ABSTRACT

An Internship with E-Publishing Companies:
InformIT.com and Macmillan Publishing

By Hongxia Shi

This report presents an account of a graduate internship that was completed in partial fulfillment of the requirements for the Master of Technical and Scientific Communication degree from Miami University, Oxford, Ohio. I performed this internship with two major e-publishing companies based in Indianapolis, Indiana, from March 20 to July 10, 2000.

In this report, I discuss my work as a Web Technical Writer/Editor at InformIT.com and Macmillan Publishing. I provide an overview of the various projects and activities I performed and the time commitments I gave to these projects and activities. I also discuss in detail the procedures I conducted to accomplish one major project. I conclude the report with a review of the Problem Solving Model and a reflection of the connection between what I learned in the MTSC program at Miami University and my practice as a Web Technical Writer/Editor in the publishing industry.
An Internship with E-Publishing Companies: InformIT.com and Macmillan Publishing

An Internship Report

Submitted to the

Faculty of Miami University

in partial fulfillment

of the requirements for the degree of

Master of Technical and Scientific Communication

Department of English

By

Hongxia Shi

Miami University

Oxford, Ohio

2000

Advisor

Prof. Jennie Dauterman

Reader

Prof. Paul Anderson

Reader

Prof. Gary Shulman
# Table of Contents

Chapter 1: Introduction................................................................. 1

Chapter 2: Internship Overview..................................................... 5  
  Project Management........................................................................ 5  
  Learning ........................................................................................ 6  
  Writing .......................................................................................... 9  
  Editing ......................................................................................... 10

Chapter 3: Major Project ............................................................... 13  
  Project Introduction ..................................................................... 14  
  Link Problems ............................................................................ 15  
  Graphic Problems ....................................................................... 16  
  Special Characters Problems ...................................................... 16  
  Text Problems ............................................................................ 17  
  Code Listing Problems................................................................. 19

Chapter 4: Reflection and Conclusion ......................................... 21  
  Problem-Solving Model ............................................................... 21  
  Working Experiences Related to the MTSC Program ................. 24  
  Conclusion .................................................................................. 26

References..................................................................................... 27

Appendix A: Editing Checklist....................................................... 28

Appendix B: A Global Deficiency Report...................................... 30

Appendix C: InformIT HTML Review Guidelines ....................... 33

Appendix D: ISO8859-1 Table of Special Characters 46

Appendix E: Link Check Report .................................................... 49

Appendix F: Editing and Shipping List .......................................... 51
Chapter 1: Introduction

From March 20 to July 10 of 2000, I performed my internship with two major e-publishing companies based in Indianapolis, Indiana, namely InformIT.com and Macmillan Publishing. My title was Web Technical Writer/Editor.

Now an independent Web company, InformIT.com was once a department in Macmillan Publishing. Both InformIT.com and Macmillan Publishing are parts of Pearson PLC.

Pearson PLC (hereinafter as Pearson) is an international media company with market-leading businesses in education, strategic business information, international television production and consumer publishing. Pearson, however, started out its life as a very different company. S. Pearson and Son started out as a small building firm in the West Riding of Yorkshire, England in 1844. In the next fifty years, Pearson entered into the media business with the acquisition of the brands that support the company. Today its professional and technology group publishes over 1,000 computer and technology titles a year. Its imprints include Macmillan Publishing, Prentice Hall, Cisco Press, Adobe Press, New Riders, Peachpit and Addison Wesley complemented by an online presence—InformIT.com (www.informit.com).

As a dot-com company, InformIT.com was chartered in the October of 1999. There were about 50 employees and five departments when I started my internship. These departments were Department of Corporate Sales, Department of Human Resource, Department of Marketing, Department of Product Development, and Department of Project Management. InformIT.com aims to be the leading e-learning and content site for
IT students and professionals. To accomplish this goal, it has developed various products with broad IT contents to attract IT customers. Its products include online IT articles, e-books, access to industry experts, training courses, and news.

The Product Development Department was a very important part of the company. From March 20 to June 2, I performed my internship in the HTML Conversion/Reviewing team in this Department. The team was responsible for providing HTML files for online IT articles and e-books. It consisted of two Technical Writers/Editors, an HTML Conversion Analyst, and a Project Manager of Conversion. As one of the Web Technical Writers/Editors at InformIT.com, my primary responsibility was to use my knowledge of HTML to edit the converted HTML files before their release to the World Wide Web. I also conducted research on the format of HTML files, and consulted with the HTML Conversion Analyst on technical issues. I sometimes worked with the other writer, but mostly I worked alone.

About 1/3 of my internship was performed at Macmillan Publishing, which is the parent company for InformIT.com. As a unit of Pearson, Macmillan Publishing is the world’s largest computer book publisher and one of the industry’s leading innovators in both print and digital formats. Its imprints include Que, Sams, BradyGAMES, the Complete Idiots Guides and Macmillan Software. Together, they sell more than 14 million books worldwide each year.

Founded in 1981, Que books have taught more than 75 million people how to use computers. Sams Publishing began in 1946 as a leading provider of electrical books. It is focused on teaching tomorrow’s programmers, developers, and systems administrators by publications from introductory tutorials to comprehensive reference books. As a fairly
new imprint, the Complete Idiot's Guides publishes books of various topics. These topics include personal finance, business, health/fitness, foreign language, new age, and relationships. BradyGAMES imprint is a leader in interactive entertainment publishing, offering value-priced strategy guides for video and PC-based games. Macmillan Software was formed in 1996 to centralize the electronic and digital assets of Macmillan Publishing. Today it develops market-leading products that range from Web page construction tools, to MP3 technology, to award winning Linux products.

Realizing published books could no longer be the only source of information in the rapidly growing information technology market, Macmillan established online publishing. Its first effort was Macmillan Online (www.mcp.com) in 1994, which has since become one of the top information technology resources on the Internet. A Conversion Group in the Production Department of Macmillan Publishing is responsible for all the conversion of published books from all Macmillan imprints and other select imprints of Pearson.

On June 5, 2000, the HTML Conversion/Reviewing team at InformIT.com was moved to Macmillan Publishing. As a result, I performed the rest of my internship at Macmillan Publishing. As far as I know, there were mainly two reasons for this shift. First, InformIT.com went through company reorganization and budget cutting after a new president was hired in April. Under the new structure, the Conversion/Reviewing team couldn't fit in any department. Second, Conversion at Macmillan started in 1994 when Macmillan Online was launched. Its conversion group had strong technical and editorial support. By shifting its conversion to Macmillan, InformIT avoided the large expenses to
maintain the conversion team, and it could still get high quality contents from Macmillan. Their relationship changed to that of a client/provider one.

After the shift, I moved to a different location and reported to a different supervisor. My job title didn’t change, and my major responsibilities were the same except that I performed HTML conversion duties as well. This duty is described later in this report.

During my internship I saw that much of what I learned in Miami University’s MTSC program transferred well to real-world projects I did at InformIT.com and Macmillan Publishing. In the following sections of this report, I’ll discuss what I have performed during my internship. The sections will be presented in this order:

Chapter 2, Internship Overview, is a discussion of the tasks I performed during my internship.

Chapter 3, Major Project, is a discussion in detail of the process I used in fulfilling my tasks.

Chapter 4, Reflection and Conclusion, is a discussion of my professional development through the internship.
Chapter 2: Internship Overview

The Production Departments in InformIT.com and Macmillan Publishing handle all the content conversion jobs. The contents come from titles produced by several different publishers within Pearson Education. Most of the titles we worked with came from titles published by Cisco Press, NRP (New Riders Press), Sams, and Que.

As a Web Technical Writer/Editor, my primary responsibilities were to correct source file errors, fix broken hyperlinks and graphic links, link cross-references, and make sure the format was consistent. It was very hard to keep consistency of styles and format for the HTML files. Different style guides were used on the original hard copies among different presses, and within one press different styles were sometimes applied to different series of titles. For example, the Sams style guide is different from that of Cisco, and Sams Teaching-Yourself series is styled differently from Sams Unleashed titles. It was my task to look at these styles and came up with one that would be used consistently throughout on the Web site. When I started my internship with InformIT.com, they didn’t have any style guide, and they didn’t know what exactly they wanted the online content to look like. Therefore, my responsibilities included not only editing (60%), but also some writing (17%), learning (13%), and project management (10%). These tasks are discussed in detail below.

Project Management

According to Badiru (1993), “project management is the process of managing, allocating, and timing resources to achieve the desired goals of a project in an efficient and
expedient manner. The objectives that constitute the desired goal are normally a combination of time, cost and performance requirements”. Badiru’s definition implies two goals for project management. The first is the efficient and effective operation of the project itself, and the second goal relates to performance.

Most of my project management efforts during my internship were spent fulfilling the first goal that Badiru discussed, i.e. scheduling the deadlines for completing projects. Sometimes, different projects were assigned to me concurrently. I had to make sure that each deadline would be met according to the company’s regeneration schedule. In InformIT.com, the regeneration process was called Regan, which scheduled the company’s Web site contents to be changed on a monthly basis. I met with my tech-editing supervisor once every week to present our deliverables and update our project schedule. The Regan dates of titles were set according to the priorities established by the Acquisition Editor and the Web Master. So updating our project priority schedule was often a continuous and time-consuming process with the constant changes of the Regan dates. However, since the projects were not too large — just some sample chapters and articles, I was able to meet all the deadlines and to proceed with the projects in a smooth manner.

Learning

During my internship, I spent about 13% of my time on focused learning activities. These included learning HTML tagging; self-instruction on a new computer application, Dreamweaver; training sessions on non-book contents and the conversion process.
Learning HTML Tagging

HTML tagging was involved throughout my internship. Though I had learned some basics about HTML at school, I felt I needed to understand the whole concept in order to correct format errors that I encountered frequently.

