

NTIA Report 01-385

On-line Development of an Interactive Web Glossary

**William J. Ingram
Evelyn M. Gray**



**U.S. DEPARTMENT OF COMMERCE
Donald L. Evans, Secretary**

John F. Sopko, Acting Assistant Secretary
for Communications and Information

April 2001

This Page Intentionally Left Blank

This Page Intentionally Left Blank

PREFACE

Certain commercial equipment, instruments, or materials are identified in this report to specify adequately the experimental procedure. In no case does such identification imply recommendation or endorsement by the National Telecommunications and Information Administration, nor does it imply that the material or equipment identified is necessarily the best available for the purpose.

This Page Intentionally Left Blank

This Page Intentionally Left Blank

CONTENTS

	page
LIST OF FIGURES	vi
LIST OF TABLES	vi
ABSTRACT	1
1. INTRODUCTION	1
1.1 Purpose of the Report	1
1.2 Virtual Meetings	2
1.3 Purpose of <i>Telecom Glossary 2000</i>	5
1.4 Expected On-Line or Internet Audience for <i>Telecom Glossary 2000</i>	5
1.5 Participants in the Development of <i>Telecom Glossary 2000</i>	6
2. CONTENT OF <i>TELECOM GLOSSARY 2000</i>	7
2.1 Disciplines Addressed in the Old and the New Glossaries	7
2.2 New Material Added to Develop <i>Telecom Glossary 2000</i>	7
2.3 Features of the Finished Glossary	8
3. BENEFITS OF ELECTRONIC UPDATING	14
3.1 Previous Updating Techniques	14
3.2 Summary of On-line Updating Techniques for <i>Telecom Glossary 2000</i>	15
3.3 Advantages of Web-based and E-mail-based Procedures	15
4. ON-LINE PROCEDURES FOR UPDATING THE GLOSSARY	18
4.1 Overview of Updating Procedures	18
4.2 Four Major Steps in the Updating Process	19
4.3 Review of the Draft on the Web	20
4.4 Web-Page Compliance with Disabilities-Access Guidelines	28
5. PROGRAMMING	29
5.1 An Overview of the Processing	29
5.2 E-mail Software	30
5.3 Interactive Editing of the Glossary	31
5.4 Database Management Software	32
5.5 Interactive Web Pages	35
5.6 Daily/Nightly Scripts	51
5.7 Files	82
6. EXECUTION	85

7. RESULTS AND CONCLUSIONS	85
8. REFERENCES	86
APPENDIX A: HISTORY OF THE GLOSSARY	87
APPENDIX B: REFERENCES FOR <i>TELECOM GLOSSARY 2000</i>	90

LIST OF FIGURES

	page
Figure 1. Example of a definition from the glossary	10
Figure 2. Example of a two-dimensional image accessible in the glossary	11
Figure 3. Example of access to a video clip in the glossary	12
Figure 4. The visual indication of events following a click on an “audio” icon	12
Figure 5. Committee method for earlier (1996) glossary update	14
Figure 6. Conceptual map of electronic processes of developing <i>Telecom Glossary 2000</i> .	18
Figure 7. <i>Telecom Glossary 2000</i> , “Foreword” page as it appeared (draft form) for Web editing	20
Figure 8. The “Main” page of the Web-based glossary, as of November 2000	21
Figure 9. The several hot spots and hyperlinks on the initial glossary Web page	22
Figure 10. The essence of the text for the “[Intro]” and “Welcome” hyperlinks	24
Figure 11. Essential contents of the “Help” button	25
Figure 12. The configuration of frames used in the glossary Web pages	26
Figure 13. Flowchart of processes for adding to the glossary and grooming it	29
Figure 14. Flowchart of the major tasks in the program called “whole”	32
Figure 15. The relationship of anchors and hot spots	62
Figure 16. The number of times that some definitions were linked	72
Figure A-1. History of the development of the ANS T1.523-2001 glossary	88

LIST OF TABLES

	page
Table 1. The Six Technical Subcommittees (TSCs) of T1	3
Table 2. Membership of the <i>Telecom Glossary 2000</i> Ad Hoc Group	6
Table 3. A Summary of the Major Functions of Each Program in the “Whole” Program ...	33
Table 4. The Top 10 Most Linked Terms	73
Table B-1. Key to Literature Source Citations for <i>Telecom Glossary 2000</i> References	91
Table B-2. Numerical List of ANSI T1 Standards Cited in the Glossary	94
Table B-3. Committee T1 Technical Reports Cited in the Glossary	99
Table B-4. Other References Cited in the Glossary	101

ON-LINE DEVELOPMENT OF AN INTERACTIVE WEB GLOSSARY

William J. Ingram and Evelyn M. Gray¹

This report describes the electronic, computer-based processes of creating an on-line, Web-based glossary of telecommunication terms—American National Standard, T1.523-2001, *Telecom Glossary 2000*. These e-mail processes or Web-based processes constituted electronic meetings or “virtual meetings” to solicit public review comments for updating the glossary. The programming design, structure, and some of the code used to gather e-mail comments and to build the glossary Web pages are discussed in the report. With the exception of the kickoff meeting and the resolution meeting, all interim meetings of the glossary-development Ad Hoc Group consisted of e-mail exchanges or Web-page modifications. This electronic meeting forum allowed a broader participation in the revisions, wider distribution of comments, speedier development of the glossary, and economy of resources. The completed product is scheduled to be available free of charge to the public on the ATIS (Alliance for Telecommunications Industry Solutions)/T1Web site.

Key words: computer communications; electronic publishing; e-mail; HTML; hypertext; Internet; programming scripts; search engine; telecommunications glossary; virtual meeting; Web

1. INTRODUCTION

1.1 Purpose of the Report

This purpose of this report is to describe the on-line, electronic-meeting processes of achieving consensus to update FED-STD-1037C to become an American National Standard, ANS T1.523-2001, *Telecom Glossary 2000*. In writing this report, the authors have documented the due process followed in updating the glossary using strictly electronic means—rather than standard, face-to-face committee meetings. This report shows that use of e-mail and “virtual meetings” has advantages over conventional, face-to-face committee meetings, in that virtual meetings are

- more timely,
- more suitable in addressing rapidly changing technology,
- more economical/conservative of resources,
- more visible,
- more accessible to a wider audience, and
- more accurate in recording immediately the consensus of the conferees.

¹ The authors are with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, CO 80305-3328.

These advantages, discussed in more detail in Section 3 below, can be reaped by developers of many other on-line products that require timely updates to keep them valid and usable.

