 user population is not limited to developers and Java developers. End users who are responsible for building web content can use ChannelPoint Merlot 1.0 as well. It was designed to ensure that users who may not be experienced with XML can easily create and modify XML documents. This document details the instructions to install and start ChannelPoint Merlot 1.0. It also details how to use ChannelPoint Merlot 1.0 to create and edit XML documents and library files of reusable XML code.

This document is not, however, intended for developers who want to modify the functionality of ChannelPoint Merlot 1.0. One of the common features of an open source project such as ChannelPoint Merlot 1.0 is that the source code is made available to the development community. Developers are encouraged to enhance the product and provide information back to the originating company. In the case of ChannelPoint Merlot 1.0, this information is shared with the development community using a mailing list, which can be accessed at the ChannelPoint Merlot 1.0 site: http://www.merlotxml.org. If you want to enhance ChannelPoint Merlot 1.0 source code for your own purposes, it is recommended that you begin by accessing the ChannelPoint Merlot 1.0 web site and click the Development button on the toolbar for information about modifying the source code. Any available instructions to modify ChannelPoint Merlot 1.0 itself can be found at the ChannelPoint Merlot 1.0 site: http://www.merlotxml.org.

Audience

The *ChannelPoint Merlot 1.0 User's Guide* is intended for developers and end users who want to create and modify XML documents and library files of reusable XML. The resulting XML documents can be used within the applications you design. The resulting library files can be distributed among a group of developers to use as needed.

Required Knowledge

Users of this document should have the following knowledge prior to using this document:

- Working knowledge of HTML, XML, XSL, and XSLT standards as published by World Wide Web Consortium (W3C) at:

  http://www.w3c.org

Typographic Conventions

The following typographic conventions are used throughout this manual.

Convention	Description
Bold	Used for headings and figure captions.
Italics	Used for the titles of other books to which we refer. Used for the titles of dialog boxes and windows. Used for values and variables. For example, "The system sets the Type to *Unknown*." Used for terms that are being defined. For example, "The process to define a step in a workflow is called *step configuration*."
Courier	Used for code, for text displayed on the computer screen, for the names of files and directories, and for text you must type. For example: Edit the `config.sys` file.

Convention	Description
Blue text	Used to indicate a hypertext link. Click words in blue to view more information. For example, click entries in the contents to read about a topic, or click a URL to open a web site in your browser. This example is not linked.
Code boxes	Used to set off code samples. Code samples longer than 10 lines are numbered.
Keys	Brackets are used to show keys. To show keys that need to be pressed at the same time, a plus (+) sign is used. For example: Press <Shift + Esc>.
Menu options	A vertical bar (pipe) is used between each menu option. For example: Select File | Print.
Procedures	Step-by-step procedure headings are bold and marked with a diamond-shaped bullet, for example: ◆ **Adding an Index**
• Field names • Tabs • Columns • Buttons • Check boxes	Regular font is generally used for these elements, unless the name is too long and confusing, in which case single quotation marks enclose the name.

Edition History

The following table shows the version of software that this edition of the *ChannelPoint Merlot 1.0 User's Guide* accompanies:

Table P - 1 Book Edition and Product Version

Edition	Accompanies
First (P4220-00)	ChannelPoint Merlot 1.0

Related Documentation

The current documentation, downloads, mailing lists, and plug-ins are made available at the following site. In addition, links to mailing lists and the bug reporting system are also included at this site.

http://www.merlotxml.org

Each of the following items is available at the URL listed above. After accessing the URL, click Documentation for the documents listed here.

- *ChannelPoint Merlot 1.0 User's Guide*
- Frequently Asked Questions
- API Javadocs
- *Setting Up and Building Merlot with Borland/Inprise JBuilder Foundation 3.5* by Gary Cramblitt, August 13, 2000 (revised 24 August 2000)
- *Merlot Plugin Development Guide*, Version 1.0b3 (August 24, 2000)

This site is external to ChannelPoint and does not contain specific DTD files; however, you can obtain DTDs from common web sources and use them to create the XML documents you need. Merlot is a universal tool for creating XML documents.

- Merlot plug-ins can be obtained and downloaded from the following URL. After accessing the URL, click Plugins.

  http://www.merlotxml.org

Contact Information

You can contact the ChannelPoint Merlot development team via the mailing list, which you can access from our web site:

http://www.merlotxml.org

1 What Is ChannelPoint Merlot?

ChannelPoint Merlot 1.0 was specifically designed to speed the development of XML documents. This chapter provides information specific to the files that are opened in ChannelPoint Merlot 1.0 and the files that are the result of ChannelPoint Merlot 1.0. It is important that you review this chapter carefully before developing an XML document or library file of reusable XML using ChannelPoint Merlot 1.0. This chapter will provide the understanding you need to ensure that you can use ChannelPoint Merlot 1.0 effectively.

Introduction to ChannelPoint Merlot

ChannelPoint Merlot 1.0 is an open source tool, built by ChannelPoint, Inc., for creating and editing Extensible Markup Language (XML) files. It is a Java-based XML modeling application intended to make the creation and editing of XML documents easier. ChannelPoint Merlot 1.0 is presented to the user as a graphical user interface (GUI), and it runs on any Java 1.2 (or later) virtual machine. ChannelPoint Merlot 1.0 is used to produce two things: valid XML documents and library files containing reusable XML. ChannelPoint Merlot 1.0 speeds the development of valid XML documents by ensuring that the resulting XML follows the notational and structural rules for XML and that it conforms to a backing DTD file.

The ChannelPoint Merlot 1.0 user population is not limited to developers and Java code developers. End users who design web content pages can use ChannelPoint Merlot 1.0 as well. ChannelPoint Merlot 1.0 was designed to make it easy for users who may not be experts with XML to create and modify XML documents. Because ChannelPoint Merlot 1.0 is presented to the user as a GUI, the user interacts with the interface to add, remove, and rearrange elements within the XML documents. The elements and attributes that are available are determined by the DTD and are presented to the user within the ChannelPoint Merlot 1.0 interface. The interface also provides an easy method for editing the element attribute values. The DTD and actual XML code is hidden from the user.

What Are These Files?

The ChannelPoint Merlot 1.0 product is used to produce two things: valid XML documents and library files of reusable XML. See "How Does ChannelPoint Merlot Work?" on page 12 for a good understanding of how ChannelPoint Merlot 1.0 works. This section will provide detailed explanations of the files that are opened in the ChannelPoint Merlot 1.0 product (called input files for the purpose of this section) and the resulting output files (called output files for the purpose of this section).

The ChannelPoint Merlot 1.0 product uses a DTD file to define and produce an XML document. A full explanation of DTD files or XML documents is beyond the scope of this document, but some of the more common concepts are discussed here to ensure that you will be able to use ChannelPoint Merlot 1.0 effectively.

This section separates the input files, which are files that can be opened in ChannelPoint Merlot 1.0, and the output files, which are the files that result from ChannelPoint Merlot 1.0. Each file is described briefly in a table and then explained more thoroughly in a separate section.

Input Files

ChannelPoint Merlot 1.0 can be used to produce XML documents and library files of reusable XML. These files are the resulting output files from ChannelPoint Merlot 1.0. The components listed in the following table are the input files, which are the

files that are opened in the ChannelPoint Merlot 1.0 product. These input files are used to generate the resulting output files. Each of the input files is explained briefly in the following table and then described in more detail in the following sections.

Component	Description
Document Type Definition (DTD)	The Document Type Definition (DTD) file is the grammar for a markup language, as defined by the designer of the language. It defines the elements, their order, and their attributes to produce a valid XML document. Every valid XML document must have a backing DTD file. ChannelPoint Merlot 1.0 uses the DTD files to produce valid XML documents. More detailed information about the DTDs is included in "Document Type Definition (DTD) Files" on page 4.
ChannelPoint Merlot 1.0 Plug-ins	The plug-ins used in ChannelPoint Merlot 1.0 are optional files used to create or modify XML documents. Two types of plug-ins currently exist: DTD and Action. A DTD plug-in is composed of a DTD and some Java code. An Action plug-in contains only Java code. A DTD plug-in is used to create XML documents, and an Action plug-in is used to define an action to be performed on an XML document. More detailed information about the plug-ins is included in "ChannelPoint Merlot Plug-ins" on page 6.

The following sections contain detailed explanations of each of the input files discussed in this section.

Document Type Definition (DTD) Files

This section will provide some additional information about Document Type Definitions (DTDs) and how they relate to the XML document resulting from ChannelPoint Merlot 1.0. The information provided in this section will also help you understand how the DTD you choose can limit the XML code you can produce. It is not the purpose of this section to provide the information necessary for you to build your own DTD; however, some of the important concepts relative to the most common DTDs will be discussed here for the purpose of education.

The DTD is the grammar for a markup language. The DTD defines the elements, attributes, and entities that make up the contents of a particular markup document. You can build and define your own DTDs or obtain one from another source. One example of a commonly used DTD is the Hypertext Markup Language (HTML) 4.0 DTD published by the World Wide Web Consortium. It defines the rules for an HTML document.

The DTD is associated with an XML document using a DOCTYPE declaration. The DOCTYPE declaration is usually included in the first few lines of the XML document, and looks similar to the following.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

A DOCTYPE declaration can reference an internal or external DTD. An internal DTD resides within the XML document itself, and it applies only to that one document. ChannelPoint Merlot 1.0 cannot currently support an internal DTD. An external DTD is a separate file that provides elements and attributes that can be used in a variety of documents. The external DTD DOCTYPE declaration is the most common.

For every element you want to include in your markup, you must have an element declaration within the DTD. An important aspect of the DTD is that it contains an exact specification of what an element can and cannot contain. An element contains attributes that can have assigned values. Following is an example element declaration called address.

```
<!ELEMENT address (street | city | state | zip_code | country?)>
```

This element declaration indicates that the address element used in the markup can contain only the street, city, state, zip code and country elements. The question mark after country indicates that it is not a required element; it is optional. Element declarations can also specify that an element can contain both text and numbers or can have no content at all (this is an empty element). For example, using the above address element and defining it as allowing both text and numbers, the code would look like the following.

```
<!ELEMENT address (street | city | state | zip_code | country? |
#PCDATA)>*
```

To assign attribute values to your elements, you'll need to ensure attribute declarations are built into the elements. An attribute list links the attributes to a defined element and creates a single attribute named *type*. Attribute lists are built within the elements in a DTD file to ensure the attributes are associated with the element. See this example.

```
<!ATTLIST address CDATA (office | home) "office">
```

The example above provides two types of address: *office* and *home*. The last reference, coded as `"office"`, indicates that the default value for this attribute is *office*.