   HTML (Hypertext Markup Language) defines a set of common styles for Web pages: headings, paragraphs, lists, and tables. It also defines character styles such as boldfaces and code examples. Each element has a name and is contained in what’s called a tag. To use HTML, I needed to learn structuring tags and text formatting tags. The structuring tags include title tag, heading tags, paragraph tag, and list tags. I also learned to create links and anchors. Links include internal links and external links. Internal links are reference links to tables, graphics, and different sections or chapters of a document; external links are links to other documents or articles on other Web sites. Anchors create links to specific points inside documents. The text formatting tags include the preformatted text tag, horizontal rule tag, line break tag, and tags for special characters, text alignment, font faces, and font sizes. I came to use this language a great deal in the editing of the web site.

Learning Dreamweaver

To help me write HTML code, I used Macromedia Dreamweaver, a professional visual editor for creating and editing cross-platform and cross-browser Web pages. I had no experience in Dreamweaver at all when I started my internship. In the first few weeks of my internship, besides finishing my daily editing work, I tried to set aside time to teach myself some basic knowledge of Dreamweaver. I briefly read the manual, performed the
tutorials, and tried out the various features. By taking the time to thoroughly acquaint myself with the application, I was soon able to use it effectively.

**Non-book Content Project Training Session**

Most of my editing tasks were to edit converted sample chapters from published titles: this was called book content project editing. There was another project group preparing HTML files of IT articles, tutorials, case studies, and expert recommendations: this was called non-book content project editing. At a certain point, the non-book content project editing group had heavy workload and needed help, so I was given a training session to learn how to put non-book contents on the Cisco Knowledge Suite (a category on InformIT Web site). Since the non-book contents were handled only slightly differently from the book-contents, I was able to perform the project after the brief training.

**Learning the Conversion Process**

After my team moved to Macmillan Publishing, my responsibilities were expanded a little bit. Besides tech-editing the converted files, I learned the Macmillan’s in-house conversion process and did some whole title conversion and sample chapter conversions.

Macmillan’s in-house conversion started with the archived CDs of PageMaker files or QuarkXpress files of published books. These files were exported into Word, and I applied Macmillan online styles to these files for the converters to recognize and convert them into different format in target HTML files. Macros already written managed special style or format problems such as note text, sidebar text, em dashes, en dashes, etc. After the Word files were properly styled and all macros were run, they were saved as one RTF file (.rtf, rich text format). The RTF file was then put through converter to generate HTML files. Different converters were used according to different layout requirements for the
resulting HTML files. Macmillan in-house conversion mostly used two converters: ERL-
Main and ERL-eSTY. These converters were actually purchased off-the-shelf and then
customized according to Macmillan specifications.

The ERL (Electronic Reference Library) Project was a project using the above-
mentioned converters to prepare categorized IT titles as electronic files. These HTML
files were then edited and archived to put in Macmillan’s Electronic Reference Library
for clients to order as needed for their Intranets. I was involved in both the converting
process and the editing process in this project.

Writing

On the day I started my internship with InformIT.com, I was told where the converted
files were, and what I should check for in general, and off I went. I didn’t have any style
guidelines, or any checklists to follow. Some oral instructions were all I got. I wanted to
say: Wait a minute! You need to sit down and do some documentation first. As I
progressed with my work, I found it very difficult to standardize the HTML files and
keep them consistent because there was no style guide or other documentation. So, as I
worked, I kept records of all style and format problems that I encountered, and did a little
bit of writing along with my editing processes.

By the second week of my internship, I had gathered an Editing Checklist that
contained the most commonly needed checks for the editing process. The checklist
proved to be very useful for a colleague who joined me later. A copy of this checklist is
presented in Appendix A.

In the third week, I presented a global deficiency report, and was able to discuss the
problems with my supervisor. By tracking and reporting the recurring issues, I was able
to help improve the whole conversion process. For example, I saw that some of the problems got fixed globally in the Conversion House in India where InformIT.com sent their conversion work. Appendix B includes a copy of the global deficiency report.

When InformIT.com decided to move the HTML Conversion/Reviewing Group out to Macmillan Publishing, I was assigned to write a set of HTML review guidelines. This document was to serve as spec-sheet and style guide when Macmillan Publishing prepared the HTML files that InformIT.com requested. I spent quite a lot of time and energy on it, and I implemented the problem-solving model I learned in the MTSC program at Miami University on it by doing audience analysis. This is discussed in detail in Chapter 4.

After I did my informal audience analysis, I studied both the items I had on my Editing Checklist and the record of style and format problems I kept for each title I edited. These studies helped me to decide what items to put in the guidelines and how detailed the contents should be. After the guidelines was developed and user-tested, it was sent to Macmillan Publishing and used by the HTML editors, the conversion technicians, and conversion analysts. It was still serving as InformIT.com’s HTML review specifics when I left Macmillan at the end of August 2000. A copy of the Guidelines is presented in Appendix C.

Editing

About 60% of my internship time was spent on HTML editing, which was slightly different from conventional editing in two respects. First, the editing was done in an HTML code writer instead of a text editor. Second, the documents being edited were all published books that had established content, organization, format and style. But during
the conversion, all these elements might get lost or changed. It was my role to make sure the converted files were as high in quality as the original ones.

Despite the slight difference, my editing processes still fell under the traditional 9 categories and 5 levels of edit proposed by Buehler (1981) (see Figure 1 below). Five of the nine categories and 4 of the levels appeared in my HTML editing (see Figure 2 below).

**Figure 1:** Buehler’s Editing Levels

<table>
<thead>
<tr>
<th>Type of Edit</th>
<th>Level of Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Level 1</td>
</tr>
<tr>
<td>Coordination</td>
<td>X</td>
</tr>
<tr>
<td>Policy</td>
<td>X</td>
</tr>
<tr>
<td>Integrity</td>
<td>X</td>
</tr>
<tr>
<td>Screening</td>
<td>X</td>
</tr>
<tr>
<td>Copy Clarification</td>
<td>X</td>
</tr>
<tr>
<td>Format</td>
<td>X</td>
</tr>
<tr>
<td>Mechanical Style</td>
<td>X</td>
</tr>
<tr>
<td>Language</td>
<td>X</td>
</tr>
<tr>
<td>Substantive</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2:** Buehler’s Levels I Encountered

<table>
<thead>
<tr>
<th>Type of Edit</th>
<th>Level of Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Level 1</td>
</tr>
<tr>
<td>Policy</td>
<td>X</td>
</tr>
<tr>
<td>Integrity</td>
<td>X</td>
</tr>
<tr>
<td>Format</td>
<td>X</td>
</tr>
<tr>
<td>Mechanical Style</td>
<td>X</td>
</tr>
<tr>
<td>Substantive</td>
<td></td>
</tr>
</tbody>
</table>

As it shows in Figure 2, the five categories that occurred in my editing were Policy, Integrity, Format, Mechanical Style, and Substantive. In policy editing, I made sure the document was consistent with InformIT.com policy. In integrity editing, I matched all figure references to figures, table references to tables, code listing references to code listings, chapter references to pertinent chapters, etc. In format editing, I edited for format adherence, such as format of headings, paragraph form, list format, note format, table
format, fonts, etc. In mechanical style editing, I edited for typographical consistency—e.g. abbreviations, special characters, capitalization, use of numbers and symbols, etc. In substantive editing, I edited for consistency, completeness, and coherence. I’ll discuss my editing processes in detail in Chapter 3.
Chapter 3: Major Project

This chapter discusses the Book Content Project I performed during my internship. This project was the major project throughout my internship (both at InformIT.com and Macmillan). The goal for this project was to prepare high quality HTML files of IT contents for the customers. This goal is also the fundamental goal for both InformIT.com and Macmillan Publishing.

There are two major reasons why this goal is central to the organizations. The main reason is that the organizations’ practical experience with IT professional development has proven the importance of high quality online contents. As I mentioned in Chapter 1, Macmillan has always been on the cutting edge of online publishing. Its first effort Macmillan Online in 1994 was a great success. The launching of InformIT.com has provided the IT market with an even greater depth of content, strength in community, and expanded commerce potential. The in-depth contents include:

- Editors’ Choice—recommended content from best-selling authors and industry experts.
- Readers’ Choice—the most popular content selected by users.
- AlphaBooks—content from books-in-progress.
- InformIT’s Free Library—free access to hundreds of complete online books.

The Book Content Project was essential for providing the in-depth contents for different IT readers. During my internship, readers’ feedback about the contents was constantly forwarded to us to assure the importance of the work we were doing for both the company and the readers.

Another reason why this goal is central to the organizations has to do with their marketing strategy. As more and more people move away from traditional purchasing to
online purchasing, InformIT.com and Macmillan also wanted to continue developing the
online market to secure both their dominant positions in supporting IT professionals and
in their sales volume.

In this chapter, I discuss this project, particularly the editing process, in great detail.
This detailed description helps illustrate my observations on screen text editing and
printed text editing, which will be discussed in Chapter four.

Project Introduction

The titles in the Book Content Project were mainly computer-related books recently
published. Sample chapters from these titles were converted and reviewed to put on the
Web site for readers to read for free before making decisions to buy the hard copies.

For this project, the files were already converted before they went through the editing
process. However, because of the complicated conversion process, those files needed
many repairs before their release. Here, I’ll give a very brief description of the conversion
procedure.

After the acquisition editors decided which titles to put on the site, input files were
requested from printing companies. These data were then marked up in XML (Extensible
Markup Language), and then run through Epic Viewer (software converting XML files to
HTML files) to generate HTML files. In the rest of this chapter, I’ll describe my editing
work for this project in detail.
Link Problems

One of the advantages of online reading is that readers can jump between modular texts freely. Both internal and external links make this possible. Every file I reviewed included some kinds of link references—both internal and external links.

Internal links are ones that readers use to shift between sections of one file or among a folder of files. These links always worked so long as the files linked were present. My work was to click on the existing links and check whether they attached to the right material, and to insert reference links where they were needed.