The previous editions of the popular glossary (which were developed in conventional, face-to-face meetings) span 25 years, as is shown in Appendix A. In developing previous editions, from 1974 to 2001, ITS has held a leadership role (as either senior editor/coordinator or as subcommittee chair) in developing and updating the entire Federal Standard 1037 series, *Glossary of Telecommunication Terms*, and its predecessor documents. The full edition of the old FED-STD-1037C (1996) is currently available on the Internet, where it has received more than 500,000 hits since August 1996, at the Web address below:

<http://glossary.its.bldrdoc.gov/fs-1037/> .

1.2 Virtual Meetings

The popular FED-STD-1037C was updated to *Telecom Glossary 2000* using exclusively electronic means and essentially no face-to-face meetings—an industry first. The details of how this work was done and the details of the programming design to accomplish the work are given in sections 4 and 5 below.

To facilitate the virtual meetings, the ITS editors created interactive Web pages and e-mail communications paths to unite the geographically dispersed committee of experts and to collect their definitions, reviews, and technical comments in an efficient, timely manner. The resulting glossary, the new American National Standard (ANS) T1.523-2001, *Telecom Glossary 2000*, is slated to be available free of charge to the public on the ATIS (Alliance for Telecommunications Industry Solutions) T1 Web site as well as on the ITS Web site

<http://www.its.bldrdoc.gov/projects/telecomglossary2000> .

Since it was first put on the Web in mid-2000 at the Web site above, the new glossary has received more than 35,000 hits, even without formal announcements or press releases. At this Web site and at the proposed ATIS/T1 Web site, the glossary contains a fully functioning search engine, and all of the features described in Section 2 below.

With foresight, the glossary's sponsors recognized the need to update the 1996 edition to advance the interoperability, reliability, and operational effectiveness of telecommunications equipment and services by standardizing telecommunications terminology used in technical specifications. They particularly wished to accommodate new Web/Internet terminology, and to address security and infrastructure-protection terms. The sponsors and editors, as members of the Federal Government, remain committed to the use of voluntary industry standards whenever such standards exist or can be developed. Further, they wished the revised glossary to retain the high status of a standard.

The ITS editors spoke with representatives of three different standards-development organizations, to select one SDO for release and promulgation of the new standard. The editors selected the T1 standards community for co-hosting the work of revising the glossary. The advantages of selecting this group were that ITS has some leadership responsibilities in the T1A1 TSC (Technical Subcommittee) on Performance and Signal Processing and, therefore, can provide an energizing force and a useful level of control in guiding the revision of the document. The contributions of Committee T1 participants expanded the revised standard's content and consensus base, and its publication as a T1-developed American National Standard maximized its benefit to the U.S. telecommunications industry. The six TSCs in T1 and their related fields of standards responsibilities are shown in Table 1 below.

Table 1. The Six Technical Subcommittees (TSCs) of T1

T1A1	Performance and Signal Processing
T1E1	Interfaces, Power and Protection for Networks
T1M1	Internetwork Operations, Administration, Maintenance, and Provisioning (IOAM&P)
T1P1	Wireless/Mobile Services and Systems
T1S1	Services, Architectures and Signaling
T1X1	Digital Hierarchy and Synchronization

In the April 2000 meeting of T1A1, following passage of Letter Ballot # 837 and under Project T1A1-20, ITS began shepherding the work of revising the glossary, working within the T1A1 community to complete the task, and specifically within the T1A1 Ad Hoc Glossary Group, convened by an ITS staff member.

The ITS editors compiled some 1250 new term names and definitions from the current literature and authoritative texts on

- Web/Internet terminology
- Security and Infrastructure Protection
- Photonics

The editors also combed the T1 literature (ANSI T1 Standards and T1 Reports) to gather another 900+ terms and definitions supporting the T1 focus on interfaces for U.S. telecommunications networks and on those functions and characteristics associated with the interconnection and interoperability of telecommunications networks at interfaces with end-user systems, carriers, and information and enhanced service providers. (These fields of interest include switching, signaling, transmission, performance, operation, administration and maintenance aspects.)

The two collections of terms (the 1250 terms from the literature and the >900 terms from the T1 documents) were integrated into one alphabetical list of ~2300 terms, which were circulated via e-mail to all members of the T1A1 Ad Hoc Glossary Group for review, comment, and revision. Rather than overwhelm the Ad Hoc Group (AHG) with a huge body of some 2300 definitions to review at one time, the editors circulated to the AHG the alphabetically organized definitions, one or two letters of the alphabet at a time, via e-mail, until the entire list of terms had been reviewed by the AHG. In this manner, in 20 to 25 weeks, all AHG reviewers had a chance to examine the 2300 proposed additions or changes to the glossary. To preserve due process, all reviewers' comments were circulated to the entire group via e-mail, and conflicting opinions were resolved by consensus. When the 2300 proposed revisions had been addressed and incorporated into the draft glossary by this consensus method, the new draft *Telecom Glossary 2000* had grown from 5800 to 8090 entries. Contribution T1A1/2000-052 summarized the scope and progress of the work. Letter Ballot #938 was a proposal for consideration of adopting the revised glossary as an American National Standard (ANS) at the November 2000 meeting of T1A1. Approval of letter ballot #938, which closed December 15, 2000, recommended adoption of the glossary as an American National Standard.

The results of the letter ballot #938 were

Approvals	42
Disapprovals	0
Abstentions	12
Ballots not returned	26

More than 170 comments were received with the return of letter ballot #938. All of the comments were either minor clarifications, amplifications, or additions to provide fuller illumination of meanings. Each of the comments was considered and was either (a) adopted, or (b) modified by consensus and adopted, or (c) rejected for the reasons specified in: T1A1 Contribution 2001-003, 23 Jan 2001, which is posted on the T1A1 Web page.

At the February 2, 2001, meeting of T1A1, this Contribution was addressed, and the revised, corrected *Telecom Glossary 2000* was presented at the Web site

<http://www.its.bldrdoc.gov/projects/telecomglossary2000>

where it was offered as an American National Standard, in just under 12 months after the project was officially launched in T1. ANSI approved T1.523-2001 as an American National Standard on February 28, 2001.

The electronic meetings — and the programming scripts used to collect, organize, and groom the results of those electronic meetings — made possible the timely completion of this very large, technical glossary.

1.3 Purpose of *Telecom Glossary 2000*

The objective of *Telecom Glossary 2000* is

- to establish and maintain uniform, up-to-date definitions for the terminology used in telecommunications

The associated goals of the glossary are

- to promote better communications among U.S. providers and users of telecommunications equipment and services,
- to enhance understanding of U.S.-developed telecommunications standards, technical reports, technical requirements, contributions, and other specifications in international standardization and trade, and
- to advance public knowledge of telecommunications concepts, and associated public-interest telecommunications capabilities.