The DOCTYPE declaration, the elements, and the attributes are all components of a DTD file that can be used to build an XML document in ChannelPoint Merlot 1.0. The information provided in this section is the minimum information you should

know to be able to understand how the DTD you select is used to build an XML document in ChannelPoint Merlot 1.0. This information should also help you understand how the DTD you choose can limit the XML code you can produce.

ChannelPoint Merlot Plug-ins

A ChannelPoint Merlot 1.0 plug-in is an optional and specialized file used in ChannelPoint Merlot 1.0 to accomplish an action with the resulting XML document. There are two types of plug-ins used with ChannelPoint Merlot 1.0: a DTD plug-in and an Action plug-in. A DTD plug-in is used when creating XML documents, and an Action plug-in is used when you want to define an action to perform on an XML document. Each of these types of plug-ins is explained in further detail in this section.

An Action plug-in can act on any XML document. It allows you to perform some action on that document. The XSLT Processor plug-in is an example of an Action plug-in. The intent of this plug-in is to apply an XSL style sheet to an XML document and produce an HTML output. This plug-in works by locating the XML style sheet processing instruction. It then applies the style sheet that is listed and saves the resulting HTML file to a temporary file. That temporary file is then displayed in a browser window for viewing. The XSLT Processor plug-in can be located on the ChannelPoint Merlot 1.0 web site (`http://www.merlotxml.org`).

A DTD plug-in comprises a DTD and some Java code to make it work in ChannelPoint Merlot 1.0. The DTD plug-in declares the root element and the DOCTYPE definition, either public or system identifiers, that appear in the beginning of a resulting XML document. The DTD plug-ins provide a way to add DTD files to ChannelPoint Merlot 1.0's DTD Chooser dialog box. This dialog box appears when you want to open a new or existing XML document and want to associate a DTD with that XML document. The DTD Chooser dialog box looks like the following figure.

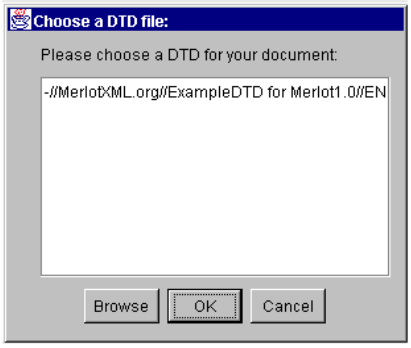

Figure 1 - 1 Choose a DTD file dialog box

The DTD plug-ins are distributed in `.jar` format. The `.jar` file extension stands for Java archive. You do not have to decompress the plug-in `.jar` files for ChannelPoint Merlot 1.0 to be able to access them. The plug-ins provided with the ChannelPoint Merlot 1.0 installation are located in the `\plugins` subdirectory. In addition, you can download additional plug-ins from the ChannelPoint Merlot 1.0 web site. The following table lists the name and description of the DTD plug-in files that are available from the ChannelPoint Merlot 1.0 web site (`http://www.merlotxml.org`). To obtain these files, access the URL and select Plugins from the navigational bar along the top of the page.

Plug-in	Description
`MerlotActionPlugin.jar`	This plug-in allows you to create Action plug-ins in ChannelPoint Merlot 1.0.
`MerlotDTDPlugin.jar`	This plug-in allows you to create DTD plug-ins.
`PluginDocPlugin.jar`	This plug-in allows you to create documentation about the plug-ins you produce in a format that can be easily presented on the ChannelPoint Merlot 1.0 web site.

A DTD plug-in can provide custom editor components for particular elements. A DTD plug-in can also provide custom icons and other custom code to validate the user's input. The ChannelPoint Merlot 1.0 `example.jar` plug-in has a custom editor that appears where a color attribute is assigned to an element in an XML document. This attribute would normally appear in ChannelPoint Merlot 1.0 as a text fiel,d and you could type the hex value of the color you want to apply to that element.

When you use the `example.jar` plug-in, and you click the color text field, a graphical color chooser appears that looks like the following figure.

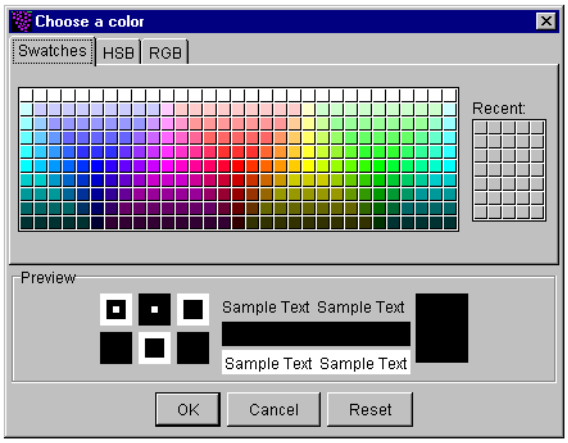

Figure 1 - 2 Choose a color dialog box

You can click the color you want to apply to the attribute. The color you select will automatically be converted to the correct hex notation and saved in the XML document in hex format.

This document will use the `example.jar` plug-in to explain the concepts. This plug-in is installed with the distributed installation files in the `\plugins` subdirectory. In addition, the complete `example.dtd` is included in Appendix A, "Example DTD" for your reference.

Output Files

The ChannelPoint Merlot 1.0 product is used to produce two things: XML documents that can be used in applications, and library files of reusable XML that can be distributed to a group of developers. The files that result from using ChannelPoint Merlot 1.0 are described briefly in the following table and then more thoroughly in the following sections.

Output	Description
Extensible Markup Language (XML)	ChannelPoint Merlot 1.0 is used to produce XML documents. XML is a flexible way to create common formats and share both the format and the data to the web. XML describes web content in terms of what data is being described. It is extensible because the markup symbols are unlimited and self-defining. XML is a formal recommendation from the World Wide Web Consortium (W3C). More detailed information about XML documents is included in "Extensible Markup Language (XML) Documents" on page 9.
Library files	A library file is a collection of commonly reused XML code that is made available to a group of developers as a shortcut to speed up development and to ensure consistency of code. A library includes a way to organize and store the reusable code pieces into sections. ChannelPoint Merlot 1.0 uses an existing XML document and allows you to build a library of commonly used elements from that XML document. See "Library Files" on page 10 for more detailed information and examples of library files.

The following sections contain detailed explanations of each of the output files discussed in this section.

Extensible Markup Language (XML) Documents

The Extensible Markup Language (XML) is a universal format used to describe structured documents and data hierarchically. Structured data is nothing more than some amount of information that can be broken down based on the different roles that information will play. You can use XML to describe any kind of data. XML is extensible, which means you can extend it to make it fit your particular needs.

XML is a markup language used to describe content. It identifies pieces of data using elements. An element is a piece of markup that you use to describe content. Elements comprised of tags, usually a start and end tag. Attributes modify specific instances of the elements with values.

A basic XML document will begin with an XML declaration that identifies it as an XML document. It is usually the first line of an XML document, and every XML document will begin with a similar declaration. One example looks like the following.

```
<?xml version="1.0">
```

A DOCTYPE declaration defines the DTD on which an XML document will be built. See "Document Type Definition (DTD) Files" on page 4 for information specific to the DTD files. The DOCTYPE declaration is usually included in the first few lines of the XML document, and it must precede the elements in the XML document. The DTD specifies the elements, associated attributes, and tags that can be used to describe the markup. The DOCTYPE declaration looks like the following.

```
<!DOCTYPE pages SYSTEM "file:/C:/dev/Arachnid/d1.2/pages.dtd">
```

All of the elements within the document are contained within the root element. The way tags are contained within each other is called the content model. The way tags can be contained within each other, and the attributes that can be assigned, is defined within the DTD.

Library Files

A library file is a collection of commonly used elements that can be reused. Library files are generally distributed to a group of developers as a shortcut to speed up development and to ensure that everyone is using the same code. You will typically want to create a library if you have XML document sections that you frequently reuse in multiple places.

A library includes a way to organize and store the reusable code pieces into sections. A library consists of sections and elements. See the following figure.

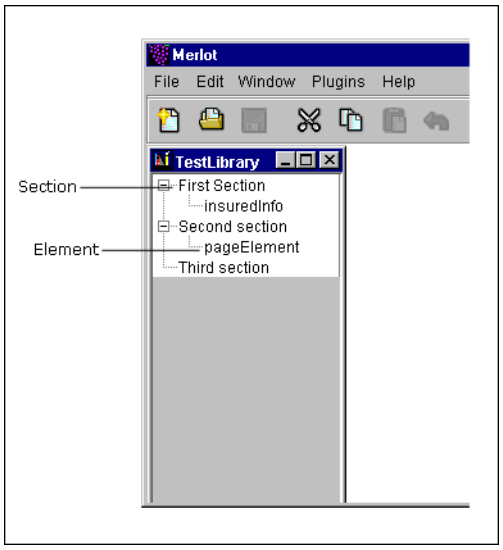

Figure 1 - 3 Library Example

The section is used to organize your elements, and you determine the name of the section and what it contains. The elements will typically contain attributes with assigned values. You will generally populate the library by defining the sections and then dragging elements from any XML document into the sections.

One example of a reusable component is a State choice input field. This type of choice input field typically appears as a drop-down list and is included where an address is required from the end user. When you are developing multiple web applications or form pages in general, it can help to have that common set of code where you can copy and paste it. Another advantage of a library is that when you must modify the common set of reusable code, you can update it in the library and then distribute that change quickly to the development group.

The ChannelPoint Merlot 1.0 product allows you to use any XML document and make a library file containing commonly reused elements. You can assign element attribute values and include the complete element in a library file. The library files are saved with the `.xmllib` extension unless you type a different extension. ChannelPoint Merlot 1.0 requires that file extension to be able to open and edit the library file. With the library file open in ChannelPoint Merlot 1.0, you can drag the elements, with their assigned attribute values, into your XML documents.

How Does ChannelPoint Merlot Work?

ChannelPoint Merlot 1.0 is used to produce two things: valid XML documents and library files containing reusable XML. This section describes how ChannelPoint Merlot 1.0 works to produce these types of documents.

When you use ChannelPoint Merlot 1.0 to build a new XML document, you must first select a DTD file. The DTD file defines the elements, the attributes, and the acceptable attribute values for an XML document. To make a new XML document, or to edit an existing XML document, you must have the associated DTD file. ChannelPoint Merlot 1.0 can open any XML document; however, if you want to edit that document, ChannelPoint Merlot 1.0 must be able to access the DTD. If ChannelPoint Merlot 1.0 cannot locate the assigned DTD for an XML document, an error message will appear, and you will not be able to make changes to the XML document. This makes sense if you consider that ChannelPoint Merlot 1.0 won't be able to determine the elements and attributes without the DTD file. The following figure displays an XML document and the element tree.