External links were generally HTTP ("Web Page") URLs (Uniform Resource Locators). Other types of URLs involved were: FTP URLs (pointing to files on FTP servers), File URLs (pointing to a file housed on a local disk), Mailto URLs (used to send electronic mail), Gopher URLs (pointing to files on a Gopher server), and Usenet URLs (pointing to newsgroups or specific news articles in a newsgroup). About 60% of these URL links were obsolete because the server no longer supported them, they were already out-of-date, or they were not formatted correctly. My job was to make sure all these URL links went to an active page when readers selected them.

When I came across a link, I would first check to see whether it had properly formatted headers, such as “http://www…,” ftp://ftp…,” “file://localhost…,” etc. Then I would check the spelling and spacing. After I finished editing a file, I usually used Netscape Navigator or Internet Explorer to check all the external and internal links. Internal links usually worked well if I had linked them to the correct files. If I had an external link that didn’t work, I had to search the World Wide Web for the proper content. This was often very time-consuming. When I found hyperlinks that were dead or
required the end user to supply a password, I would delete the links and keep the text. I would note these on the link report when the files were sent to InformIT.com.

Graphic Problems

When a sample chapter contained graphics, it could be extremely difficult to present them in the text as hard copies do. We decided to link these figures to pop-up windows so the files would be easier to handle. For example, if there were a Figure 1.1 in the text file, I would link Figure 1.1 to the actual figure housed in a graphics folder inside the file folder. The tag looked like this:

\[ \text{<a href = “javascript:popUp('elementLinks/01fig01.gif')”}>Figure 1.1</a>.} \]

Here \textit{javascript:popUp} is a JavaScript routine put at the beginning of the HTML file to pop the figure window up. The \textit{elementLinks} is the graphics folder.

Besides inserting the pop-up tags to link the figures, I also checked to make sure the referencing graphics matched. Often, there were unclear or oversized graphics, which I had to rework in PhotoShop. If the graphics couldn’t be made better in PhotoShop, I sent them to be reconverted for better quality.

Special Characters Problems

The most common special characters encountered in the titles were em dashes and en dashes. These characters were never converted correctly. Almost all em dashes were converted as two hyphens (--), and en dashes were converted as one hyphen (-). Here is a story that made me think how helpful my training in Technical Writing had been to me even if on small things such as em dash and en dash.
When I first started the Macmillan ERL project, I noticed em dashes in the form of -- appeared in many of their ready-to-deliver files. I asked an editor and an HTML technician about this em dash and en dash issues (none of them had technical writing background), and they acted like they had never heard about them before. Em dashes and en dashes had apparently been confusing to my colleagues even though they are very important in the publishing business. In HTML tagging, em dashes, en dashes, and other character entities such as ®, |, and µ all have their own numeric codes. We used the numeric codes from The ISO8859-1 Table of Special Characters, because character codes are not always correctly interpreted by some browsers. A copy of The ISO8859-1 Table of Special Characters is attached in Appendix D.

**Text Problems**

Text problems are very tedious and time-consuming, yet very important. Presenting to the readers a Web page that is full of broken sentences or missing text is unprofessional and will drive the readers away. So I spent quite some time doing content and format checks.

In content checks I compared the electronic files against the hard copies to check for missing, misplaced or duplicated text. Sometimes I found copyediting / proofreading errors—typos or misused words in the hard copy itself. For example, in chapter 4 of title 1578700469 (this number is the book’s ISBN), there were instances like these “one of most important features…,” “At this point, your changes will not take affect,” and “…have to keep track of to many…” where they should have been “one of the most important features…”, “At this point, your changes will not take effect”, and “…have to keep track of too many…”.
In format checks I read HTML files against the hard copies to correct inconsistent
typefaces on emphasized words, word clusters (words run together without any spaces in
between), excessive or insufficient space between paragraphs, and the format of note text
boxes and footnotes.

Most of the note text boxes in printed copies were either within a lined text box or a
shaded text box. In online texts it would not be practical to use lined or shaded boxes.
The InformIT style required that we put the text in <blockquote> tags. So I would check
that all note texts were in <blockquote> tags and all note headers—NOTE, CAUTION,
TIP, WARNING, PETER’S PRINCIPLE, etc. would match what appeared on the hard
copies.

While it was easy to deal with the note text, footnotes were difficult to handle. In the
printed copies, footnotes appeared at the bottom of a page, but in online text, there is no
“page”. A Web page is one single file that may be many pages long. So footnotes that
appeared at the bottom of a hard copy page would appear in the middle of a file, and it
could be very hard to tell which place the note referred to. When I first came across
footnotes in the files and raised this issue, the resulting discussions generated several
“solutions”. At last, we decided to put a mark [ref] at the place where traditional “1” or
“2”, etc. would appear, and link it to a pop-up window which contained the actual
footnote text. So when I came across a footnote, I would delete the number, put [ref]
there, and then copy the footnote text into a new HTML page that used the same naming
convention as the others. After I had saved the file in the graphics folder, and linked [ref]
to the file that I had just created, I would check to see whether the link would work. It
was fun to see footnote files I created pop up on the screen.
Code Listing Problems

Some books I reviewed contained computer-programming codes that were several pages long. These code listings were converted into different faulty formats during the conversion process. One typical faulty format was the <div class=“literallayout”> format. I would first replace this kind of format with <pre class=“programslisting”> format to remove the extra line breaks. Then I would check to see whether the lines were too long. When a line of preformatted code exceeds 72 characters, I would insert a code continuation character to break the line so that readers didn’t have to scroll over the browser to view the codes. But the code continuation character may cause problems when readers copy and paste the code in usage immediately. So InformIT.com utilized the power of hyperlinks to provide its end users with a version of the code listings that can be copied-and-pasted into an application.

I typically generated 2 copies of the file when I had long code listings. The first copy would be the copy with the code continuation character inserted. This character was a .gif (graphics interchange format) file. This image was placed in the elementLinks folder along with other graphics. I used this tag to insert the image:

<img src=“elementLinks/ccc.gif”>, where “ccc” was the name of the image file.

The second copy would be the hyperlink copy. I used the JavaScript pop-up codes for graphics to pop up code listings that needed code continuation characters. This would provide the end user with a true code listing without the .gif image. So before inserting the code continuation characters, I would copy and paste the original code listings into a new HTML file and save this file in the elementLinks folder. A reference line such as “See this code sample” would be inserted and linked to the HTML file. I used this tag
line to insert the file link: `<a href="javascript:popup('elementLinks/04code01.html')">`, where “04code01.html” was the name of the HTML file. Examples of both copies are presented in Appendix C.

After I finished checking all these problems in the chapters of a title, I would use a feature in Dreamweaver to check the links between those chapters and to make sure there were no broken internal or external links. Then I would present a Link Check Report together with Editing and Shipping List to my supervisor. A copy of a Link Check Report is included in Appendix E. A copy of the Editing and Shipping List is included in Appendix F.
Chapter 4: Reflection and Conclusion

In this section of the report, I will discuss how what I learned at school transferred well to real-world projects even in a rather limited role of an HTML editor. First, I’ll discuss how the Problem-Solving Model has helped me in my limited writing tasks and in improving the usability of the contents I edited. Second, I’ll discuss how my experience as a student in the MTSC program related to my internship experience.

Problem-Solving Model

Problem-Solving Model is a technical communication model proposed by Dr. Paul V. Anderson from Miami University, Oxford, Ohio. This model was first introduced to me in “Fundamentals of Technical Writing” course at Miami University in fall 1998. It describes the following five recursive stages of documentation:

- **Define the Problem.** This is the initial investigation into the objectives, audience, and context of the document. It also includes defining constrains or conventions that apply to the document.

- **Design Activities.** This is the stage to design related tasks and to make preliminary decisions about aspects of the document, such as its medium, form, style, means of production, etc. Gathering additional information and producing a review draft are also activities at this stage.

- **Test Initial Document.** This is the stage to design procedures for usability testing. It also includes presenting draft to reviewers and/or the intended audience.
• **Implement the Design.** This stage involves revising the draft and producing the final document to deliver to the intended audience.

• **Evaluate the Solution.** This stage determines the document’s effectiveness. Revision should be made if necessary.

This model provided me a framework around which I could schedule my activities. It helped me to think more clearly about my writing and editing projects. In the course of producing the HTML Guidelines and conducting the editing projects, I found myself constantly referred to this model on conducting Audience Analysis, Task Analysis, and Usability Testing.

**Audience Analysis**

When producing the HTML Guidelines, I determined that the primary audience of the Guidelines would be people in the conversion group of Macmillan Publishing. Other audiences might be HTML people in InformIT.com and department managers on both sides. Within the Macmillan conversion group, people would use the Guidelines differently. The editors’ goal was to follow them word by word to make sure every item would be covered in the editing. The conversion technician’s goal was to use them as a general guide while conducting the conversion. The conversion expert’s goal was to thoroughly transfer the guidelines when defining style sheets. With these audiences in mind, I decided there should be both explanations and example tags under each item.

By doing informal audience analysis like this, I was able to keep my audiences in mind while I was editing. The audiences for the items I edited consisted of people who visited the company’s Web site. Though the primary audience would be the IT students and professionals, everyone who had access to the Internet could be the audience.
Thinking through the different ways users might want to read on the site enabled me to suggest solutions to some problems, such as footnotes and long listings of code, and provide them with the best navigation tools. From the way they read on-line, to the way they click through a site, to the way they react to various kinds of contents, Web users demand information that is tailored to their unique needs. That’s why usability expert Jakob Nielsen (1998) urges web designers to put users and contents first in all their decisions about design process. His usability studies have found that users care primarily about good content on web sites, ignoring such features as self-promotion or flashy graphics treatments. Because good content in a consistent format makes it easy for users to navigate and grasp information.

**Task Analysis**

My task analyses were also informal. I usually just wrote them out on a piece of notebook paper that I could easily refer to while I was writing or editing. By doing so, I was able to identify areas of the design that were missing, disorganized, or inappropriate in both my writing and editing. Because technical writers must think about tasks from the users’ perspective, rather than from the converting technician’s prospective, we are uniquely qualified to comment on the usability of the products.