1.4 Expected On-Line or Internet Audience for *Telecom Glossary 2000*

As with the 1996 edition of FED-STD-1037C, the revised glossary is intended to be of value to telecommunications professionals from Government, the private sector, and academia, and of particular value to workers in Federal telecommunications procurement. Specifically, the glossary is designed to be used by

- telecom planners and managers,
- acquisitions managers,
- telecom equipment providers,
- telecom service providers,
- telecom equipment manufacturers,
- government standards writers,
- government standards users,
- researchers,
- technical writers/editors,
- O&M workers,
- telecom instructors,
- NS/EP (National Security/Emergency Preparedness) planners, and
- T1 Standards developers and users.

In developing *Telecom Glossary 2000*, to expand it beyond the Federal Government sphere of users, the developers included inputs from industry—especially T1A1 and other TSCs (technical subcommittees)—as the standard migrated from a Federal Standard to a National Standard.

Analysis of the initial 250,000 hits on the 1037C Web page showed that the 1996 glossary has been extensively used by all U.S. industry sectors, and by organizations in dozens of other countries. The 1996 glossary Web page has received more than 500,000 hits to date. Requests have been received

to translate it into Chinese and Hungarian, and requests for copies have come from Romania, Argentina, and other sectors of the international community.

The hit counter for the new ANS T1.523-2001, *Telecom Glossary 2000*, has registered—at this writing—more than 35,000 hits, even without the benefit of a formal notice or press release announcing its presence on the Web.

1.5 Participants in the Development of *Telecom Glossary 2000*

Anyone expressing an interest in participating in the glossary’s revision was invited to do so by submitting their name and e-mail address to the editors so that they could be added to the e-mail exploder list. At the time that the standard was approved, the revision team included the following members (from the glossary e-mail mailing list) shown in Table 2 below.

Committee participation also reflected the on-line electronic trends in communication, incorporating the medium into the process.

Table 2. Membership of the *Telecom Glossary 2000* Ad Hoc Group

ORGANIZATION (bold indicates industry or academia)	NAME OF REPRESENTATIVE
ABC TV	Warner Johnston
SBC Ameritech	Lorence F. Brown
Assisi Consulting	Don Josephs
Compuserve	John Boal
Ensemble Communications, Inc.	Marsha Jovanovic
Fluke Corporation	Fred Kaudel
Global One Communications, Inc.	Paul W. Vince
New Mexico State University	E. Johnson
Qwest (formerly U S West)	Candace Bruce-Andrews, Gary Stephans
SAIC	Henry M. Kluepfel
Siemens	Agus Salim
University of Canberra	Sakari Mattila
Federal Communications Commission	Whitey Thayer
U.S. Department of Agriculture	Bill Demarr
U.S. Department of Commerce	Jorome T. Gibbon
U.S. Department of Commerce, NTIA	John K. Lady
U.S. Department of Commerce NTIA/ITS	Neal Seitz, William Ingram, Evie Gray
U.S. Department of Defense, DISA	J. Davies, G. Ring
U.S. Department of Defense, USAF	Johan Dekker
U.S. Department of Energy	Carol Blackston
U.S. Dept. Health /Human Services	J. Price

U.S Department of Interior	James Dolezal
U.S. Department of Justice	Henry Ickes
U.S. Department of Transportation, FAA William J. Hughes Technical Center	James S. Beaty
U.S. Department of Transportation, FHWA	William S. Jones
U.S. Department of Treasury	Tahir Latif, Gary Allmond

Essentially all of the Federal agencies in the above list also participated in the development of the older 1996 FED-STD-1037C, but for the *Telecom Glossary 2000*, several industry representatives and members of academia joined the on-line development work and were added to the Ad Hoc Group (indicated by the **bold-font** in the above list).

2. CONTENT OF *TELECOM GLOSSARY 2000*

2.1 Disciplines Addressed in the Old and the New Glossaries

The disciplines addressed in Federal Standard 1037C included:

- Telephony,
- Fiber Optics Communications,
- Radio Communications,
- Radar,
- TV: VHF, UHF, Cable, HDTV,
- Facsimile,
- Networks: INs, Open Network Architecture, ISDN, B-ISDN, Network Management,
- COMSEC,
- Data Processing,
- Premises Wiring; Grounding/Bonding,
- Spectrum Sharing, and
- NS/EP (National Security/Emergency Preparedness).

All of the above fields were retained in the new *Telecom Glossary 2000*, and other fields were added from e-mail and from the consensus of electronic meetings. The details of the new contents and their sources are discussed in Section 2.2 below.

2.2 New Material Added to Develop *Telecom Glossary 2000*

The proposed *Telecom Glossary 2000* used as its baseline document FED-STD-1037C, *Glossary of Telecommunication Terms*. To this baseline document, using e-mail and electronic-meetings techniques, the developers of the new glossary added 2300 new entries. The sources of all entries, as indicated on the Web document, are as follows:

- o 1250 entries (preceded by an open **red** dot in the alphabetical listing frame) are from the open literature, and they include:
 - Web / Internet terminology,
 - Security terminology, and
 - Photonics terms,
- # 900+ entries (preceded by a **blue** square bullet) are from T1 standards and reports, and
- ! 5800 entries (preceded by a **black** dot) are either from the 1996 edition of FED-STD-1037C, or are definitions proposed and reviewed by industry standards workers.

Combined with the original 5800 entries from FED-STD-1037C, and with the deletion of obsolete terms, the sum of the entries in *Telecom Glossary 2000* is now more than 8000.

2.3 Features of the Finished Glossary

The completed *Telecom Glossary 2000* contains more than 8000 entries, internally linked source citations to the list of references, approximately 90 illustrations, 4 video clips, 2 audio clips, and a fully functioning search engine that operates in basically the same manner [3] as the search engine for the Web version of FED-STD-1037C. Peripheral materials that have been revamped or expanded beyond the FED-STD-1037C version include:

- revised front matter (with expanded reference list),
- revised Appendix of Acronyms/Abbreviations, and
- modified Web pages to provide easier access for the disabled.

2.3.1 Glossary Entries

The definitions are written as a single defining phrase with accompanying tutorial notes giving examples, units of measure, typical uses, or other tutorial information. For term names with multiple definitions, the definitions are numbered using bold numerals (often with the more general definition listed as No. 1), and the restricting discipline is usually cited in an opening phrase for each definition.

Rules of style and syntax used for *Telecom Glossary 2000* are similar to those rules that were used to write the definitions in FED-STD-1037C, as explained in an earlier report [4]. For example, the new glossary defines the *singular form*, and defines entries in positive terms by explaining what an item *is*, rather than what *it is not*.

In each definition, when another defined term is used, that term is hyperlinked to its own definition. For instance, if the definition of *computer* contains the word *data* in it, the word *data* appears on the screen in a color (usually blue) and with an underline. Clicking on the hyperlinked word *data* takes the viewer immediately to the definition of the word *data*. The techniques for developing this automatic hyperlinking using program scripts and an anchors list are described below in Section 5, and are also described in an earlier report [3].