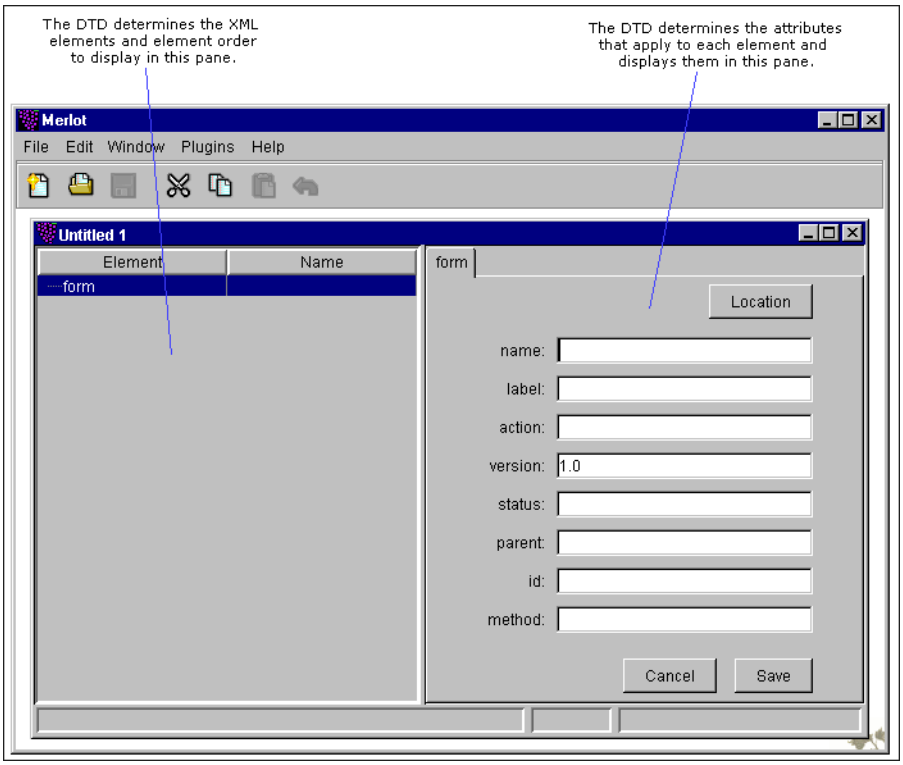

Figure 1 - 4 XML Document in ChannelPoint Merlot 1.0

ChannelPoint Merlot 1.0 eliminates errors that can be inadvertently typed by the XML developer by adding the tags, elements, and attributes as you select them. ChannelPoint Merlot 1.0 ensures the resulting XML documents follow the notational and structural rules for XML. ChannelPoint Merlot 1.0 uses the DTD file to ensure that the attributes are accurately enclosed within the elements. ChannelPoint Merlot 1.0 also ensures that the elements are accurately located within the resulting XML document. The closing tags are included for you, and the resulting XML document will be valid.

ChannelPoint Merlot 1.0 is also used to make library files. A library file is a collection of commonly reused XML code that can be distributed among individual developers. Library files are a simple way to speed up development and to ensure consistency between different developers. Library files are explained in detail in "Library Files" on page 10.

To make a new library file, you need an XML document from which to copy the XML. Using ChannelPoint Merlot 1.0, you will open both the XML document, containing the XML you want to use to build the library, and a new library file. You will then copy elements from the XML document into your library file. See the image below.

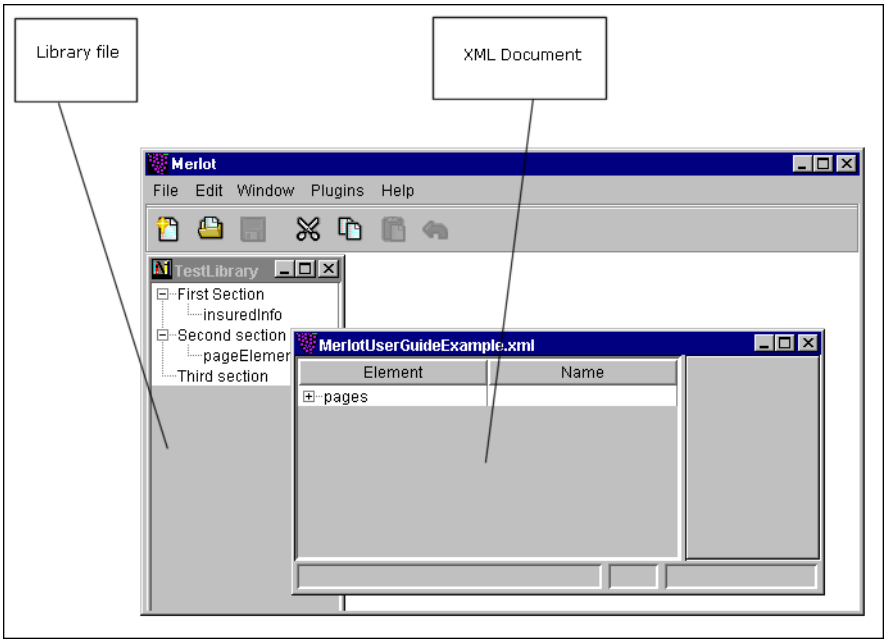

Figure 1 - 5 Library File in ChannelPoint Merlot 1.0

When you are building a library file, you use an existing XML document (or multiple XML documents) and copy the reusable elements with assigned attribute values into the library file. The library file can then be distributed to other developers working on the same project. To use the library file as a source of XML, you will open the library file and copy elements from it into an XML document you are creating. Because you started with valid XML code when you created the library file, you can be sure that the library will contain only valid XML code.

2 Installing ChannelPoint Merlot

This chapter describes the steps to install the ChannelPoint Merlot 1.0 product. This chapter also lists the required hardware and software that must be installed prior to beginning the installation of ChannelPoint Merlot 1.0. Procedures to remove ChannelPoint Merlot 1.0 from your system are provided for you to use when necessary.

Requirements

ChannelPoint Merlot 1.0 requires a Java Virtual Machine (JVM) that implements Sun Microsystem's Java 1.2 Standard Edition. The recommended versions are JDK™ or JRE version 1.2.2 or later. ChannelPoint Merlot 1.0 has been tested to work with Sun Microsystem's JDK 1.2.2 and 1.3 on Microsoft® Windows®, Linux x86, and Solaris™ (x86 and Sparc) platforms. It has also been tested with IBM's JDK 1.3 on Linux. Other platforms with a working JVM version 1.2.2 or later should work with ChannelPoint Merlot 1.0; however, only the ones listed in this paragraph have been fully tested.

Following are the requirements for running ChannelPoint Merlot 1.0:

- Java 1.2.2 or greater Standard Edition
- Processor of 300 Mhz or higher for Intel architecture chips
- 32 MB of RAM is the minimum, 64 MB or higher is recommended

Note: As of the writing of this document, there is no Java 2 platform for the classic Macintosh Operating Systems; however, MacOS X has a Java 2 Virtual Machine. ChannelPoint Merlot 1.0 has not been tested with this product yet.

The instructions for the Java Development Kit (JDK) listed above are not included in this document. The installation instructions in this section assume you have installed the JDK correctly.

A browser is required to obtain the distributed installation files from the ChannelPoint Merlot 1.0 web site. Ensure as well that an internet browser such as Microsoft Internet Explorer or Netscape Navigator® is installed.

The installation instructions in this section assume you have installed WinZip or an appropriate tool to decompress the installation file.

Installation Procedures

The download and installation of ChannelPoint Merlot 1.0 should take only a few minutes, depending upon the speed of your Internet connection. The ChannelPoint Merlot 1.0 installation involves the following steps:

- Downloading the installation files to your computer
- Unpacking the distributed installation file
- Starting ChannelPoint Merlot 1.0

The instructions to download the distributed files will vary slightly depending on the browser you use. The installation instructions are separated into sections below. Use the instructions appropriate to your choice of browser.

Downloading the Installation Files Using Netscape

The first step to install ChannelPoint Merlot 1.0 is to obtain one of the distributed installation files listed below.

- `Merlot-1.0.zip` (2.5 Mb)
- `Merlot-1.0.tar.gz` (2.2 Mb)

You can download the distributed installation files from the ChannelPoint Merlot 1.0 open source site, `http://www.merlotxml.org`. The recommended location to save the distributed ChannelPoint Merlot 1.0 installation file is to `C:\Program Files\ChannelPoint`. The following instructions assume that you have created this subdirectory and that you will use it for this procedure.

Use these instructions to download ChannelPoint Merlot 1.0 using the Netscape browser.

1. Using your browser, navigate to `http://www.merlotxml.org`.

   The ChannelPoint Merlot 1.0 open source web site appears.

2. Click Download in the navigation bar at the top.

   The Merlot downloads page appears listing the binary and source code distribution files in `.zip` (for Windows) and `.tar.gz` (for UNIX) formats. The source code distribution files are for developers who want to modify ChannelPoint Merlot 1.0 source code functionality. The instructions for those processes are not included in this document. The remaining instructions assume you will download and install the correct binary distribution file.

3. Under the Binary Distributions heading, right-click the `Merlot-1.0.zip` file (or the `Merlot-1.0.tar.gz` file) and select Save Link As. . . from the menu.

The *Save As* dialog box appears.

4. Navigate to the `C:\Program Files\ChannelPoint` subdirectory and click Save.

   Netscape saves the distributed file to this subdirectory.

The download is complete. Continue with "Unpacking the Distributed Files" on page 19 to complete the installation and start ChannelPoint Merlot 1.0.

Downloading the Installation Files Using Internet Explorer

The first step to install ChannelPoint Merlot 1.0 is to obtain one of the distributed installation files listed below.

- `Merlot-1.0.zip` (2.5 Mb)
- `Merlot-1.0.tar.gz` (2.2 Mb)

You can download the distributed installation files from the ChannelPoint Merlot 1.0 open source site, `http://www.merlotxml.org`. The recommended location to save the distributed ChannelPoint Merlot 1.0 installation file is to `C:\Program Files\ChannelPoint`. The following instructions assume that you have created this subdirectory and that you will use it for this procedure.

Use these instructions to download ChannelPoint Merlot 1.0 using the Internet Explorer browser.

1. Using your browser, navigate to `http://www.merlotxml.org`.

   The ChannelPoint Merlot 1.0 open source web site appears.