**Usability Test**

The Problem-Solving Model reminds us of the importance of evaluation to any communications project. During the evaluation phase, you prepare for usability test as you plan for external evaluations of the success of the projects. My colleague did the usability test of the Editing Checklist and the HTML Review Guidelines that I wrote. As for the documents I edited, circumstances required me to be an audience and do usability
test by myself. After I edited one file, I would put them on the World Wide Web and click to see whether they worked well. From these usability tests, I was able to define some problems at an early stage and prevent further damage they might have caused.

Working Experiences Related to the MTSC Program

Looking back, I found my experiences as a student in the MTSC at Miami University were very helpful for the success of my internship. Three of my experiences in the MTSC program were particularly relevant to my working experiences during the internship. The first experience was from my “Organizational Communication Theory and Research” course, in which I learned how organizational structure impacts organization performance. The second experience was from the “Technical and Scientific Editing” course, in which I learned basic technical editing skills. The third experience was from the “Technical and Scientific Writing” course where I learned some principles of online design. Below is a detailed discussion of how these experiences were related to my working experiences.

Impact of Organizational Structure

I learned from my “Organizational Communication Theory and Research” that the structure of an organization is a critical factor in determining how information is disseminated within the organization. In turn, how information is shared in an organization impacts employee productivity and satisfaction.

During my internship at InformIT.com, I quickly realized that the company had a very loose organizational structure. On the one hand this loose organizational structure cultivated an informal atmosphere, but on the other hand it prevented direct information sharing.
As I mentioned in Chapter 1, the HTML Review group consisted of myself as a Technical Writer/Editor, another Technical Writer/Editor, an HTML Conversion Analyst, and a Project Manager of Conversion. A Content Project Manager and a Manager of Research and Development also got involved in HTML Reviewing issues from time to time. Lacking information sharing, my supervisor and the other two managers sometimes were not on the same page, because we sometimes got different directions from these three managers. And on some occasions, we had to guess what was the right thing to do because we were not given directions. As a result, we found ourselves frequently going back to make changes on what we had been told to do or on what we had figured out to do by ourselves. Therefore, the loose atmosphere affected our productivity. Clear and up-to-date directions were what we were longing for then.

**Need for Diplomacy in Technical Editing**

I remember in doing our class projects in “Technical and Scientific Editing” class, we were required to attach a note to the edits to explain why we made the changes and to persuade the readers to accept the changes.

In the process of my editing, I had several occasions in which I had to explain the changes I made in an original text because of typos, cross-reference errors, or improper usage of words. Thanks to my training in the editing course, I was able to be very diplomatic in explaining these to the editors. Diplomacy was also needed when I was discussing solutions to the editing problems with my colleagues and supervisors. All of them had years of online experience with Macmillan Publishing. My diplomacy, respect to them, and goodwill enabled me to cooperate with them in a pleasant way.
**Need of Online Design Knowledge**

From my readings on online design in “Technical and Scientific Writing” class, I knew that Web editing has its own unique characteristics and activities because of the differences between screen text and printed text.

As we all know, it is very important to have control of the text while we read. In printed texts, readers can consult the table of contents, headings, and subheadings to control their reading of the text. But with a screen text, readers have difficulty sizing up the whole text, and getting a full sense of how much information may be present or how much has been viewed. So it is important for online text to contain navigable screen-sized module texts so that readers will not become disoriented.

It is also true that reading from screens usually slows people down and stresses their eyes. So restricting the text line length and making spacious text is also very important. With such online design knowledge in mind, my editing was very efficient and effective.

**Conclusion**

Overall, my internship shows that what I had learned at school in the MTSC program transferred well to my real-world projects. My experience shows that without the foundation of a clear understanding of the Problem-Solving Model, the focus of any document could have become obscured among the detailed information. My experience on the job taught me that basic technical editing skills and online design knowledge is important for a successful experience in the publishing industry.

My internship experience was very valuable to me in developing my career as a technical writer. It has convinced me that technical writers can contribute meaningfully and significantly in any document development or editing job.


Appendix A

Editing Checklist

This document is not a complete list, but it contains some of the most commonly needed checks.

Web Site Problems
Check all the web site links and make sure they work.

Graphic Problems
1. Check all figure links. Change a dead figure link into a javascript pop-uplink by using tag such as `<a href="javascript:popUp('elementLinks/16fig01.gif')">`.
2. Check figure itself. Some figures are very unclear, and others may miss some parts.
3. Check for misplaced figures according to the hard copy.

Table Problems
Check indentation and typeface of table headings.

Character Problems
1. Check for double hyphens and replace by em-dash.
2. Check for extraneous underscores_ and replace by one underscore. (This is something Jenny came across. I didn’t run into this one.)
3. Abbreviations such as “he’s” “you’re” sometimes appear as “he&#162;” “you&#162;” in html file, which are displayed as a small empty boxes in the browser. Should be change into normal text.
4. Some special character entities, such as Greek characters will be captured incorrectly. Check and change them.
5. Check for superscript and subscript characters.

Text Problems
1. Check for missing text, misplaced text, and duplicated text.
2. Check for correct typeface for some emphasized words according to the hardcopy.
3. Check for word clusters (words run together without any space between). Sometimes appears after italicized words, sometimes appears surrounding anchors that occur in a text string.
4. Check for excessive or insufficient space between paragraphs.
5. Check to make sure all Note, Caution, Tip, Warning headers are displayed correctly according to the hard copy.
6. Check for table notes or footnotes. Create new files for these notes and locate them in the “elementLinks” directory. Use JavaScript pop-upwindows to display these notes.
7. Check all chapter reference links. Make sure they work.
8. Look for section level references and link them to the right sections.
Code Listing Problems
1. Most code listings were in <div class="literallyout"> format, which is not the right format. Check and replace this format with <pre class="programlisting"> format, then get rid of the excessive line breaks (<br>) after the <pre> tag is applied.
2. Check for code lines that are too long. Insert a code continuation character (graphic) to show the continuation of the codes. The graphic should be located in the “elementLinks” directory.

Quiz/Quiz Answer Problems
Some titles have Quiz/Quiz Answers Parts. There are excessive links existing in these parts. Make sure to insert a link to link the quiz questions to the quiz answers. Delete the excessive links if appropriate.
Appendix B

A Global Deficiency Report

• Several varieties of in-text notes to the reader are usually used: Notes, Tips, Cautions, and Warnings are the most although we do occasionally encounter text coded with a <div class='note'> tag that looks more like a sidebar. The header for these items sometimes doesn’t import at all. When it does, it is always tagged with a <p class> tag (e.g., <p class = 'normaltitle'> , <p class = ‘tiptile’> , etc.) The header, however, always imports as NOTE regardless of the <p class> tag. That is, even when a Tip imports, and the header tag says <p class='tiptitle'>, the header says NOTE, and has to be manually changed. We’d like for the conversion routine to automatically make the header match the <P class> tag.

• All notes, tips, etc. need to be manually <blockquote>-ed. To differentiate them from subsequent text. Without blockquoting, there is no way to tell where the note leaves off and the regular text resumes. Could we add a <blockquote> tag to the <div class='note'> tag, so this happens automatically?

• An extra line is always inserted after a <preformatted> paragraph. This results in too much white space whenever a style other than <preformatted> follows. The extra line break has to be manually removed at present.

• Links for all figure references require manual updating. If we could set the conversion routine to automatically point to the Element Links directory (e.g., “element links/02fig01”), these links would usually work.

• We are correcting virtually the identical text problems in every book. Double hyphens (Em-dashes) and extraneous underscore (_ ) characters most frequently need to be fixed.

• All figures have at least one reference link embedded in the text. The caption is embedded in the text, too. When you view the figure, no information accompanies the graphic. Would it be possible in future to capture the caption and figure number as part of the graphic? I believe these are redundant, sitting alone in the text, and do more for the user if they’re with the graphic, explaining what the picture is.

• No external links (links to other web sites) work as-is. They always require that the “http://” portion of the link be manually added. Is there any way to have the conversion routine insert this bit of text whenever it encounters a link beginning with a “www”?
• Internal links (to sections within the same book) are only consistently included for the following instances:

  Links to named anchors (figures, tables, examples) in the same file.

  Chapter links that include the target chapter number (i.e., “... see Chapter 3, Using the ...”).

References to other sections of the book that don’t include the chapter number—because, usually, they are sub-parts of a chapter—are never automatically linked. We’re setting these links manually, at present. Is there away to include them in the automatic conversion?

• The automatic paragraph numbering used in all Quiz and Quiz Answers sections doesn’t match the book (the html is all numeric). For example, the book does it this way:

  3. Question text question text question text?

     A. Answer text answer text.

     B. Answer text answer text.

     C. Answer text answer text.

     D. Answer text answer text.

But the html files do it this way:

  3. Question text question text question text?

     1. Answer text answer text.

     2. Answer text answer text.

     3. Answer text answer text.

     4. Answer text answer text.

This means that the quiz answers are always wrong, because they still indicate the alphabetic choice. This has to manually corrected. Can we add an alphabetic option to the <li> tag, to alleviate this?

• We find, in some files that are used to lead off a section (i.e, Workshop, Part 1 (title), References, etc.), that the entire contents of this file consist of the the title. Is this really what is wanted? It seems a wasted click for the user, constructed this way, and will likely appear to them to be an error. At the least, wouldn’t links to the related sub-Sections be helpful? One way to deal with this would be to add a bit of
boilerplate anywhere such titles occur alone, just to keep it from looking like a mistake. An example for the Workshop section:

**Workshop**

*You can use the Workshop to reinforce the material you’ve just covered. You’ll find some pertinent questions and their answers in Q&A. Then test your mastery of the information using the Quiz, and check how you did with the Quiz Answers.*

We can continue doing this manually, if necessary.

- Whenever a table includes a footnote immediately following it, the footnote imports badly. It is always out of place (usually right after the table header), is always set to a red font color, and always includes a few extraneous tags. This is not visible from the Dreamweaver window; only in the browser, as follows:

  `<font color = 'red'><entry>
  Source: Alcatel at frcatel utc.sk/hwb/ta_AWG.html
  </entry></font>`

  This particular example comes from Ch04 (table 4-2) in 1578701175. It appears to be a problem with the conversion routine. We’re currently locating the offending material, cutting out the text, and deleting the extraneous tags. Then we place the text where it belongs and format it properly. It’s a bit time-consuming.