2.3.2 List of Acronyms

The list of abbreviations, acronyms, and initials is given in the appendix of *Telecom Glossary 2000*. Each of the **defined** entries in that alphabetical list of acronyms is hyperlinked to the full definition in the body of the glossary. For example, the appendix lists the entry *dB* as an abbreviation and explains that *dB* stands for *decibel*. The *dB* in the acronym list is hyperlinked to the definition of *dB* in the body of the glossary. (Only defined entries in that appendix are hyperlinked.)

2.3.3 The Source Material Used for Expanding the Glossary

The terms and accompanying definitions contained in *Telecom Glossary 2000* are drawn from authoritative sources such as Federal Standard 1037C, *Glossary of Telecommunication Terms*, 1996; T1 Committee Standards and T1 Technical Subcommittee Standards and Reports; documents from the International Telecommunication Union; documents from the International Organization for Standardization; and documents from the American National Standards Institute, and numerous authoritative dictionaries and lexicons. Numerous U.S. Government publications (*e.g.*, the *Code of Federal Regulations, Title 47*) were also drawn upon for this and earlier versions of the standard. All of the normative references for the *Telecom Glossary 2000* are given its Foreword on the Web; the specific titles and dates for these references are repeated below in Appendix B of this report.

The editors of this standard have rewritten many definitions as deemed necessary either to reflect technology advances or to make definitions, which were phrased in specialized terminology, more understandable to a broader audience.

In those cases where definitions were taken verbatim from copyrighted sources, the editors received *written* permission from the copyright holder to use the definition. The definitions from 47 CFR, from the *NTIA Manual*, from the *Radio Regulations*, and from other treaty documents were used verbatim.

The coded source citation is shown in hyperlinked square brackets [] following most definitions. The alphabetical References list details the full title for each document that is source cited in square brackets. Clicking on the hyperlinked word *References* (which appears below the text of each definition) takes the viewer to the top of the alphabetical list of References in the Foreword to the glossary. A simple scrolling down this list allows the viewer to find the full title of the reference whose shortened name was given in square brackets at the end of the definition. For instance, if the definition in question relates to *base Earth station*, the on-screen entry looks like Figure 1:

base Earth station

base Earth station: An Earth station in the fixed-satellite service or, in some cases, in the land mobile-satellite service, located at a specified fixed point or within a specified area on land to provide a feeder link for the land mobile-satellite service. [NTIA] [RR]

This HTML version of Telecom Glossary 2K was last generated on Fri Oct 27 17:19:46 MDT 2000. References can be found in the Foreword.

Figure 1. Example of a definition from the glossary.

The reader who reads the glossary on the Web can click on the underlined hyperlinked word “References” shown below the definition. That hyperlink takes the reader to the Foreword, where he/she can scan the alphabetical legend of source citation codes to find the full title of the reference cited here. In this case, the full title for the reference cited as “[NTIA]” is *NTIA Manual of Regulations and Procedures for Federal Radio Frequency Management*. The full title for the source cited as “[RR]” is *International Radio Regulations*.

Definitions with no source citation following them are either (a) from FED-STD-1037C (which was the baseline building block for the new glossary), or (b) from a body of definitions proposed and reviewed by industry standards workers in the course of developing the draft *Telecom Glossary 2000*.

The full list of Normative References, which was the source material used for expanding the glossary, and which was called the *Legend* in earlier editions of the glossary, is given in Appendix B of this report and is detailed in the Foreword of the on-line glossary.

2.3.4 Illustrations, Audio and Video Clips

The glossary includes approximately 90 figures to illustrate technical processes, configurations, and phenomena. The figures are shown only in small icon form on the Web page for their appropriate definitions. The small icon presentation speeds delivery of the definition for those who access the Web with a modem. Anyone can view the full-size graphical image by clicking on the hyperlinked icon that says . These images are created in the .gif format. An example of a two-dimensional graphical image is shown in Figure 2.

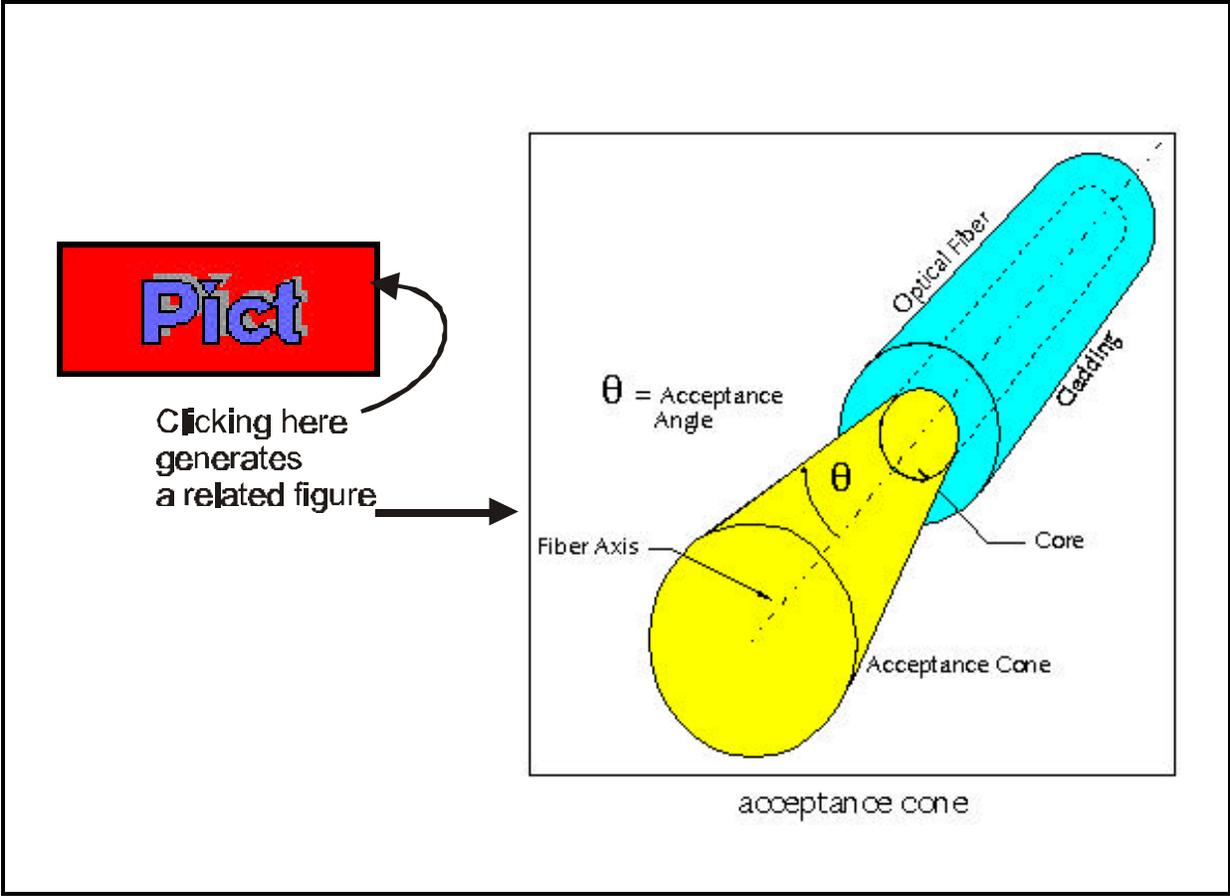


Figure 2. An example of a two-dimensional image accessible in the glossary.