2. Click Download in the navigation bar at the top.

   The Merlot downloads page appears listing the binary and source code distribution files in `.zip` (for Windows) and `.tar.gz` (for UNIX) formats. The source code distribution files are for developers who want to modify ChannelPoint Merlot 1.0 source code functionality. The instructions for those processes are not included in this document. The remaining instructions assume you will download and install the correct binary distribution file.

3. Under the Binary Distributions heading, click the `Merlot-1.0.zip` (or `Merlot-1.0.tar.gz` for UNIX systems) file to download it.

   A screen similar to the following may appear depending on your browser settings.

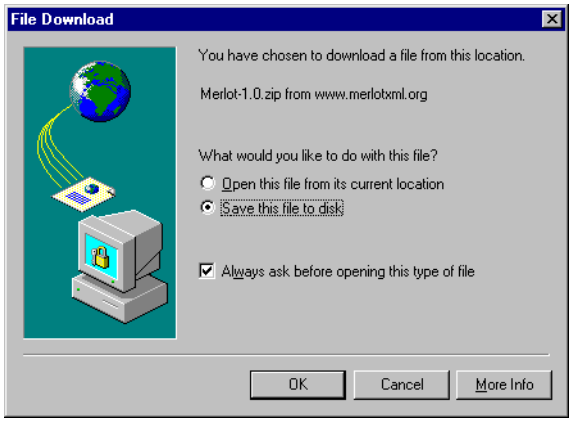

Figure 2 - 1 Downloading Using Internet Explorer

4. Follow the on-screen instructions to save the distributed file to the `C:\Program Files\ChannelPoint` subdirectory.

The download is complete. Continue with the next section to complete the installation and start ChannelPoint Merlot 1.0.

Unpacking the Distributed Files

After you download the distributed ChannelPoint Merlot 1.0 files, it is recommended that you unpack or unzip the files to the `C:\Program Files\ChannelPoint` subdirectory. This is the recommended location. The following instructions assume that you will use this subdirectory. The following instructions also assume you are using WinZip.

Use the following instructions to unpack the ChannelPoint Merlot 1.0 files.

1. Navigate to the `C:\Program Files\ChannelPoint` subdirectory and locate the `Merlot-1.0.zip` (or `Merlot-1.0.tar.gz` for UNIX systems) file.

   The WinZip window appears with the contents of the file listed.

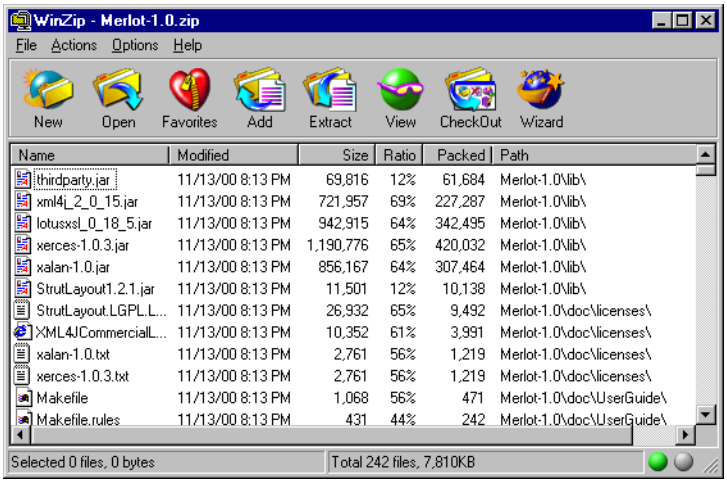

Figure 2 - 2 Unpack Installation Files with WinZip

2. Click Extract.

   The *Extract* dialog box appears.

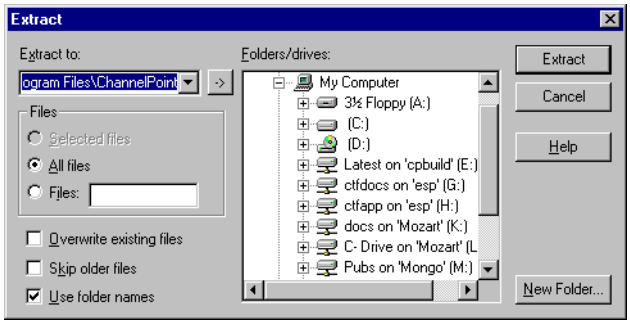

Figure 2 - 3 WinZip Extract Dialog

3. Select the `C:\Program Files\ChannelPoint` subdirectory in the *Extract* dialog box. This is where the installed ChannelPoint Merlot 1.0 files will reside.
4. Under Files, select All Files.
5. Select the Use folder names check box.
6. Click Extract.

   The extract process will unpack the ChannelPoint Merlot 1.0 files and save everything to the `C:\Program Files\ChannelPoint` subdirectory.

7. When finished, close the WinZip window.
8. Using Windows Explorer, navigate to the `C:\Program Files\ChannelPoint` subdirectory and review the directory structure similar to the following:

```
Merlot-1.0
  \doc
  \dtd
  \lib
  \plugins
  CHANGES.txt
  CREDITS.txt
  LICENSE.txt
  Merlot.bat
  Merlot.sh
  Merlot-1.0.jar
  README.txt
```

The installation is complete. The following table provides a quick description of the subdirectories installed with ChannelPoint Merlot 1.0.

Subdirectory	Description
`\doc`	This subdirectory contains the User, License, and API documentation relative to ChannelPoint Merlot 1.0. It is recommended that you do not save files to this subdirectory.
`\dtd`	This subdirectory contains the `library.dtd` file, which is for creating library files. It is recommended that you do not save files to this subdirectory.
`\lib`	This subdirectory contains third-party files crucial to ChannelPoint Merlot 1.0. Do not save files or make changes to this subdirectory.
`\plugins`	This subdirectory contains the plug-ins that are included with the distributed installation files. It is recommended that you save your DTD plug-in files to this subdirectory.

The following table provides a quick description of the contents of the text files that are installed with ChannelPoint Merlot 1.0.

Text File	Description
`CHANGES.txt`	This file contains brief informational notes about changes made to ChannelPoint Merlot 1.0, including the dates those changes were made.
`CREDITS.txt`	This file contains a list of the people and organizations who have contributed to the ChannelPoint Merlot 1.0 open source project.
`LICENSE.txt`	This product is distributed under an Apache-style license. Please read this file before you begin using ChannelPoint Merlot 1.0.
`README.txt`	This file contains very basic instructions to install, start, and run ChannelPoint Merlot 1.0.

See the next section for instructions to start ChannelPoint Merlot 1.0.

Starting ChannelPoint Merlot

The method you use to start ChannelPoint Merlot 1.0 will depend upon the operating system you have installed. For example, if you have the Java Development Kit 1.2 (or greater) installed and running, then you can double-click the `Merlot-1.0.jar` file to start ChannelPoint Merlot 1.0. Similarly, you will double-click the `Merlot-1.0.bat` file for a DOS/Windows installation and the `Merlot-1.0.sh` file for a UNIX installation. Each of these files is a simple start-up script containing the java command line for starting ChannelPoint Merlot 1.0.

The following instructions assume the Java Development Kit installation option explained above. Substitute the correct start-up script in the instructions below as needed.

1. To start ChannelPoint Merlot 1.0, double-click the `Merlot-1.0.jar` file.

   The initial ChannelPoint Merlot 1.0 user interface opens. It looks like the following image.

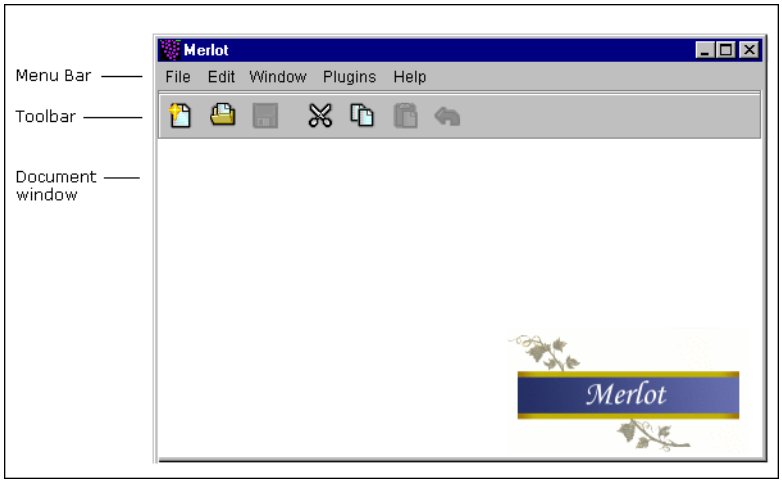

Figure 2 - 4 Initial ChannelPoint Merlot 1.0 Screen

Initially, no DTD is selected, so there are no elements or attributes with which to work. See Chapter 4, "Using ChannelPoint Merlot" for information about using this product. The image above is intended to give you an idea of the basic user interface.

Uninstalling ChannelPoint Merlot

You may find it necessary to uninstall ChannelPoint Merlot 1.0 at some time, either to save disk space or simply to update the installation files. The procedure to uninstall ChannelPoint Merlot 1.0 from your computer is very simple. ChannelPoint Merlot 1.0 doesn't add any special files or settings to your computer when it is installed, so it doesn't need a special uninstall shield to remove it.

Note: The uninstall instructions assume you used the recommended subdirectory to install ChannelPoint Merlot 1.0. If this is not how you installed ChannelPoint Merlot 1.0, you will have to adjust the following directions accordingly.

Use the following instructions to remove ChannelPoint Merlot 1.0 from your computer.

1. Locate and expand the `C:\Program Files\ChannelPoint` subdirectory on your computer.
2. Delete the Merlot subdirectory and all files within it.

The uninstallation is complete.

3

Getting Started

This chapter outlines what you need to know before you begin building an XML document in Merlot. The online reference materials, including the URLs, are listed in this chapter. Merlot's limitations and common coding standards are also detailed in this chapter.

Before You Begin

Before you begin coding an XML document with Merlot, it is important to understand the components that will go into your XML document. These components are detailed in "What Are These Files?" on page 3. Please ensure you have thoroughly reviewed that section.

Of course, before you can begin coding an XML document, you need a good sample of the different data you will need to describe with your markup file, and an adequate understanding of how the data fits together in a document structure. This is an activity in data exploration, and your markup and solution will be better for it in the end.

You should also be able to locate the necessary reference materials you may need. See the following section.

Reference Materials

The ChannelPoint Merlot 1.0 product is an open source project. All current documentation, downloads, bug reporting tools, mailing lists, and plug-ins are available at the following web site.

`http://www.merlotxml.org`

Information specific to the World Wide Web Consortium's XML, XSL, and XSLT standards can be found at the following URL.