- The conversion routine consistently sets some text styles backward—that is, it’s supposed to be bold, but the file has it italic. Text that’s supposed to be italic, however, usually is.
Appendix C

InformIT.com HTML Review Guidelines

Chapter hierarchy (HA, HB, HC, HD, HE, HF)

HTML allows for six heading levels, starting with <h1> (the largest) and ending at <h6> (the smallest). In the printed form, most PTG books list the HA (chapter number) and HB (chapter title) separately. However, InformIT’s specifications are slightly different.

The HA and HB should be combined onto one line of text and tagged as shown below.

Note that both the HA/HB and the HC styles are tagged as <h2>.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>9</td>
</tr>
<tr>
<td>(b)</td>
<td>Printing</td>
</tr>
<tr>
<td>(c)</td>
<td>Local Printing from Caldera</td>
</tr>
<tr>
<td>(d)</td>
<td>How Does Caldera Locate My Printer?</td>
</tr>
<tr>
<td>(e)</td>
<td>How Does Caldera Locate My Printer?</td>
</tr>
<tr>
<td>(f)</td>
<td>How Does Caldera Locate My Printer?</td>
</tr>
<tr>
<td>(g)</td>
<td>How Does Caldera Locate My Printer?</td>
</tr>
<tr>
<td></td>
<td>&lt;h2&gt;9. Printing&lt;/h2&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;h2&gt;Local Printing from Caldera&lt;/h2&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;h3&gt;How Does Caldera Locate My Printer?&lt;/h3&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;h4&gt;How Does Caldera Locate My Printer?&lt;/h4&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;h5&gt;How Does Caldera Locate My Printer?&lt;/h5&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;h6&gt;How Does Caldera Locate My Printer?&lt;/h6&gt;</td>
</tr>
</tbody>
</table>

Chapter Opener Listings

Most PTG titles use a chapter opener of some kind. In most instances, the chapter opener will be converted into HTML along with the rest of the chapter. However, if the chapter opener merely lists the C-heads contained in that chapter, it should not be converted into HTML.

Similarly, if the chapter begins with a pseudo-list of C-heads intertwined with the text, the list of C-heads should not be linked. InformIT is generating a mini-TOC for each chapter and is placing it at the top of the chapter file, thus creating redundancy.

The tagging structure for the chapter opener should look like this:

```html
<p>In This Chapter</p>
<ul>
  <li><p>Local Printing from Caldera</p></li>
  <li><p>Tools and Locations Used in Printing</p></li>
  <li><p>Understanding the Printing Process</p></li>
  <li><p>Using COAS to "Administrate" Your Printer</p></li>
  <li><p>Setting Up and Using a Remote Printer</p></li>
  <li><p>Formatted Printing</p></li>
  <li><p>Using Your Printer with Other Applications</p></li>
  <li><p>Useful Print Capability Enhancing Programs (and Where to Find Them)</p></li>
  <li><p>Some Secrets of the Guru (Junior Division)</p></li>
</ul>
```
Paragraphs

Paragraphs are simple and straight-forward. They should begin with the <p> tag and end with the </p> tag. An example is below:

<p>Linux supports many different kinds of file systems. From a user's perspective, they all look pretty similar: They all look like hierarchical trees of directories and files. However, the low-level bookkeeping details differ from system to system, as do some of the capabilities. Let's look at some file systems.</p>

Monospaced Text within Text and Headings

In text

<p>Serial devices likewise have entries in the <tt>/dev</tt> directory. Serial devices" entries will be similar to <tt>/dev/ttyS?</tt> Where the <tt>?</tt> is replaced by the serial port number (<tt>0</tt>, <tt>1</tt>, <tt>2</tt>, <tt>3</tt>, or <tt>4</tt>), and these are also character special devices.</p>

In headings

<h4><a name="ch08lev3sec5">Telling <i><tt>cfdisk</tt> About Disk Geometry</i></a></h4>

Code Listings

Code listings appear in a variety of formats throughout PTG titles. InformIT utilizes the power of hyperlinks to provide its end users with a version of the code listings that can be copied-and-pasted into an application. As a result, we will need to generate two copies of the code in those cases shown below.

A brief run-down of the common code listing occurrences follows.

**C1 listings (single lines of code)**

<pre>$ cat /proc/devices | grep lp</pre>

**Listings (C2-CX and LC2-LCX)**

<pre>$ cat /proc/ide/hda/geometry</pre>

**Code Continuation Characters**

The standard placement of the code continuation character in Que and Sams titles is following 80 characters of computer code. (At this time, I do not have a guideline for books produced at the Parkwood facility.) However, InformIT's requirement is that code lines are broken after 72 characters. What this means is that there may be instances where a code continuation character is not used in the print form, but will need to be inserted in the HTML files. Also, if one appears in print, we will need to adjust the placement in the HTML files.

The code lines should be broken near a space and not in the middle of a word. The code for the ccc.gif should be placed at the start of the second line of code and should not have any spaces before or after the image reference.
InformIT uses a JavaScript pop-up window for code listings which contain a code continuation character to provide the end user with a true code listing without the .gif image. This file should be placed in the elementLinks folder.

Example 4.1

{command used is #ls -l /dev/lp*}
crw-rw---- 1 root lp 6, 0 Apr 3 16:47 /dev/lp0
<img src="ElementLinks/ccc.gif" width="14" height="9">(Printer at address 0x3BC)
crw-rw---- 1 root lp 6, 1 Apr 3 16:47 /dev/lp1
<img src="ElementLinks/ccc.gif" width="14" height="9">(Printer at address 0x378)
crw-rw---- 1 root lp 6, 2 Apr 3 16:47 /dev/lp2
<img src="ElementLinks/ccc.gif" width="14" height="9">(Printer at address 0x278)
</pre>

See this code sample<a href="javascript:popUp('elementLinks/04code01.html')"></a></p>

Code within a Numbered List

<ol>
<li>For a quick check, enter: #ls</li>
<li>When the LISA screen comes up, select 3 (System Configuration), 2 (System Configuration), and 6 (Configure daemon/server autostart).</li>
<li>Scroll down to view Printer Server (LPD). This probably will be at number 12.</li>
<li>Verify that Printer Server (LPD) has an X in the parenthesis indicating autostart when the system boots up. If that X is present, then the spooler (line printer daemon) will start up automatically every time the system is rebooted.</li>
</ol>

Code within a Bulleted List

<ul>
<li>For a quick check, enter: #ls</li>
<li>When the LISA screen comes up, select 3 (System Configuration), 2 (System Configuration), and 6 (Configure daemon/server autostart).</li>
<li>Scroll down to view Printer Server (LPD). This probably will be at number 12.</li>
<li>Verify that Printer Server (LPD) has an X in the parenthesis indicating autostart when the system boots up. If that X is present, then the spooler (line printer daemon) will start up automatically every time the system is rebooted.</li>
</ul>

Lists

Most PTG titles will use a combination of both numbered and bulleted lists throughout the text. This section covers the common applications of these list elements.

NL - NX

<ol>
<li>Highlight the partition whose type you wish to change. Press the up arrow or down arrow until the right partition is highlighted.</li>
<li>Select Type.</li>
<li>Enter the code for the type of partition you wish to create.</li>
</ol>
BL - BX

<ul>
<li>Is the computer turned on?</li>
<li>Is Caldera OpenLinux installed?</li>
<li>Is the printer plugged in to the appropriate port?</li>
<li>Is the printer turned on?</li>
</ul>

BL - BX with more than one paragraph in a list item.

<ul>
<li>First paragraph in a list item
   Second paragraph in a list item</li>
<li>Is Caldera OpenLinux installed?</li>
<li>Is the printer plugged in to the appropriate port?</li>
<li>Is the printer turned on?</li>
</ul>

UL, UD, US, UX

<table>
<tbody>
<tr valign="top">
<td width="72"><tt>/dev/lp0</tt></td>
<td width="269">Character special device for LPR1:</td>
</tr>
<tr valign="top">
<td width="72"><tt>/dev/lp1</tt></td>
<td width="269">Character special device for LPR2:</td>
</tr>
<tr valign="top">
<td width="72"><tt>/dev/lp2</tt></td>
<td width="269">Character special device for LPR3:</td>
</tr>
</tbody>
</table>

Bulleted List inside a numbered list

<ol>
<li>Step one of many</li>
<ul>
<li>high step</li>
<li>low step</li>
</ul>
<li>The last step</li>
</ol>

Q&A Sections - Questions Only

<h3><a name="N11138">Q&amp;A</a></h3>

<p>The following are questions to test your understanding of the topics covered in this chapter. After you have answered the questions, you will find the answers in Appendix A, "Answers to Quizzes and Q&amp;As, <a href="ap01s01.html#N49867">Answers to Chapter 2 Q&A Questions</a>." If you get an answer wrong, review the answer and ensure that you understand the reason for your mistake. If you are confused by the answer, refer back to the text in the chapter to review the concepts.</p>
1. State two reasons to use an IP tunnel.

2. Will EIGRP carry SAP updates?

3. Why is it important to ensure the RIP/SAP update timers are synchronized?

4. In configuring an IP tunnel, how many IP tunnels may be created with the same source and destination address?

5. Associate the appropriate IOS feature to solve the network congestion problem experienced on the network in the following table.

**Q&A Sections - Questions and Answers**

Note that with questions AND answers, the entire question should be bold, not just the number.

1. State two reasons to use an IP tunnel.
   
   One, you need to get somewhere. Two, Wile Coyote can't use it only Road Runner can.

2. Will EIGRP carry SAP updates?
   
   One, you need to get somewhere. Two, Wile Coyote can't use it only Road Runner can.