The glossary contains four video clips. Anyone can view the full-size video clip by clicking on the icon that says **Video**. For the fullest appreciation of the video clips, the viewing window should be enlarged to nearly full screen. An example of the presentation of the “video” icon and a still frame from one such video is shown in Figure 3.

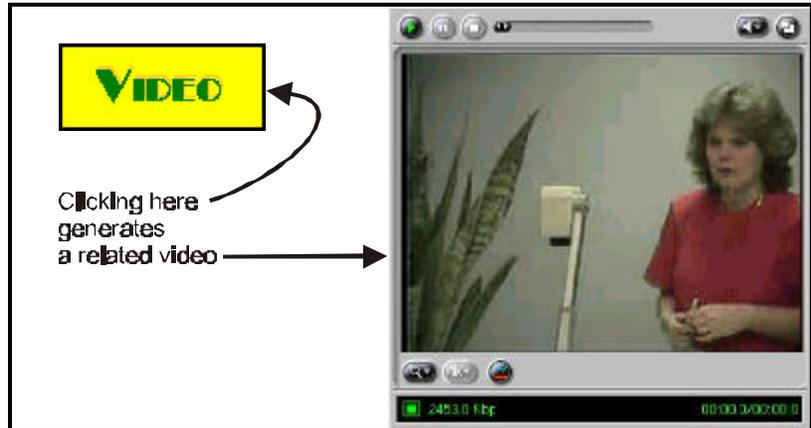


Figure 3. An example of access to a video clip in the glossary.

Two audio files are included in the glossary. One can hear the audio tones by clicking on the icon labeled **Audio**. Figure 4 illustrates the visual portion of the events following a click on the audio icon.

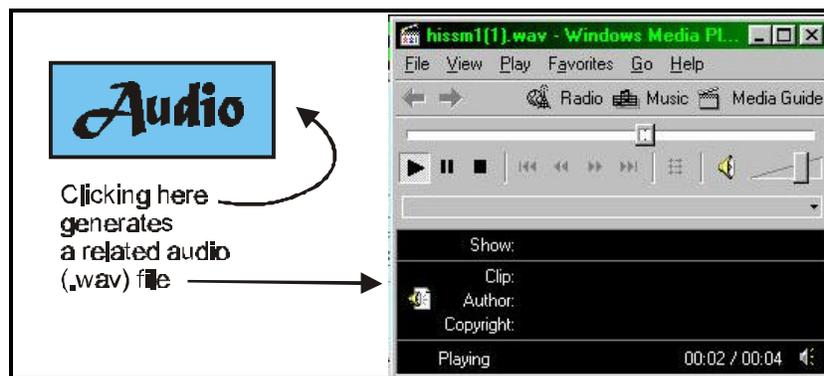


Figure 4. The visual indication of events following a click on an “audio” icon.

2.3.5 Search Engine

The fully functioning search engine is accessible either (a) by clicking on the words “Search Engine” that appear in the upper right-hand frame of the initial Web page for the glossary, or (b) by clicking on the search engine icon in the bottom frame of the glossary.

Operating the glossary search engine is intuitive, much the same as operating any other search engine. The glossary search engine obviates the need for any index, which was provided for FED-STD-1037C, and which was useful in hard copy.



The benefits of the Web-based search engine for the glossary include:

- Tailored access to the text of all definitions,
- Ranking of search results,
- Speedy, organized use of the entire glossary or only those portions of interest, and
- Hyperlinks to all search results.

The search engine for *Telecom Glossary 2000* has been enhanced so that it provides results in two useful formats:

1. A format that ranks results as a hierarchy of hyperlinked (clickable) term names most likely of interest, and
2. A format that presents the results on one (Web) page showing the full text of all definitions caught in the net of that search.

3. BENEFITS OF ELECTRONIC UPDATING

3.1 Previous Updating Techniques

In the past, updates in the FED-STD-1037 series were developed using principally face-to-face committee meetings. For more than two decades, ITS has held a leadership role in developing those updates: chairing the meetings, soliciting technical definitions, recording the results, incorporating the decisions into the glossary database, developing the coordination drafts, incorporating public review comments into the final draft, and performing related tasks.

For the 1996 edition of the Federal Standard, the definitions were recorded in the computer database during committee meetings. That process is illustrated in Figure 5. The process required 10 one-week meetings over a 1-year time span. It was a labor-intensive process.