`http://www.w3.org/`

Coding Standards

ChannelPoint follows the coding conventions published by Sun Microsystems, Inc., available at the following URL: `http://java.sun.com/docs/codeconv`.

In addition to the standards published by Sun Microsystems, ChannelPoint recommends you use the following standards with XML documents created using Merlot.

- Comments should be included to provide information not readily available within the markup itself.
- The terms *tag* and *element* are both used when discussing markup. These terms are often used interchangeably, but they are actually two separate entities. An element is a piece of markup that you use to describe content. Elements are

composed of tags, usually a start and end tag. Throughout this document, these terms will be used as described here.

- ChannelPoint Merlot 1.0 ensures that the resulting XML document contains fully conforming XML, which means the documents follow the notational and structural rules for XML. These are handled by ChannelPoint Merlot 1.0:
  - Include the XML document type declaration at the start of the document.
  - Include strictly hierarchical tags (nested elements should be completely enclosed within other elements).
  - Include close tags for all tags.
  - include a slash before the greater-than sign within every empty element (such as />).
  - Include attributes and values within quotes.
  - Include the correct usage of character entities for special characters (<, >, and ").

Merlot's Limitations

ChannelPoint Merlot 1.0 is an open source project, which means that a diverse group of developers is using the source code and providing improvements to the project. This also means that the current stage of development will have some limitations, and ChannelPoint Merlot 1.0 has a few limitations. This section briefly lists the limitations along with a description.

- ChannelPoint Merlot 1.0 cannot currently support an internal DTD. This means that the resulting XML document cannot contain the DTD within the XML document. The DTD must be a separate file.
- ChannelPoint Merlot 1.0 does not yet support namespaces or XML schemas.
- ChannelPoint Merlot 1.0 supports most of the XML constructs, except `CDATA`. It also does not yet support `ID` or `IDREF` as an attribute type.
- ChannelPoint Merlot 1.0 does not yet allow you to edit the resulting XML code. For example, once you have created a document, you cannot change the DOCTYPE or the root element using Merlot. If you need to make a change to the root element, for example, you will have to open the XML document in some available text editor and modify it from there.

Refer to the ChannelPoint Merlot 1.0 mailing lists at the Merlot web site (`http://www.merlotxml.org`) for any available information about these limitations.

4 Using ChannelPoint Merlot

This chapter provides the detailed instructions to use ChannelPoint Merlot 1.0 to build XML documents and library files. This chapter assumes you have installed the ChannelPoint Merlot 1.0 product according to the instructions in Chapter 2, "Installing ChannelPoint Merlot." If you are using ChannelPoint Merlot 1.0 to build an XML document, ensure that you have obtained a valid DTD document. If you are using ChannelPoint Merlot 1.0 to build a library, ensure that you have an XML document from which to obtain the library elements.

Overview

This chapter provides the instructions to create a new XML document and a new library of reusable elements. This chapter also provides the instructions to modify an existing XML document or library file, when necessary. This chapter assumes that you have thoroughly reviewed the information in Chapter 1, "What Is ChannelPoint Merlot?".

Without a valid DTD, ChannelPoint Merlot 1.0 cannot present the menu of elements that you can add to your XML document. You can view and edit the element attributes without the associated DTD, but you will not be able to add anything new to the XML document without the DTD. ChannelPoint Merlot 1.0 presents the XML as an element tree, which is just a graphical representation of the Document Object Model (DOM).

Throughout this document, the `merlot-example.jar` plug-in will be used to demonstrate the elements and attributes. The ChannelPoint Merlot 1.0 plug-ins are explained in "ChannelPoint Merlot Plug-ins" on page 6. The `merlot-example.jar` plug-in is included with the ChannelPoint Merlot 1.0 installation as a generic sample. The full content of the `merlot-example.dtd`, included within the `merlot-example.jar` plug-in, is included in Appendix A, "Example DTD" for your reference. In the figure below, you can see that the `merlot-example.jar` was used to add a few elements and attributes to a temporary XML document.

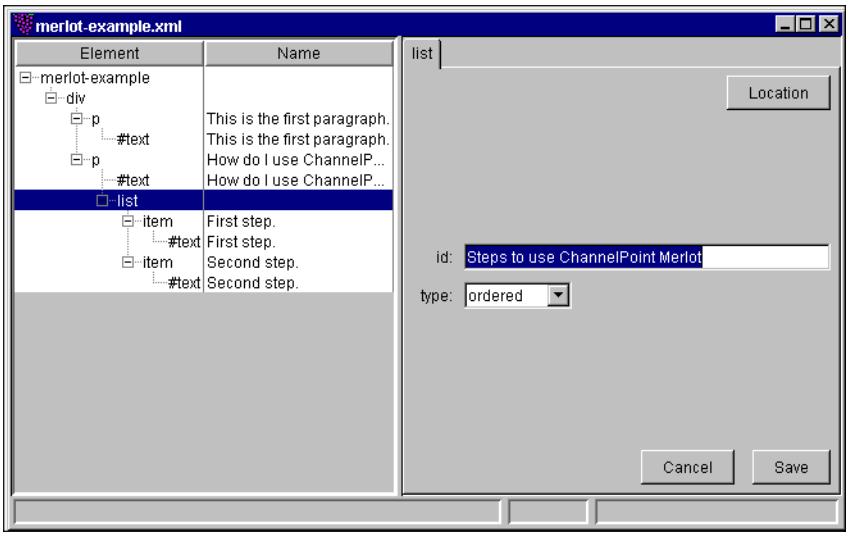

Figure 4 - 1 The Element Tree and Attribute Pane

The attribute pane appears to the right of the element tree pane, as shown in Figure 4 - 1 above. The attribute pane displays the attributes that can be assigned to this element, per the DTD. The attribute values are typed into fields, as shown above, and then applied to the associated element.

The instructions in this chapter will be somewhat limited in that the instructions cannot take into account all the possible variations that can be built into a DTD. DTD files are by definition highly specialized files for a specific purpose. For the purpose of this document, the `merlot-example.jar` plug-in will be used to provide adequate examples of the concepts and the instructions detailed in the following sections. You should realize that your DTD and the resulting XML document will be very different from those demonstrated in this document.

Creating a New XML Document

This section details the steps involved in creating a new XML document from scratch using a sample DTD. See Chapter 2, "Installing ChannelPoint Merlot" for instructions to install and start ChannelPoint Merlot 1.0. If you have an existing XML document that you want to modify, skip to "Modifying Documents" on page 46 for information and instructions.

The instructions in this section assume that you have a valid DTD file. Without a valid DTD, ChannelPoint Merlot 1.0 cannot present the menu of elements to add to your XML document. You can view and edit documents without the associated DTD, but you will not be able to add anything new to the XML document without the DTD.

Note: If your DTD is contained within a plug-in, ensure that you saved the plug-in to the `/plugins` subdirectory.

Use these procedures to create a new XML document with a selected DTD.

1. Start ChannelPoint Merlot 1.0.

   The initial ChannelPoint Merlot 1.0 user interface appears without a DTD selected. To create a new XML document, you must first select the DTD on which this document will be built.

2. Select File | New File.

   The dialog box to choose a DTD appears.

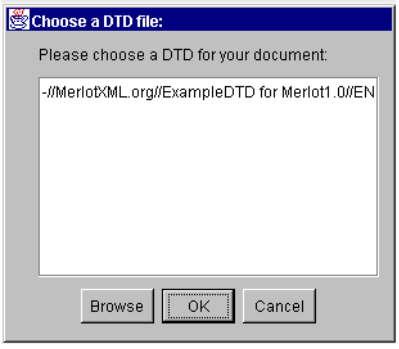

Figure 4 - 2 Choose a DTD file dialog box

Note: If you do not have a DTD, an error message appears asking you to locate a DTD. It is not within the scope of this document to provide the instructions to obtain a DTD.

3. Select a DTD from the list and click OK.

   After choosing the DTD on which to base your new XML document, ChannelPoint Merlot 1.0 presents the document editing window with the root form element selected. See the following figure as an example. The element tree can be expanded and collapsed to view the elements available within each parent element.

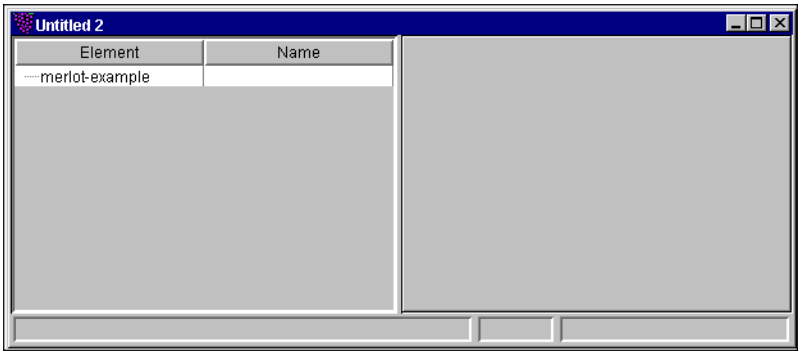

Figure 4 - 3 DTD Plug-in Selected

Once you have started the initial XML document, you are ready to start adding new elements to it. Continue with the next section.

◆ Adding Elements

ChannelPoint Merlot 1.0 builds the element tree menus using the DTD at run-time. This is to ensure that you can create XML documents that are valid according to the DTD. In addition, you can only access the elements that are available in the DTD. The element tree can be expanded and collapsed to view the elements available within each parent element. Click each element node to expand that node. Once expanded, you can click the element node to collapse it.

Use these instructions to add elements (or nodes) to your XML document.

1. To add new elements to your document, right-click the element in the element tree that you want to add.
2. Select Add from the menu.

The menu appears similar to the following image.

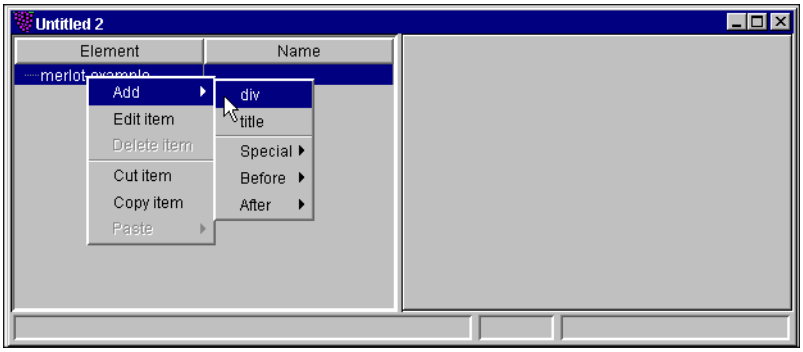

Figure 4 - 4 Element Menus

ChannelPoint Merlot 1.0 displays only the valid elements as determined by the selected DTD file. When you add an element into another, ChannelPoint Merlot 1.0 places it at the end of the parent element. You can use the Add | Before and the Add | After menu options to add an element precisely where you want it.