3. Why is it important to ensure the RIP/SAP update timers are synchronized?
   
   One, you need to get somewhere. Two, Wile Coyote can't use it only Road Runner can.
4. In configuring an IP tunnel, how many IP tunnels may be created with the same source and destination address?

One, you need to get somewhere. Two, Wile Coyote can't use it only Road Runner can.

5. Associate the appropriate IOS feature to solve the network congestion problem experienced on the network in the following table.

One, you need to get somewhere. Two, Wile Coyote can't use it only Road Runner can.

### Tables

Tables should always have a border of 1. They should not have any cellpadding or cellspacing applied, not even 0. Tables should not have any width set on either the table as a whole or on any column of the table. Table Rows should be vertically aligned. There should be no horizontal alignment on any row or column or cell. The "header" columns of each table should not be bold, italic or center aligned. Setting these off will be accomplished by a stylesheet applied to the <th> tag. Please adhere strictly to these guidelines regardless of the styles indicated in the printed book.

### TH, TC, TS, TB, TX

Table Reference in text

Table 8.2 shows how various character devices are named.

Table

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev/ttyS0</td>
<td>The first serial port (COM1)</td>
</tr>
<tr>
<td>/dev/ttyS1</td>
<td>The second serial port (COM2)</td>
</tr>
<tr>
<td>/dev/lp0</td>
<td>The first parallel port</td>
</tr>
<tr>
<td>/dev/mouse</td>
<td>Usually a symbolic link to the mouse device, which is often /dev/psaux for a PS/2 mouse or one of the serial ports for a serial mouse.</td>
</tr>
</tbody>
</table>
Table footnote
All footnotes should be numbered as they appear in the material, including footnotes appearing in tables.

<p>Table 8.2 shows how various character devices are named.</p>

<h4>Table 8.2. Character Devices</h4>
<table border="1">
<thead>
<tr valign="top">
<th width="156">Device Name</th>
<th width="539">Meaning</th>
</tr>
</thead>
<tbody>
<tr valign="top">
<td width="156"><tt>Dev/ttyS0</tt></td>
<td width="539">The first serial port (COM1)</td>
</tr>
<tr valign="top">
<td width="156"><tt>/dev/ttyS1</tt></td>
<td width="539">The second serial port (COM2)</td>
</tr>
<tr valign="top">
<td width="156"><tt>/dev/lp0</tt></td>
<td width="539">The first parallel port</td>
</tr>
<tr valign="top">
<td width="156"><tt>/dev/mouse</tt></td>
<td width="539">Usually a symbolic link to the "real" mouse device, which is often <tt>/dev/psaux</tt> for a PS/2 mouse or one of the serial ports for a serial mouse.</td>
</tr>
</tbody>
</table>

Note, Tip Caution, Warning Sidebar, Margin Note and Other Boxes

<NO>
<p>Consistency is very important here. If you lie to the computer about where to find the printer, it will believe you!</p>
</NO>

<CA>
<p>Consistency is very important here. If you lie to the computer about where to find the printer, it will believe you!</p>
</CA>
TI
<p class="tiptitle"><a name="N34166">TIP</a></p>
<p>Consistency is very important here. If you lie to the computer about where to find the printer, it will believe you!</p>

SB
<p class="normaltitle"><a name="N30228">Background Material</a></p>
<p>Note the rather restrictive permissions on the device. The permissions on a device file control access to the device driver in exactly the same way regular file permissions control access to a file.</p>

WA
<p class="warningstitle"><a name="N34166">WARNING</a></p>
<p>Consistency is very important here. If you lie to the computer about where to find the printer, it will believe you!</p>

New Term
<p class="normaltitle">NEW TERM</p>
<p>The end-of-file, or &lt;span class='newterm'&gt;EOF&lt;/span&gt;, character is used to close the file and is entered by holding down the Ctrl key and pressing the D key. Add a new feature to this simple editor by using the &lt;tt&gt;&lt;&lt;/tt&gt;, or &lt;i&gt;here&lt;/i&gt;, redirection operator which tells the shell that the EOF character is the character that immediately follows the &lt;tt&gt;&lt;&lt;/tt&gt; operator.</p>

Peter's Principle
<p class="normaltitle">Peter's Principle</p>
<p>Consistency is very important here. If you lie to the computer about where to find the printer, it will believe you!</p>

Notes in a numbered list
<ol>
<li>Is the computer turned on?</li>
<li>Is Caldera OpenLinux installed?</li>
<li>NOTE</li>
<li>Consistency is very important here. If you lie to the computer about where to find the printer, it will believe you!</li>
<li>Is the printer plugged in to the appropriate port?</li>
<li>Is the printer turned on?</li>
</ol>
Notes in a bulleted list

- Is the computer turned on?
- Is Caldera OpenLinux installed?
- Consistency is very important here. If you lie to the computer about where to find the printer, it will believe you!
- Is the printer plugged in to the appropriate port?
- Is the printer turned on?

Notes with More than One Paragraph

Consistency is very important here. If you lie to the computer about where to find the printer, it will believe you!
The permissions on a device file control access to the device driver in exactly the same way regular file permissions control access to a file.

Notes with Code In Them

```
#cat /proc/devices | grep lp
```

Figures and Figure Links

Figure references in text

Select a printer driver at the prompt. (See Figure 9.1.)

Figure Captions and Linking

FN, FC

Figure Captions in Bulleted and Numbered lists
<li> <p>Start this procedure again by entering: </p> 
</li>

<pre class="programlisting">#lisa</pre>

<li> <p>When the LISA screen comes up, select 3 (System Configuration), 1 (Hardware Configuration), and 6 (Configure Printer). </p> 
</li>

<li> <p> Select a printer driver at the prompt. (See <a href="javascript:popUp('elementLinks/09fig01.gif')">Figure 9.1.</a>)</p> 
</li>

<p><b><a name="ch09fig1" href="javascript:popUp('elementLinks/09fig01.gif')">Figure 9.1.</a> The Select Printer Driver dialog box.</b></p>

<script language="JavaScript">
<!--
window.open(pPage,/popWin','resizable=yes,scrollbars=yes,width=550,height=420,toolbar=no');
// -->
</script>

<p>Equations</p>

Although not very common in Que and Sams titles, equations are commonly used in the higher-level titles published by imprints such as Cisco Press and MTP. The equation can be presented either as a graphic or a built equation.

If displayed as a graphic

<p><img src="ch03eq01.gif"></p>

If displayed as text

<p><a name="ch004for1"><b>Formula 4.1</b></a></p>

<tt>c = m + g</tt><sup>*</sup><tt>V</tt><sup>0.5</sup>

<p>Button Icons</p>

Button icons should remain at the size in which they appear in the text (usually either 1p or 1p6 tall by proportional).

<p>Icons in a Table</p>

<h4><a name="ch08table2"><b>Table 8.2. Examples of Word 2000 toolbar buttons.</b></a></h4>
<table border="1">
<thead>
<tr valign="top">
<th><p>Button</p></th>
<th><p>Description</p></th>
</tr>
</thead>
<tbody>
<tr valign="top">
<td><p><img src="ic01.gif"></p></td>
<td><p>The New Blank Document Button</p></td>
</tr>
<tr valign="top">
<td><p><img src="ic02.gif"></p></td>
<td><p>The Open document button</p></td>
</tr>
<tr valign="top">
<td><p><img src="ic03.gif"></p></td>
<td><p>The Save button</p></td>
</tr>
<tr valign="top">
<td><p><img src="ic04.gif"></p></td>
<td><p>The Print button</p></td>
</tr>
</tbody>
</table>

Icons in text

<p><img src="ic05.gif"> Click on the print button to print your document.</p>

Hyperlinks

Cross references/Internal Links

Occasionally there may be internal references to a chapter that was not requested by InformIT. Since they may request the referenced chapter in the future, please go ahead and create the hyperlink to the other chapter.

<p>Now that you have configured the spooler to start automatically, you can move on to creating a printer that the spooler can work with. Note that Caldera OpenLinux makes several print spoolers available. For information on these options, refer to the section entitled "Tools and Locations Used in Printing." </p>

Q&A Sections - Linking to Answers in another file

<h3><a name="N11138">Q&amp;A</a></h3>

The following are questions to test your understanding of the topics covered in this chapter. After you have answered the questions, you will find the answers in Appendix A, "Answers to Quizzes and Q&amp;As," in the file "ap01s01.html." If you get an answer wrong, review the answer and ensure that you understand the reason for your mistake. If you are confused by the answer, refer back to the text in the chapter to review the concepts.

<p>1. State two reasons to use an IP tunnel.</p>
<div class="qandaentry">
<div class="question">
<p><b>2. </b>Will EIGRP carry SAP updates?</p>
</div>
</div>

<div class="qandaentry">
<div class="question">
<p><b>3. </b>Why is it important to ensure the RIP/SAP update timers are synchronized?</p>
</div>
</div>

<div class="qandaentry">
<div class="question">
<p><b>4. </b>In configuring an IP tunnel, how many IP tunnels may be created with the same source and destination address?</p>
</div>
</div>

<div class="qandaentry">
<div class="question">
<p><b>5. </b>Associate the appropriate IOS feature to solve the network congestion problem experienced on the network in the following table.</p>
</div>
</div>

**URLs/External Links**

External links should not be formatted as `<tt>`/`<tt>`, bolded or italicized. Any hyperlink which is either dead or requires the end user to supply a password to access the site should not be active links and should be noted on the link report when the files are sent to InformIT.

**HTTP**

```
<a name="N35336"></a> <a href="http://www.im.lcs.mit.edu/magnus/nenscript/">
  http://www.im.lcs.mit.edu/magnus/nenscript/</a>
```

**Anonymous FTP**

```
```

**Non-Anonymous FTP**

```
<a href="ftp://username:password@ftp.foo.com/home/foo/homepage.html">
  ftp://username:password@ftp.foo.com/home/foo/homepage.html</a>
```

**File**

```
```

```
<a href="file:///dir1/dir2/file">file</a>
```

```
<a href="file://localhost/dir1/dir2/file">file</a>
```

**Mailto**

```
<a href="mailto:steve.pool@macmillanusa.com">steve.pool@macmillanusa.com</a>
```

**Gopher**

```
<a href="gopher://gopher.myhost.com/">gopher.myhost.com</a>
```

**Usenet**

```
<a href="news:comp.databases.sybase">comp.databases.sybase</a>
```
**Special Characters and ISO Entities**

The most common special character encountered in PTG books is the em dash (—). Because of this, an example of how to code the character is provided.