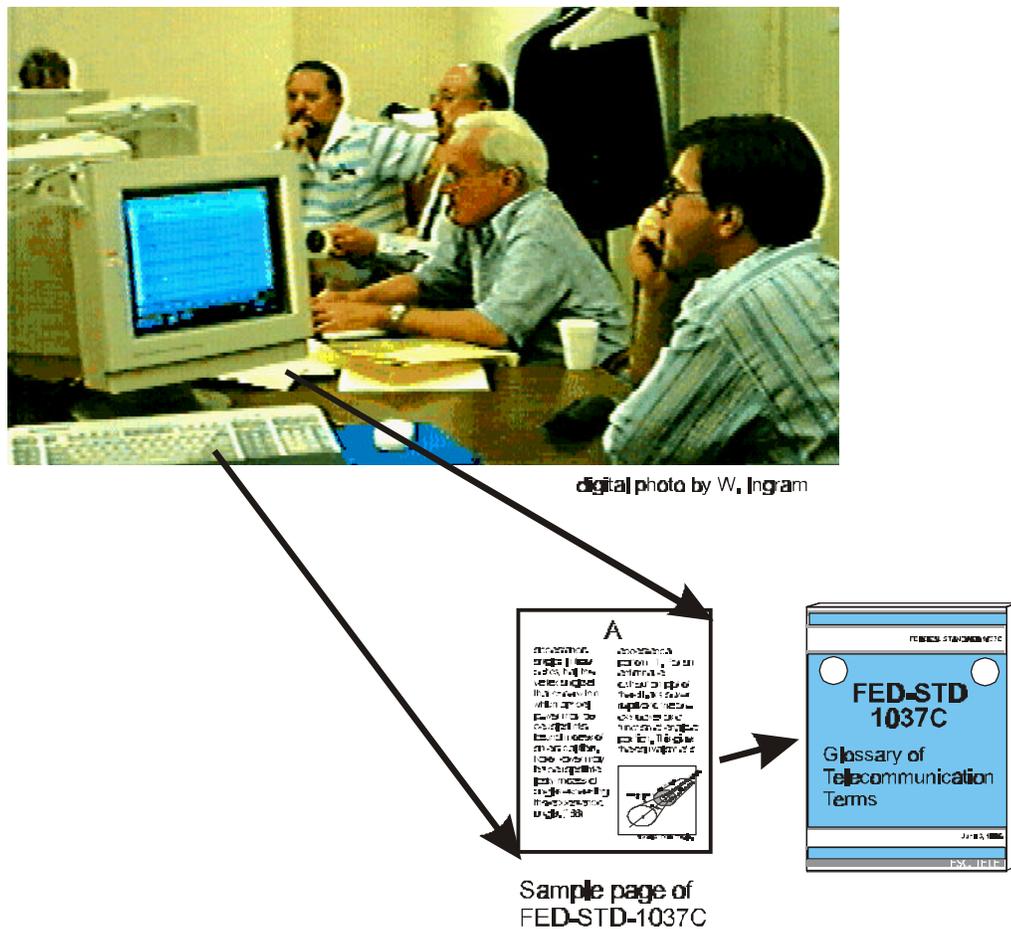


Figure 5. Committee method for earlier (1996) glossary update.

3.2 Summary of On-line Updating Techniques for *Telecom Glossary 2000*

For the *Telecom Glossary 2000* edition, the ITS editors decided to make use of the ubiquity of e-mail and the Internet to introduce efficiencies into the collaborative revision process. They chose to conduct “virtual meetings,” and to collect the experts’ comments via e-mail and via the Web, thus removing the expense of face-to-face meetings. The basic elements of the process are illustrated in Figure 6, and the details of the procedures are explained in Section 4.

3.3 Advantages of Web-based and E-mail-based Procedures

The electronic-meetings technique used for updating the current glossary has the advantages of

- speed,
- efficiency and economy,
- wider participation of experts, and
- accuracy and accessibility.

Use of such electronic meetings or virtual meetings would provide the same four advantages for other time-sensitive on-line products that require timely updates to keep them valid and usable. Each of these four advantages is discussed in detail below.

3.3.1 Speed

The electronic meetings for updating the glossary allowed the editors to collect experts’ critical reviews and technical proposals via e-mail in a much more expeditious manner than would have been possible with scheduled meetings for collecting such information. As mentioned, more than 2300 entries were addressed and incorporated into the glossary within 12 months. This speed is unlikely to have been possible with scheduling and conducting face-to-face meetings.

3.3.2 Economy and Efficiency

The cost of participation in the electronic meetings, while not free, was limited to the cost of computer access and Web access. The cost of travel was eliminated almost entirely. The conservation of paper-production resources was greatly enhanced by the Web-development procedures because no paper copy drafts of the glossary were distributed. The electronic meetings also increased the efficiency and speed of the review process itself, because experts were free to focus only on their special areas of expertise and provide only a cursory review of other disciplines. Further, the reviewer could tackle the individual definitions of interest during spare moments, without having to block out a week of 8 or 9 hours a day (plus travel time) for glossary meetings.

3.3.3 Wider Participation

The Web reaches a far wider audience than one can reasonably expect to reach with face-to-face meetings. Experts' participation in electronic meetings is possible even if those experts wish to comment on just a few technical definitions, whereas their travel to face-to-face meetings might not have been possible or economically justifiable for them.

E-mail exchanges were also considerably less expensive than mailings of hard copy. As a result, e-mail solicitation and distribution of comments was extended to a wider audience of experts, many of whom initiated contact with the editors (and many of those experts would have been unreachable with older committee-roster contact methods).

3.3.4 Accuracy of the Record

Archival records of e-mail discussions and the resulting definitions were much easier to distribute via e-mail than via hard copy. For earlier editions of this glossary, the record of decisions on each definition was distributed via hard copy only. The financial cost of distributing that hard-copy record was burdensome because the record was voluminous and its development was labor-intensive. Without review of the hard-copy record, the subcommittee for the old glossary (using old revision techniques) had no way of determining the accuracy of the changes being made to the old draft of FED-STD-1037C before those changes were printed in the Coordination Draft. With the new e-mail procedures and Web-based updating mechanisms used for *Telecom Glossary 2000*, reviewers were able to examine daily the results of their proposals as adopted by consensus. This more frequent, more prompt examination of the database changes resulted in greater accuracy for the new glossary.

Further, in the new Web-based glossary update, accuracy was much more easily and readily assured by the mechanisms available for frequent updates. These updates (corrections or changes) were made with a simple HTML command after the change had been developed by the technical experts and sanctioned by due process. This meant that a current Web version of the glossary was more accurate (more technically correct and up-to-date) than any existing printed version that may have been sent to press weeks or months before the latest Web change had been made.

The wider distribution available for the Web glossary also enhanced its accuracy by subjecting its contents to a greater number of eyes and a broader selection of examiners than would have been possible with traditional distribution of hard copy. The more reviewers for a document, the more accurate that document is likely to be.

3.3.5 Availability of the Glossary

Because printing costs are essentially eliminated and because of the ubiquitous nature of the Web, the on-line version of the glossary is much more readily available than a hard-copy version would be. One of the limiting factors for availability of hard-copy is cost. Paper, printing, and distribution

costs are significant for a document that has grown beyond 600 pages. (At typical 2001 costs of 5¢ per page, the document would cost more than \$30 for photocopying alone. Shipping costs would also be significant.) Additionally, the hard-copy version would be lacking in many of the enhancements available in the Web version because the hyperlinked search engine, video files and audio files cannot be stuffed into a traditionally printed hard copy. For anyone with a computer and Web connection, the glossary (including its hyperlinked search engine, video files, and audio files) is fully available at no charge beyond the cost of Web hook-up.

4. ON-LINE PROCEDURES FOR UPDATING THE GLOSSARY

4.1 Overview of Updating Procedures

The ITS editors developed on-line procedures, illustrated guidelines, and instruction lists to shepherd participants through the Web-based updating process. The conceptual map of that process is illustrated in Figure 6. Each of the 4 steps highlighted in the figure is explained in Section 4.2.

The procedures developed (shown in Figure 6) for online updating and development of *Telecom Glossary 2000* were grouped into four main processes. The underlying principles for these procedures were

- ease of use for online updating of the glossary,
- compliance with ANSI due-process rules,
- ease of access to the glossary, and
- protection of the integrity of the database representing the glossary.