3. To add an element before the selected element, select Before from the menu.

   The element's attribute(s) appear in the attribute pane, which appears to the right of the element tree and looks like the following figure.

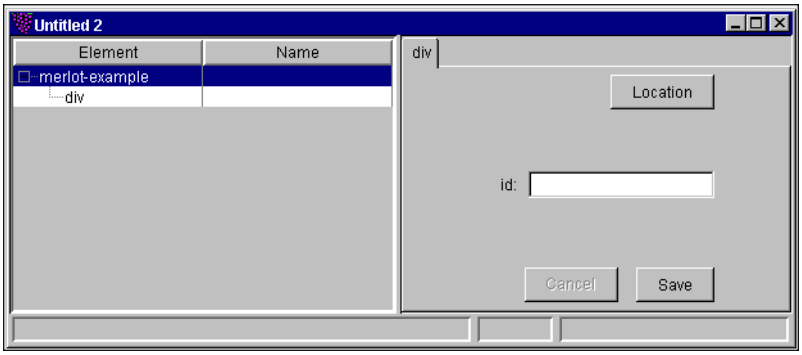

Figure 4 - 5 Element Tree with Attribute Pane

The element you choose determines the attribute fields that appear for that element.

Note: The Location button is used to locate and highlight the element's place within the element tree. This function is particularly useful when you have a large XML document to edit.

4. Type the element attribute values in the fields in the attribute pane.

Note: The required attribute flag icon (not shown above) appears next to attribute fields that are required. The icon designates those attributes that must have assigned values for the element to be complete. If you click Save before assigning a value to the required attribute, you will receive an error message.

5. When you are finished, click Save.

   Notice how the element tree changes to include the new element you just added.

6. To add a comment to the XML document, right-click the element where you want the comment to be.
7. Select Add | Special | Comment from the menu.
8. The comment attribute pane appears, as shown here.

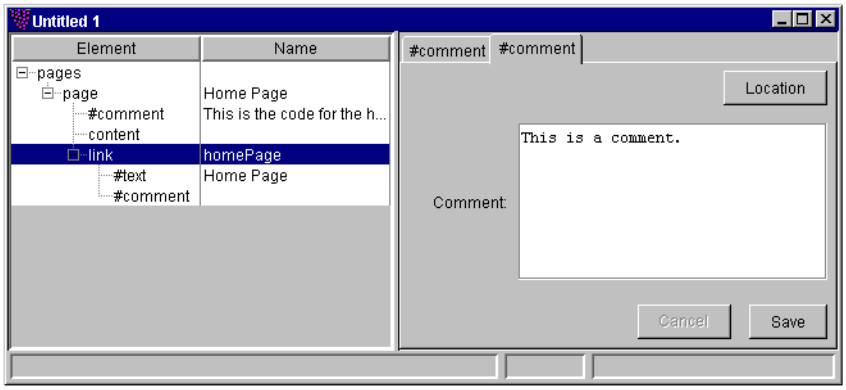

Figure 4 - 6 Attribute Pane for Comment Element

In the image above, notice that the Cancel button is unavailable. When you are creating new elements in an XML document, the Cancel button is unnecessary. When you are editing an existing element, the Cancel button is available for you to cancel your procedure. Clicking Cancel leaves the original XML code intact.

9. Type the text of your comment into the Comment field.
10. When you are finished, click Save.

Repeat the instructions in this section to continue adding elements to your XML document, as necessary.

Continue with the next section for information about how to edit the attribute values for an element.

◆ Editing the Element Attributes

When you add a new element to your XML document, you can double-click that element to view the element's attribute pane. The attribute pane lists the current attributes and the assigned values. The attribute pane is available to you to make changes, as necessary. This section describes how to edit an existing element's attribute values.

Use the instructions in this section to edit an existing element's attributes.

1. Locate the element you want to edit.
2. Right-click the element and select Edit Item from the menu.

   - OR -

   Double-click that element.

   The attribute pane appears with the current values you typed. It will look similar to the following figure.

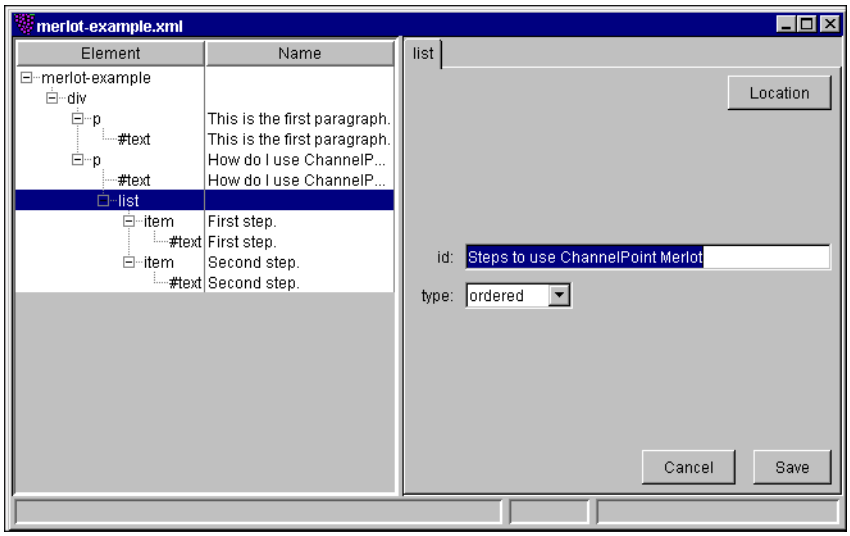

Figure 4 - 7 Editing Attribute Values

Note: When you are modifying existing elements, the Cancel button appears to provide a way for you to stop your procedure without altering the XML document. When you are adding new elements, the Cancel button is not available.

3. Click in the attribute fields and type the new values.

4. When you are finished, click Save.

   The XML document and the element tree are updated automatically.

See the next section to move elements within an XML document.

◆ Moving Elements

When you move an element in your XML document to a new location, you are basically restructuring the XML document. If the element's destination location does not allow the item being inserted, you will receive an error message.

This section describes how to restructure your XML document using ChannelPoint Merlot 1.0. Two methods for moving elements are available with ChannelPoint Merlot 1.0, and each is explained in the following sections.

Using a Drag-and-Drop Operation

Use these instructions to move elements in the element tree using a drag-and-drop operation.

1. Locate the element you want to move.
2. Select the element and drag it to its new location.

   Lines appear as you drag the element to help you see the element tree location. If the destination element doesn't allow the element you are dragging, you will see an error message similar to the following.

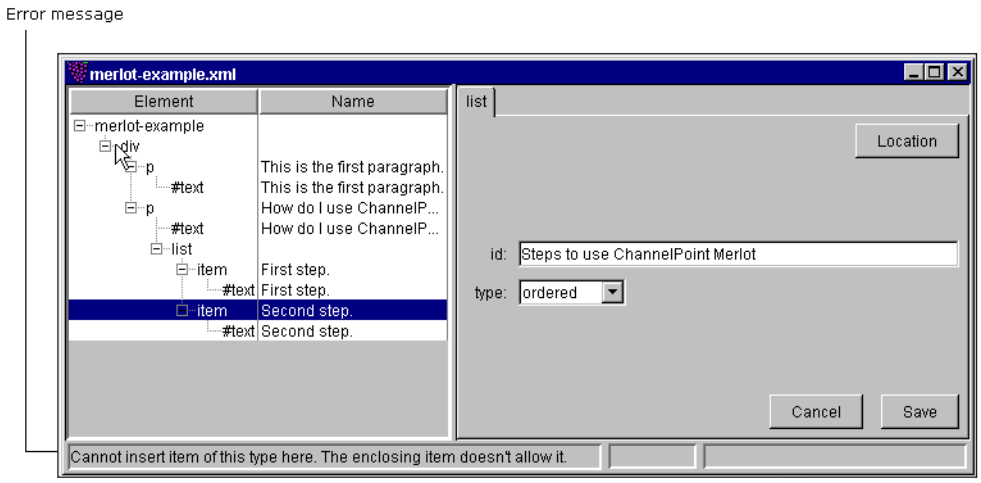

Figure 4 - 8 Drag-and-Drop error

If the element can be moved to the new location per the rules of the DTD, that element will be moved and you will see the element in its new location immediately.

Using Copy and Paste

Use these instructions to move elements within the element tree using the copy and paste Options.

1. Locate and select the element you want to move.
2. Select Edit | Copy to copy that element to the Clipboard.

   - OR -

   Click the Copy button on the toolbar to copy that element to the clipboard.
3. Locate and select the destination element in the element tree.
4. Select Edit | Paste into to copy the element into the element tree.

   - OR -

   Click the Paste button on the toolbar to copy the element into the element tree.

Note: You can also select Edit | Paste before or Edit | Paste after to paste the element before or after the selected element in the element tree.

The item is copied into the element tree. If your intent was to copy the element to the destination location, you are finished. If your intent is to relocate that element, return to the originating element and delete that element using the instructions in the next section.

Continue with the next section for additional instructions.

◆ Deleting Elements

When you delete an element from the element tree, you are essentially removing that element from your XML document. You are not deleting the element from the backing DTD file. The instructions to modify the DTD are not included in this document.

Use these instructions to delete elements from the element tree.

1. Locate and select the element you want to remove in the element tree.
2. Right-click the element and select Delete Item from the menu.

   - OR -

Click the Cut button on the toolbar.

The item is deleted from the element tree.

Note: You will not receive a prompt to confirm the deletion; however, if you mistakenly delete an element, you can undo that change. The first item in the Edit menu is Undo, and the title of this menu item will toggle to the last action performed. For example, if you add an element, and then you want to undo your changes, the first item in the Edit menu will read *Undo Add*. You can also use the Undo button on the tool bar or the <Ctrl + Z> key sequence.

◆ Saving the New XML Document

You must save your XML document before you can use it within your web applications. The save feature of ChannelPoint Merlot 1.0 is similar to other interfaces, and it is very easy to perform.

1. Select File | Save As to save the XML document.

   - OR -

   Click the Save button on the toolbar.

   The *Save As* dialog box appears.

2. Select the subdirectory location where you want the XML document to be saved.
3. Type the file name you want to assign to this XML document in the File name field.
4. Click Save.

The file is saved. You can now use the XML document in your applications.