**em dash**

```html
<p>Under Linux, you can use either fdisk (a command-line oriented program) or cfdisk (a menu-driven program) to partition disk drives. I recommend cfdisk because it's much simpler; the menu-driven interface is easier to use than the command-line fdisk.</p>
```

**ISO Characters**

It is recommended that you use numeric codes from The ISO8859-1 Table of Special Characters. Character codes are not always correctly interpreted by some browsers.

The ISO8859-1 Table of Special Characters is presented separately.

**Special Elements**

At this time, InformIT is not using content from any special element (IFC, IBC, tearcard, insert, etc.). If their needs change, the spec guide will be modified.
## Appendix D

### The ISO8859-1 Table of Special Characters

<table>
<thead>
<tr>
<th>Description</th>
<th>Character</th>
<th>Numeric Code</th>
<th>Character Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quotation mark</td>
<td>&quot;</td>
<td>&amp;34;</td>
<td>&quot;</td>
</tr>
<tr>
<td>Ampersand</td>
<td>&amp;</td>
<td>&amp;38;</td>
<td>&amp;</td>
</tr>
<tr>
<td>Less-than sign</td>
<td>&lt;</td>
<td>&amp;60;</td>
<td>&lt;</td>
</tr>
<tr>
<td>Greater-than sign</td>
<td>&gt;</td>
<td>&amp;62;</td>
<td>&gt;</td>
</tr>
<tr>
<td>Non-breaking space</td>
<td> </td>
<td>#160;</td>
<td></td>
</tr>
<tr>
<td>Inverted exclamation</td>
<td>i</td>
<td>#161;</td>
<td>¡</td>
</tr>
<tr>
<td>Cent sign</td>
<td>¢</td>
<td>#162;</td>
<td>¢</td>
</tr>
<tr>
<td>Pound sterling</td>
<td>£</td>
<td>#163;</td>
<td>£</td>
</tr>
<tr>
<td>General currency sign</td>
<td>?</td>
<td>#164;</td>
<td>¤</td>
</tr>
<tr>
<td>Yen sign</td>
<td>¥</td>
<td>#165;</td>
<td>¥</td>
</tr>
<tr>
<td>Broken vertical bar</td>
<td></td>
<td></td>
<td>¦</td>
</tr>
<tr>
<td>Section sign</td>
<td>§</td>
<td>#167;</td>
<td>§</td>
</tr>
<tr>
<td>Dieresis</td>
<td>”</td>
<td>#168;</td>
<td>¨</td>
</tr>
<tr>
<td>Copyright</td>
<td>©</td>
<td>#169;</td>
<td>©</td>
</tr>
<tr>
<td>Feminine ordinal</td>
<td>ª</td>
<td>#170;</td>
<td>ª</td>
</tr>
<tr>
<td>Left angle quote</td>
<td>«</td>
<td>#171;</td>
<td>«</td>
</tr>
<tr>
<td>Not sign</td>
<td>¬</td>
<td>#172;</td>
<td>¬</td>
</tr>
<tr>
<td>Soft hyphen</td>
<td>-</td>
<td>#173;</td>
<td>­</td>
</tr>
<tr>
<td>Registered trademark</td>
<td>®</td>
<td>#174;</td>
<td>®</td>
</tr>
<tr>
<td>Macron accent</td>
<td>¯</td>
<td>#175;</td>
<td>¯</td>
</tr>
<tr>
<td>Degree sign</td>
<td>°</td>
<td>#176;</td>
<td>°</td>
</tr>
<tr>
<td>Plus or minus</td>
<td>±</td>
<td>#177;</td>
<td>±</td>
</tr>
<tr>
<td>Superscript two</td>
<td>²</td>
<td>#178;</td>
<td>²</td>
</tr>
<tr>
<td>Superscript three</td>
<td>³</td>
<td>#179;</td>
<td>³</td>
</tr>
<tr>
<td>Acute accent</td>
<td>´</td>
<td>#180;</td>
<td>´</td>
</tr>
<tr>
<td>Micro sign</td>
<td>µ</td>
<td>#181;</td>
<td>µ</td>
</tr>
<tr>
<td>Paragraph sign</td>
<td>¶</td>
<td>#182;</td>
<td>¶</td>
</tr>
<tr>
<td>Middle dot</td>
<td>·</td>
<td>#183;</td>
<td>·</td>
</tr>
<tr>
<td>Cedilla</td>
<td>¸</td>
<td>#184;</td>
<td>&amp;cédil;</td>
</tr>
<tr>
<td>Superscript one</td>
<td>¹</td>
<td>#185;</td>
<td>¹</td>
</tr>
<tr>
<td>Masculine ordinal</td>
<td>°</td>
<td>#186;</td>
<td>º</td>
</tr>
<tr>
<td>Right angle quote</td>
<td>»</td>
<td>#187;</td>
<td>»</td>
</tr>
<tr>
<td>Fraction one-fourth</td>
<td>¼</td>
<td>#188;</td>
<td>¼</td>
</tr>
<tr>
<td>Fraction one-half</td>
<td>½</td>
<td>#189;</td>
<td>½</td>
</tr>
<tr>
<td>Fraction three-fourths</td>
<td>¾</td>
<td>#190;</td>
<td>¾</td>
</tr>
<tr>
<td>Inverted question mark</td>
<td>¿</td>
<td>#191;</td>
<td>¿</td>
</tr>
<tr>
<td>Capital A, grave accent</td>
<td>À</td>
<td>#192;</td>
<td>À</td>
</tr>
<tr>
<td>Capital A, acute accent</td>
<td>Â</td>
<td>#193;</td>
<td>&amp;Acute;</td>
</tr>
<tr>
<td>Capital A, circumflex accent</td>
<td>À</td>
<td>#194;</td>
<td>Â</td>
</tr>
<tr>
<td>Capital A, tilde</td>
<td>À</td>
<td>#195;</td>
<td>&amp;Atild;</td>
</tr>
<tr>
<td>Capital A, dieresis</td>
<td>À</td>
<td>#196;</td>
<td>Ä</td>
</tr>
<tr>
<td>Capital A, ring</td>
<td>À</td>
<td>#197;</td>
<td>Å</td>
</tr>
<tr>
<td>Capital AE ligature</td>
<td>À</td>
<td>#198;</td>
<td>Æ</td>
</tr>
<tr>
<td>Symbol</td>
<td>Unicode</td>
<td>HTML Entity Code</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>Capital C, cedilla</td>
<td>Ç</td>
<td>Ç Ç</td>
<td></td>
</tr>
<tr>
<td>Capital E, grave accent</td>
<td>É</td>
<td>È È</td>
<td></td>
</tr>
<tr>
<td>Capital E, acute accent</td>
<td>È</td>
<td>É É</td>
<td></td>
</tr>
<tr>
<td>Capital E, circumflex accent</td>
<td>Ê</td>
<td>Ê Ê</td>
<td></td>
</tr>
<tr>
<td>Capital E, dieresis</td>
<td>Ë</td>
<td>Ë Ë</td>
<td></td>
</tr>
<tr>
<td>Capital I, grave accent</td>
<td>Í</td>
<td>Ì Ì</td>
<td></td>
</tr>
<tr>
<td>Capital I, acute accent</td>
<td>Í</td>
<td>Í Í</td>
<td></td>
</tr>
<tr>
<td>Capital I, circumflex accent</td>
<td>Í</td>
<td>Î Î</td>
<td></td>
</tr>
<tr>
<td>Capital I, dieresis</td>
<td>Í</td>
<td>Ï Ï</td>
<td></td>
</tr>
<tr>
<td>Capital Eth, Icelandic</td>
<td>&lt;ETH&gt;</td>
<td>Ð Ð</td>
<td></td>
</tr>
<tr>
<td>Capital N, tilde</td>
<td>N</td>
<td>&amp;209; Ñ</td>
<td></td>
</tr>
<tr>
<td>Capital O, grave accent</td>
<td>Ó</td>
<td>Ò Ò</td>
<td></td>
</tr>
<tr>
<td>Capital O, acute accent</td>
<td>Ó</td>
<td>Ó Ó</td>
<td></td>
</tr>
<tr>
<td>Capital O, circumflex accent</td>
<td>Ó</td>
<td>Ô Ô</td>
<td></td>
</tr>
<tr>
<td>Capital O, dieresis</td>
<td>Ó</td>
<td>Õ Ö</td>
<td></td>
</tr>
<tr>
<td>Capital O, tilde</td>
<td>Ó</td>
<td>&amp;214; Ö</td>
<td></td>
</tr>
<tr>
<td>Multiply sign</td>
<td>¥</td>
<td>&amp;215; ×</td>
<td></td>
</tr>
<tr>
<td>Capital O, slash</td>
<td>Ø</td>
<td>&amp;216; Ø</td>
<td></td>
</tr>
<tr>
<td>Capital U, grave accent</td>
<td>Ù</td>
<td>&amp;217; Ù</td>
<td></td>
</tr>
<tr>
<td>Capital U, acute accent</td>
<td>Ù</td>
<td>&amp;218; Ú</td>
<td></td>
</tr>
<tr>
<td>Capital U, circumflex accent</td>
<td>Ù</td>
<td>&amp;219; Û</td>
<td></td>
</tr>
<tr>
<td>Capital U, dieresis</td>
<td>Ù</td>
<td>&amp;220; Ü</td>
<td></td>
</tr>
<tr>
<td>Capital Y, acute accent</td>
<td>Ý</td>
<td>&amp;221; Ý</td>
<td></td>
</tr>
<tr>
<td>Capital THORN, Icelandic</td>
<td>&lt;THORN&gt;</td>
<td>&amp;222; Þ</td>
<td></td>
</tr>
<tr>
<td>Small sharp s</td>
<td>ß</td>
<td>&amp;223; ß</td>
<td></td>
</tr>
<tr>
<td>Small a, grave accent</td>
<td>à</td>
<td>&amp;224; à</td>
<td></td>
</tr>
<tr>
<td>Small a, acute accent</td>
<td>á</td>
<td>&amp;225; á</td>
<td></td>
</tr>
<tr>
<td>Small a, circumflex accent</td>
<td>â</td>
<td>&amp;226; â</td>
<td></td>
</tr>
<tr>
<td>Small a, tilde</td>
<td>ã</td>
<td>&amp;227; ã</td>
<td></td>
</tr>
<tr>
<td>Small a, dieresis</td>
<td>ä</td>
<td>&amp;228; ä</td>
<td></td>
</tr>
<tr>
<td>Small a, ring</td>
<td>å</td>
<td>&amp;229; å</td>
<td></td>
</tr>
<tr>
<td>Small ae ligature</td>
<td>æ</td>
<td>&amp;230; æ</td>
<td></td>
</tr>
<tr>
<td>Small c, cedilla</td>
<td>ç</td>
<td>&amp;231; ç</td>
<td></td>
</tr>
<tr>
<td>Small e, grave accent</td>
<td>è</td>
<td>&amp;232; è</td>
<td></td>
</tr>
<tr>
<td>Small e, acute accent</td>
<td>é</td>
<td>&amp;233; é</td>
<td></td>
</tr>
<tr>
<td>Small e, circumflex accent</td>
<td>ê</td>
<td>&amp;234; ê</td>
<td></td>
</tr>
<tr>
<td>Small e, dieresis</td>
<td>ë</td>
<td>&amp;235; ë</td>
<td></td>
</tr>
<tr>
<td>Small i, grave accent</td>
<td>í</td>
<td>&amp;236; ì</td>
<td></td>
</tr>
<tr>
<td>Small i, acute accent</td>
<td>í</td>
<td>&amp;237; í</td>
<td></td>
</tr>
<tr>
<td>Small i, circumflex accent</td>
<td>ì</td>
<td>&amp;238; î</td>
<td></td>
</tr>
<tr>
<td>Small i, dieresis</td>
<td>í</td>
<td>&amp;239; ï</td>
<td></td>
</tr>
<tr>
<td>Small eth, Icelandic</td>
<td>&lt;eth&gt;</td>
<td>&amp;240; ð</td>
<td></td>
</tr>
<tr>
<td>Small n, tilde</td>
<td>ñ</td>
<td>&amp;241; ñ</td>
<td></td>
</tr>
<tr>
<td>Small o, grave accent</td>
<td>ò</td>
<td>&amp;242; ò</td>
<td></td>
</tr>
<tr>
<td>Small o, acute accent</td>
<td>ó</td>
<td>&amp;243; ó</td>
<td></td>
</tr>
<tr>
<td>Small o, circumflex accent</td>
<td>ô</td>
<td>&amp;244; ô</td>
<td></td>
</tr>
<tr>
<td>Small o, tilde</td>
<td>õ</td>
<td>&amp;245; õ</td>
<td></td>
</tr>
<tr>
<td>Small o, dieresis</td>
<td>õ</td>
<td>&amp;246; ö</td>
<td></td>
</tr>
<tr>
<td>Division sign</td>
<td>÷</td>
<td>&amp;247; ÷</td>
<td></td>
</tr>
<tr>
<td>Small o, slash</td>
<td>ø</td>
<td>&amp;248; ø</td>
<td></td>
</tr>
<tr>
<td>Small u, grave accent</td>
<td>ù</td>
<td>&amp;249; ù</td>
<td></td>
</tr>
<tr>
<td>Small u, acute accent</td>
<td>ú</td>
<td>&amp;250; ú</td>
<td></td>
</tr>
<tr>
<td>Small u, circumflex accent</td>
<td>û</td>
<td>&amp;251; û</td>
<td></td>
</tr>
<tr>
<td>Small u, dieresis</td>
<td>ü</td>
<td>ü</td>
<td>ü</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>Small y, acute accent</td>
<td>y´</td>
<td>ý</td>
<td>ý</td>
</tr>
<tr>
<td>Small thorn, Icelandic</td>
<td>&lt;thorn&gt;</td>
<td>þ</td>
<td>þ</td>
</tr>
<tr>
<td>Small y, dieresis</td>
<td>ý</td>
<td>ÿ</td>
<td>ÿ</td>
</tr>
</tbody>
</table>
# Appendix E