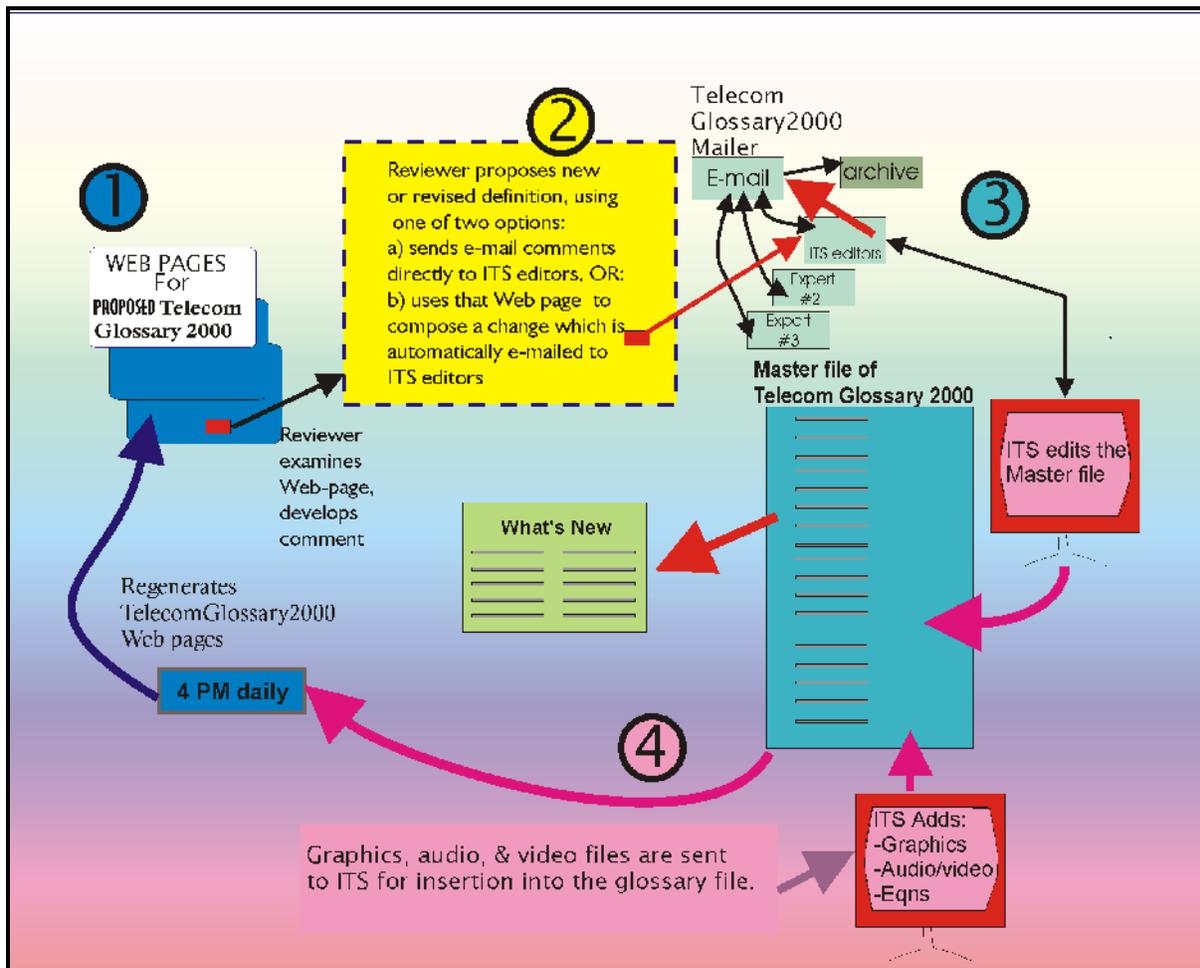


Figure 6. Conceptual map of the electronic processes of developing *Telecom Glossary 2000*.

4.2 Four Major Steps in the Updating Process

The four procedural steps that are numbered and highlighted in the figure are:

1 The user joined the discussions by sending his/her e-mail address to:
evie@its.bldrdoc.gov or to
bing@its.bldrdoc.gov

Anyone could submit proposals of new/revised definitions for the glossary, but only members of the discussion list discussed the changes (via e-mail) and arrived at a consensus.

2 The reviewer submitted a new definition (or revision) directly via e-mail to
evie@its.bldrdoc.gov or to
bing@its.bldrdoc.gov

OR:

he/she submitted a proposed new definition (or revision) via the *Telecom Glossary 2000* Web page (which automatically generated equivalent e-mail).

3 The editors at ITS sent all proposed revisions (via the e-mail exploder) to:
a) experts on the mailing list,
b) the Master file, and
c) the “What’s New” file.

4 The editors at ITS edited the Master file and inserted multimedia submitted by participants. Daily at 4 pm (MDT) new Web pages of the glossary were generated from this Master File to include new proposals. (Details of this step are discussed in Section 5.)

The procedures eventually adopted for electronic review and editing of the glossary are the ones shown in Figure 6. These procedures were modified from the early ones, as follows: originally, the Web site was password protected, but the password severely hindered the revision process. The original procedures for updating the glossary also included the capability for the reviewer to modify a given definition directly on the Web using HTML code. But this capability was terminated when the password was removed, because policing the Web site to protect it from inappropriate hacker comments would have been an arduous task. Discussions with reviewers showed that most of them preferred sending their revisions via e-mail text rather than in HTML code, so that removing the HTML Web-editing feature was not a hindrance, and removing the password was a positive advantage. In their final form, the procedures for collecting reviewer comments allowed for the submission of e-mail comments only. Those e-mail messages were sent only to the ITS editors, who then circulated them to the Ad Hoc Group. This two-stage process provided the necessary protection that had originally been provided by the password.

4.3 Review of the Draft on the Web

All interested viewers and participants in the review process were able to examine the draft of *Telecom Glossary 2000* at the Web address:

<http://www.its.blrdoc.gov/projects/t1glossary2000>

When they accessed that page, they were able to see the Web page illustrated in Figure 7.