Creating a Library

Libraries act as an easy way to distribute reusable elements. A library is a set of commonly used elements or code that is stored in a single location for easy access by a number of developers. Libraries are often used for template content. Libraries can be distributed among multiple developers or made available using a shared network drive. You can create a library with ChannelPoint Merlot 1.0, and then use the elements in that library to populate an element tree for your XML documents. The library files created with ChannelPoint Merlot 1.0 are saved with the `.xmllib` extension. If you want to use ChannelPoint Merlot 1.0 to edit the library file later, it must have this file extension.

This section includes the instructions to create a new library and copy elements into that library.

◆ Creating a New Library

A library file is used to store commonly used elements for later reuse. Use these instructions to create a new library of reusable elements.

1. Start ChannelPoint Merlot 1.0.
2. Select File | New Library.

   The *Name of library* dialog box appears, as shown here.

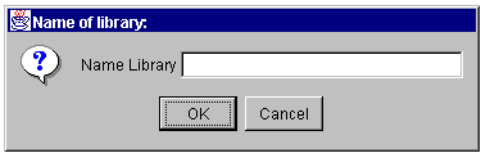

Figure 4 - 9 Name of library dialog box

3. Type the name of the library.
4. Click OK.

   The *Create new library file* dialog box appears.
5. Select the subdirectory where you want to store the library.
6. Type the library file name in the File name field.

Warning! Keep the automatic `.xmllib` extension!

7. Click Save.

   The library is saved with the name you specified, and the ChannelPoint Merlot 1.0 screen changes to look similar to the following figure.

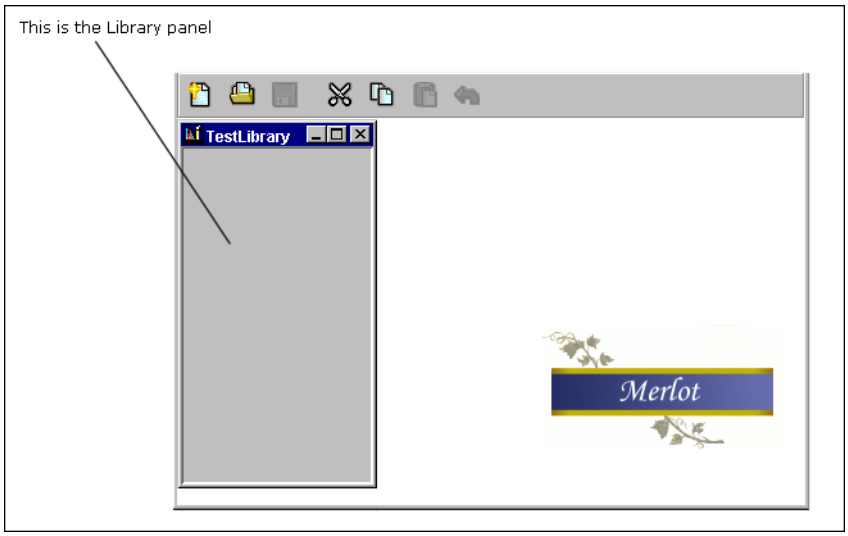

Figure 4 - 10 New Library Is Saved

When the new library is first saved, the library panel is empty. You must add sections to it before you can add any elements to the library. Continue with the next section to add sections to the library.

◆ Adding Sections to a Library

You must add sections to the library before you can add elements. The sections are a way to organize the elements into categories for later reuse. Use these instructions to add sections to the new library.

1. Right-click in the library panel and select New Section.

   The *Name of new section* dialog box appears, as shown here.

Figure 4 - 11 Name of new section dialog box

   The name indicates a category of reusable elements and not the name of the elements. It is within this category that you will save your elements for later reuse and distribution.

2. Type the name of the library section.
3. Click OK.

Note: The library is automatically saved when you quit ChannelPoint Merlot 1.0 or close the library file.

When finished, continue with the next section to add elements to the library.

◆ Adding Elements to a Library

Ensure that the library has been created and saved using the instructions in "Creating a New Library" on page 40. Use the instructions in this section to add elements to the library you just created.

1. Open the library file (with an `.xmllib` extention) by selecting File | Open Library.

   The library appears in the ChannelPoint Merlot 1.0 window.

2. Select File | Open File to open the XML document containing the elements you want to save to the library file.

   The XML document (with a selected backing DTD file) appears in the ChannelPoint Merlot 1.0 window. See the figure for an example.

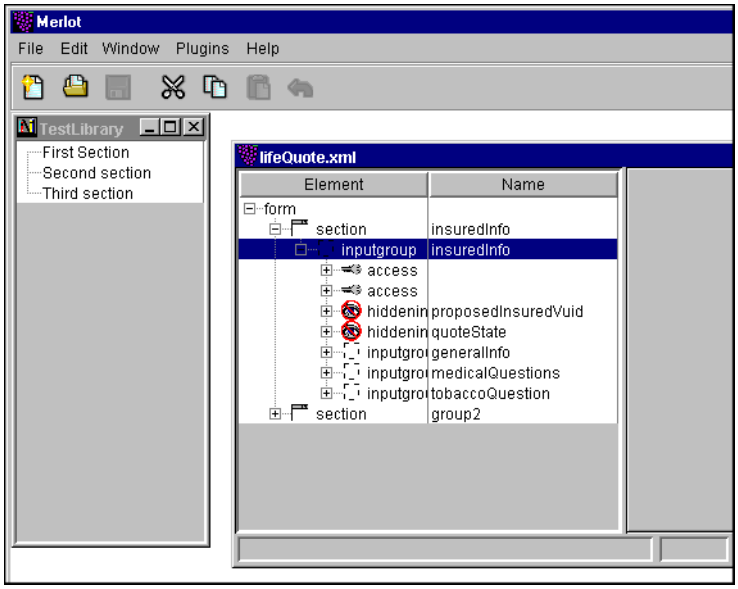

Figure 4 - 12 Library and XML Document Open

3. Locate the elements in the XML document that you want to save to the library.
4. Click the element and drag it to the appropriate section in the library.

   - OR -

   Right-click the element and select Copy Item. Then, right-click the library section where you want to drop the item and select Paste from the menu.

   The *Name of new item* dialog box appears as shown below.

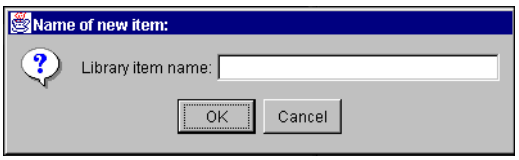

Figure 4 - 13 Library item name dialog box

5. Type the name of this element and click OK.

6. The element is copied into the library section with the name you typed applied.
7. Repeat steps 3 through step 6 to add the additional elements into the library sections.

Note: The library is automatically saved when you quit ChannelPoint Merlot 1.0 or close the library file.

See the following section for instructions on using the elements contained in a library.

◆ Using Elements From a Library

To use the elements in a library file, you must first create a library or open an existing one. See "Creating a New Library" on page 40 to create a library; see "Adding Sections to a Library" on page 41 to add sections to your library; and see "Adding Elements to a Library" on page 42 to include elements in your library file.

Once you have a library with the elements you want to use, you can reuse these elements by copying them into your XML documents. When the element is copied into a new XML document, you must ensure that the DTD and DOCTYPE definition are correct for that element. If you copy more than one element from a library, you should be aware that ChannelPoint Merlot 1.0 checks only the first element for validity with the destination XML document. ChannelPoint Merlot 1.0 does not enforce that the document an element came from and the document it is being copied into have the same DTD or DOCTYPE definition.

Use the instructions in this section to take advantage of the reusable elements stored in a library.

1. Open a library by selecting File | Open Library.

   The *Open File* dialog box appears.

2. Select the library file you want to use and click Open.

   The library file appears in the ChannelPoint Merlot window.

3. Open the XML document you want to copy the elements into by selecting File | Open File.

   - OR -

   Open a new XML document by selecting File | New File, and then select the backing DTD file.

   The XML document also appears in the ChannelPoint Merlot window.

4. Locate the element you want to copy into the destination XML document.

   When you are adding elements from a library to an XML document, there are multiple ways to get the elements into the destination document. Use one of the following options.

   - Right-click the element in the library pane, and select Copy from the menu. Then, right-click the correct location in the destination XML document, and select the correct Paste option from the menu.
   - Select the element in the library pane and, while holding the mouse button down, drag the element to the correct location in the destination XML document.
   - Select the element in the library pane and select Edit | Copy from the menu. Then, select the correct location in the destination XML document, and select Edit | Paste from the menu.

   If the element you are copying isn't allowed by the destination XML document's DTD, you will see an error message along the bottom of the XML document's window. This means that the element you are trying to copy isn't allowed within the enclosing element of the destination XML document.

5. Repeat step 4 until you have added all the library elements you need to the destination XML document.

Note: Changes to the library file are automatically saved. The library is also automatically saved when you quit ChannelPoint Merlot 1.0 or close the library file.

The library of elements will probably contain only a few of the elements you will need for your final XML document. In most cases, after you have copied the reusable elements from the library, you will still have to add additional elements to the final XML document. See "Adding Elements" on page 33 for instructions to add more elements to your final XML document.

Modifying Documents

The ChannelPoint Merlot 1.0 product is not limited to creating new XML documents and library files. ChannelPoint Merlot 1.0 can be used to make changes to any XML document, even those not created using ChannelPoint Merlot 1.0. In addition, ChannelPoint Merlot 1.0 can be used to modify a library file.

This section provides the basic instructions to make changes to an existing XML document or library file. These instructions reference other sections of this chapter for the specific modifications you can make.

◆ Making Changes to an XML Document

When you make changes to an existing XML document, you must be aware of where the document is used within your web applications. This section includes only the basic instructions, and it cannot take into account all the problematic variables that can happen when changes are made to an XML document that is in use within a web application.

Use the instructions in this section to open an existing XML document, make changes to that document, and then save the document.

1. Start ChannelPoint Merlot 1.0.
2. Open an XML document by selecting File | Open File.

   The *Open File* dialog box appears.
3. Locate and select the XML document, then click OK.

   The *Choose a DTD file* dialog box appears, as shown below.

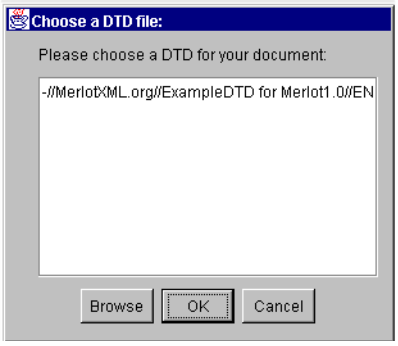

Figure 4 - 14 Choose a DTD file dialog box

4. Select the DTD on which this XML document is built.
5. Click OK.

   The XML document appears in the ChannelPoint Merlot 1.0 window.
6. After the XML document is open, you can make any of the following changes:
   - See "Adding Elements" on page 33 to add new elements to the XML document.
   - See "Editing the Element Attributes" on page 36 to modify the element attribute values.
   - See "Moving Elements" on page 37 to move elements to a new location within the XML document.
   - See "Deleting Elements" on page 38 to remove elements from the XML document.
7. When you are finished, Select File | Save to save the XML document

   - OR -

   Click the Save button on the toolbar.