## Link Check Report

0789721252

<table>
<thead>
<tr>
<th>Broken Links:</th>
<th>Link</th>
<th>Link Checked</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>done-C-head/pt01ch04s04.html</td>
<td><a href="http://www.powerquest.com">www.powerquest.com</a></td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt01ch04s10.html</td>
<td>troubleshootingicon.gif</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt01ch04s10.html</td>
<td>#recoveryconsole</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt01ch04s10.html</td>
<td>troubleshootingicon.gif</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt01ch04s10.html</td>
<td>troubleshootingicon.gif</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt02ch08s01.html</td>
<td>troubleshootingicon.gif</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt02ch08s01.html</td>
<td>troubleshootingicon.gif</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt02ch08s04.html</td>
<td>troubleshootingicon.gif</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt02ch08s04.html</td>
<td>troubleshootingicon.gif</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt03ch14s02.html</td>
<td><a href="http://www.deja.com">www.deja.com</a></td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt03ch14s02.html</td>
<td>troubleshootingicon.gif</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt03ch14s03.html</td>
<td>troubleshootingicon.gif</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>done-C-head/pt03ch14s03.html</td>
<td>troubleshootingicon.gif</td>
<td>Yes</td>
<td>Fixed</td>
</tr>
<tr>
<td>External Links:</td>
<td>Link</td>
<td>Link Checked</td>
<td>Comments</td>
</tr>
<tr>
<td>----------------</td>
<td>------</td>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>done-C-head/pt01ch04s04.html</td>
<td>javascript:popUp('element Links/04fig01.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt01ch04s10.html</td>
<td>javascript:popUp('element Links/04fig02.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt01ch04s11.html</td>
<td>javascript:popUp('element Links/04fig03.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt01ch04s11.html</td>
<td>javascript:popUp('element Links/04fig04.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt01ch04s11.html</td>
<td>javascript:popUp('element Links/04fig05.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt01ch04s15.html</td>
<td>javascript:popUp('element Links/04fig06.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt01ch04s15.html</td>
<td>javascript:popUp('element Links/04fig07.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s01.html</td>
<td>javascript:popUp('element Links/08fig01.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s01.html</td>
<td>javascript:popUp('element Links/08fig02.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s02.html</td>
<td>javascript:popUp('element Links/08fig03.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s03.html</td>
<td>javascript:popUp('element Links/08fig04.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s03.html</td>
<td>javascript:popUp('element Links/08fig05.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s03.html</td>
<td>javascript:popUp('element Links/08fig06.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s03.html</td>
<td>javascript:popUp('element Links/08fig07.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s03.html</td>
<td>javascript:popUp('element Links/08fig08.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s04.html</td>
<td>javascript:popUp('element Links/08fig09.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s04.html</td>
<td>javascript:popUp('element Links/08fig10.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s04.html</td>
<td>javascript:popUp('element Links/08fig11.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s04.html</td>
<td>javascript:popUp('element Links/08fig12.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s04.html</td>
<td>javascript:popUp('element Links/08fig13.gif')</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>done-C-head/pt02ch08s06.html</td>
<td>javascript:popUp('element Links/08fig14.gif')</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
Appendix F

Editing and Shipping List
HTML Check and Shipping List

ISBN: ___________________________  Date: ___________________________

Title: ___________________________

Unsortoc.xml

☐ I have included the unsortoc.xml file

JavaScript

☐ I have verified that I have included the correct JavaScript for this conversion.

Cross-references

☐ All cross-references for this conversion have been properly linked.

Graphics

☐ N/A

☐ All graphics are complete, clear and the right size.

☐ All graphics are linked and have a javascript:popUp applied.

☐ All graphics are contained in the elementLinks directory.

☐ All graphics are correctly numbered.

Icons

☐ N/A

☐ I have verified if the book contains icons they are properly formatted, correctly sized and clear.

Tables

Tables should be formatted per the guidelines NOT by the book style

I have checked tables for:

☐ Correct formatting

☐ All Table heads are Bold

☐ All Table heads are left aligned

☐ Correct number of columns

☐ All text is included

☐ All table footnotes have been converted to a javascript popup. ☐ N/A

☐ If the table contains icons, they are correctly sized and included. ☐ N/A
Cross-references to tables have been linked.

Special Characters

I have searched and replaced all special characters with the proper ISO character.

(IE -- to &#151; …)

Listings

N/A

Cross-references to listings have been linked.

Code lines have the correct indentation and spacing.

Code continuation icons are correctly placed in long code lines.

I have verified that all bold, italic and bolditalic fonts are correctly formatted in the listing.

I have verified all code and code listings are correctly tagged as <pre class="programlisting">…</pre>

I have removed extra <br> tags in code listings and any unnecessary blank lines.

Footnotes

N/A

All footnotes have been correctly formatted as a JavaScript popup.

Text

I have verified that all text is present for this conversion.

Mono text

I have ensured all mono text is correctly tagged in the content with <tt> tags.

Bold and Italic text

I have ensured all bold, italic and bolditalic text is correctly tagged in the content.

Links

All links for URLs Graphics, Tables, Listings and Cross-references use double quotes instead of single quotes.

Notes, Tips, Cautions, Warnings Boxes…

I have verified all note boxes are correctly formatted and use the correct headers.

Link Checklist Form

I have completed and included the Link Checklist form in this submission.

I have checked all URLs to ensure they are correctly formatted and go to an active page. N/A

I have contacted InformIT about any bad URLs I have encountered. N/A