The file is saved with the same name and subdirectory location. See the next section for instructions to make changes to a library file.

◆ Making Changes to a Library File

When you make changes to an existing library file, you must be aware that the elements included in that file may have been transferred into a functioning XML document. The changes you make to the library file will not immediately impact other developers, but the changes will impact when developers try to copy elements from the library into the existing XML documents. This section includes the basic instructions to make changes to a library file.

Use the instructions in this section to open an existing library file, make changes to that library file, and then save the library file.

1. Start ChannelPoint Merlot 1.0.
2. Open a library file (with an `.xmllib` extension) by selecting File | Open Library.

   The *Open File* dialog box appears.
3. Locate and select the library file, then click OK.
4. The library file click the Save button on the toolbar.

   in the ChannelPoint Merlot 1.0 window.

5. Open the XML document you want to copy the elements into by selecting File | Open File.

   - OR -

   Open a new XML document by selecting File | New File, and then select the backing DTD file.

   The XML document also appears in the ChannelPoint Merlot 1.0 window.

6. After these files are open, you can make the following changes to the library file:

   - See "Adding Sections to a Library" on page 41 to add new sections (or categories) to a library file.
   - See "Adding Elements to a Library" on page 42 to add new elements to the library file.

Note: The library file is automatically saved when you make changes. The library is also automatically saved when you quit ChannelPoint Merlot 1.0 or close the library file.

A

Example DTD

The ChannelPoint Merlot 1.0 installation includes an example plug-in called `example.jar`. This plug-in includes the `merlot-example.dtd` that was used to capture the images in previous chapters of this document. The full content of the `merlot-example.dtd` is included below for reference purposes.

```
1:  <!-- Example DTD for Merlot 1.0 -->
2:
3:  <!-- root element -->
4:  <!ELEMENT merlot-example (title?,div*)>
5:
6:  <!-- title element -->
7:  <!ELEMENT title (#PCDATA)>
8:
9:  <!-- div element -->
10: <!ELEMENT div (head?,(p | img)*)>
11: <!ATTLIST div id ID #IMPLIED>
12:
13: <!-- header -->
14: <!ELEMENT head (#PCDATA)>
15: <!ATTLIST head id ID #IMPLIED
16:                color CDATA #IMPLIED>
17:
18: <!-- paragraph element -->
19: <!ELEMENT p (#PCDATA | list)*>
20: <!ATTLIST p id ID #IMPLIED
21:             color CDATA #IMPLIED>
22:
23: <!ELEMENT list (item+)>
24: <!ATTLIST list id ID #IMPLIED
25:           type (unordered | ordered) "unordered">
26:
27:
28: <!ELEMENT item (#PCDATA)>
29:
```

```
30: <!ELEMENT img EMPTY>
31: <!ATTLIST img src    CDATA #REQUIRED
32:               width  CDATA #IMPLIED
33:               height CDATA #IMPLIED
34:               alt    CDATA #IMPLIED>
```

Glossary

Action plug-in – An Action plug-in is used when you want to define an action to perform on an XML document. An Action plug-in can act on any XML document and allows you to perform some action on that document. The XSLT Processor plug-in is an example of an Action plug-in.

attribute list – An attribute list links the attributes to a defined element and creates a single attribute named *type*. Attribute lists are built within the elements in an XML document to ensure the attributes are associated with the element.

attribute pane – The attribute pane appears to the right of the element tree, and it lists the current values for each attribute as they apply to the associated element. The attribute pane provides a way for the user to interactively apply values to each of the elements' attributes. Multiple attribute panes can be open at the same time.

attributes – In an XML document, an attribute is a sub-element within an element. The attributes of an element provide further description of that element and ensure the element is differentiated from other elements.

ChannelPoint Merlot – ChannelPoint Merlot is an open source XML editor that is driven by a DTD. The DTD is essentially a plug-in to ChannelPoint Merlot. The DTD determines which XML elements are valid in a form modeled with ChannelPoint Merlot.

DOCTYPE declaration – The DOCTYPE declaration is a required part of an XML document. The DOCTYPE declaration specifies the DTD associated with this XML document. It is usually included in the first few lines of the XML document. A DOCTYPE declaration can reference an internal DTD, which is DTD code included within the actual XML document, or an external DTD, which is a DTD saved as a separate file. The external DTD DOCTYPE declaration is the most common.

Document Object Model (DOM) – The Document Object Model is a common application programming interface for accessing documents within a web browser. The Document Object Model provides a language and platform-neutral object model for web pages. Because the Document Object Model interacts with document structures, it can be used by any application that access applications.

Document Type Definition (DTD) – The document type definition is the grammar for a markup language, as defined by the designer of the language. The Document Type Definition can be included within an XML document or contained in a separate file.

DOM – document object model. *See* document object model.

DTD – Document Type Definition. *See* Document Type Definition.

DTD plug-in – A ChannelPoint Merlot plug-in is an optional and specialized file used in ChannelPoint Merlot to accomplish an action with the resulting XML document; a DTD plug-in is one type of plug-in used with ChannelPoint Merlot. A DTD plug-in comprises a DTD and some Java code to make it work in ChannelPoint Merlot 1.0. The DTD plug-in declares the root element and the DOCTYPE definition, either public or system identifiers, that appear in the beginning of a resulting XML document. The DTD plug-ins provide a way to add DTD files to ChannelPoint Merlot's DTD Chooser dialog box.

element – In XML, an element is a piece of markup that you use to describe a set of content. Elements comprise tags, usually a start and end tag. Elements and tags are not the same thing.

element tree – The element tree contains the elements and attributes that can be used to build an XML document. ChannelPoint Merlot presents the DTD as an element tree, which is a graphical representation of the Document Object Model (DOM). Using the element tree, you can add elements and the assigned attributes by interacting with the ChannelPoint Merlot GUI.

Extensible Markup Language (XML) – Extensible Markup Language is a flexible way to create common formats and share both the format and the data to the web. XML describes web content in terms of what data is being described. It is extensible because the markup symbols are unlimited and self-defining. XML is a formal recommendation from the World Wide Web Consortium (W3C).

Extensible Style Sheet Language (XSL) – Extensible Style Sheet Language is a language that describes how data is presented in a browser window to the end user. An XSL style sheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses the formatting vocabulary. Like any style sheet language, XSL can be used to create style definitions that can be used for a single XML document or for multiple XML documents.

HTML – Hypertext Markup Language. *See* Hypertext Markup Language.

Hypertext Markup Language – Hypertext Markup Language is the set of markup symbols or codes inserted into a file that is intended for display in a web browser. The markup tells the browser how to display the text, images, and links for the user. HTML is a formal recommendation by the World Wide Web Consortium (W3C) as of the writing of this document.

Java Virtual Machine – The Java Virtual Machine (JVM) is software that converts the Java intermediate language (bytecode) into machine language so it can be executed. The original JVM was made and distributed by Sun Microsystems, but since then many companies have developed their own versions. A JVM is incorporated into some web browsers so that Java applets that are encountered by the browser's user can be executed. A JVM is often installed on a web server to execute server-side Java programs. A JVM can also be installed on a client machine to individual Java applications.

JHTML file – JHTML is a version of HTML specific to ATG Dynamo® that allows developers to put Java code into regular HTML documents. Each JHTML content file you build represents a single page of the web application. Each content file is saved with the JHTML extension. You build the content files that make up the web application pages and pull the content into the user's browser. Each content file is built as a page fragment because the layout and style are controlled by the skeleton and skin elements. The page contains only the content of each web application page.

JVM – Java Virtual Machine. *See* Java Virtual Machine.

library – A library is a collection of frequently used code or text to be reused across multiple platforms. The ChannelPoint Merlot 1.0 product allows you to open any XML document and create a library that can be used by multiple developers.

namespace – A unique name that identifies an organization that has developed an XML schema. It serves as a prefix so that multiple schemas can be used to define tags in an XML document. XML namespaces allow a single XML document to mix elements and attributes from multiple DTD files by adding a prefix to the element and attribute names. ChannelPoint Merlot 1.0 does not currently support namespaces.

open source – Open source refers to a set of source code that is provided to the development community free of charge. The intent is to produce a better product by having a larger group of developers working with the code and providing feedback. Open source software allows any organization to use and modify the product for its specific uses rather than wait for the vendor of the proprietary product to implement the suggestions for improvement in a subsequent release.

plug-in – Plug-ins are auxiliary programs that can be incorporated into larger programs to enhance their capabilities. A ChannelPoint Merlot plug-in is an optional and specialized file used in ChannelPoint Merlot to accomplish an action with the resulting XML document. There are two types of plug-ins used with ChannelPoint Merlot: a DTD plug-in and an Action plug-in. *See* Action plug-in and DTD plug-in.

required attribute – A required attribute is an attribute that must be included in the element when that element is defined in the XML document. The element is incomplete until that attribute is defined. An error message appears if the user attempts to save the element attributes without defining the required attributes. *See* required attribute flag.

required attribute flag – The required attribute flag is a graphical icon that appears in the attribute editing panel next to any attribute that is denoted as required. The icon is intended to alert the user that the attribute must be defined. An error message appears if the user attempts to save the element attributes without defining the required attributes. *See* required attribute.

root element – The root element of an XML document is the defining element of an XML document and encloses all other elements within the XML document. Each XML document can have only a single root element. The root element is considered a vital part of a well-formed XML document according to the XML specification published by the W3C.

style sheet – The style sheet is a master file used to store the margins, fonts, and other style-related classes. Names are applied to the styles, and the styles are applied to all text, images, and pages uniformly across the web application that has called the style sheet.

Unicode – Unicode is a superset of the ASCII character set that uses two bytes for each character rather than one. Able to handle 65,536 character combinations rather than just 256, Unicode contains the alphabets of most of the world's languages.

Universal Transformation Format – Universal Transformation Format is a method for converting 16-bit Unicode characters into 7- or 8-bit characters. *See also* Unicode.

UTF – Universal Transformation Format. *See* Universal Transformation Format.

value – The content of a field or attribute as it applies to an element.

XML – Extensible Markup Language. *See* Extensible Markup Language.

XSL – Extensible Style Sheet Language. *See* Extensible Style Sheet Language.

Index