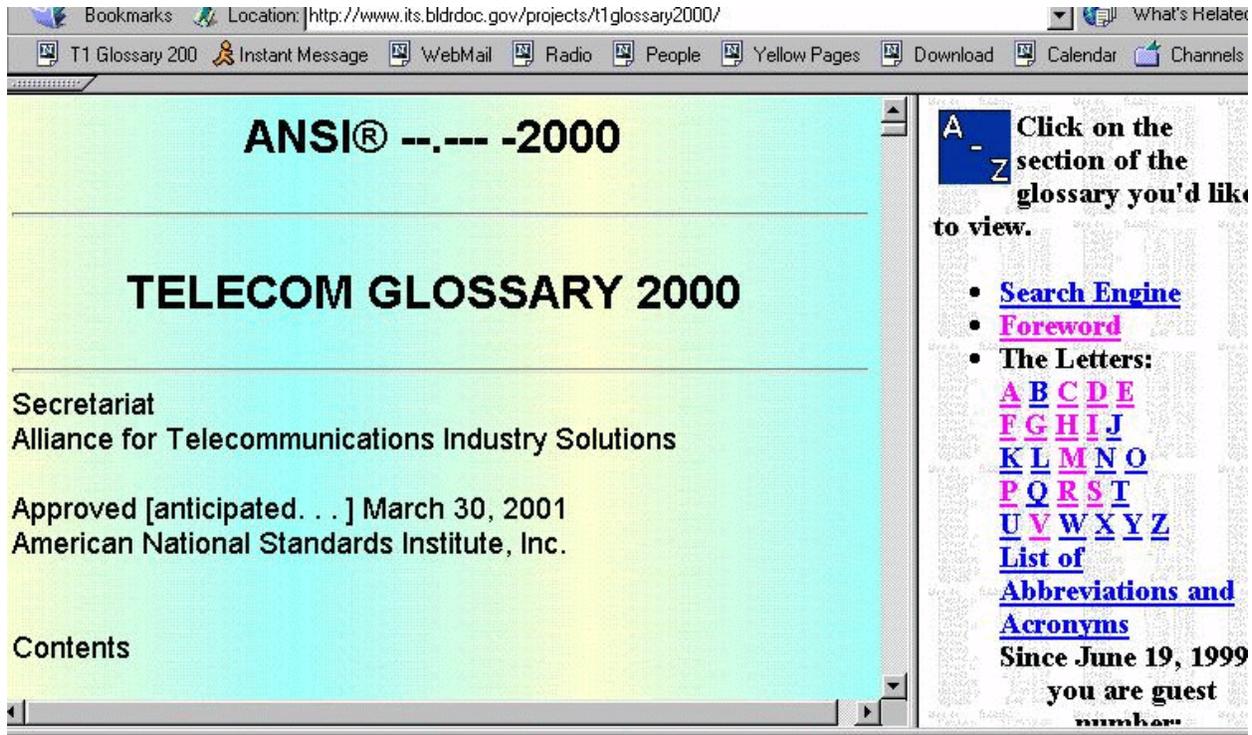


Figure 7. *Telecom Glossary 2000*, “Foreword” page as it appeared (draft form) for Web editing.

When the viewer clicked on the “Main” hot spot, the page shown in Figure 8 appeared. Some particulars of this Web page have changed as the glossary migrated toward completion. But, in draft form, the glossary’s opening Web page contained many hot spots and hyperlinks. Seven of those useful links—which allow the user to propose new definitions, to modify existing definitions, or to send e-mail to the ITS editors—are illustrated in Figure 9.

<http://www.its.bldrdoc.gov/projects/t1glossary2000>

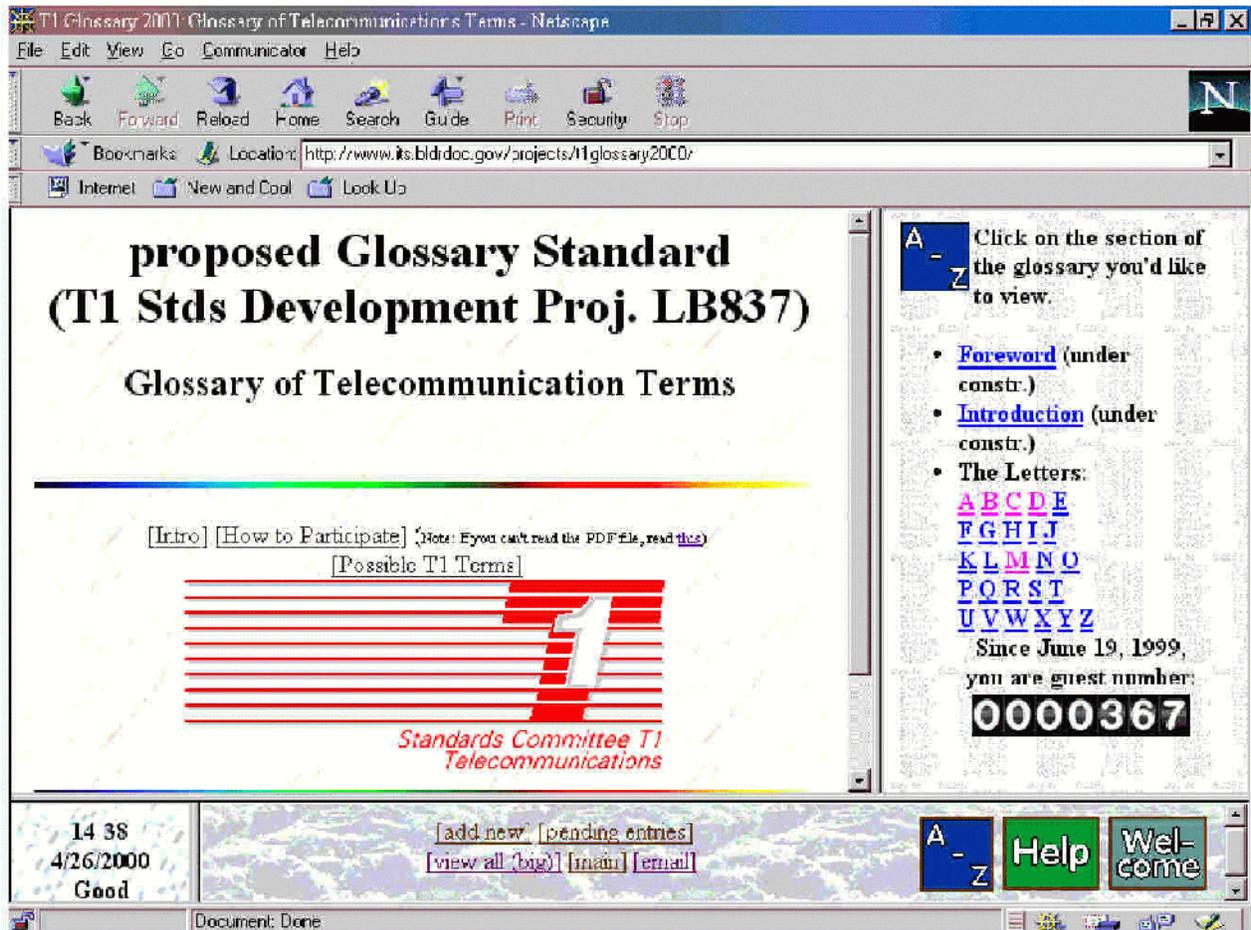


Figure 8. The “Main” page of the Web-based *Telecom Glossary 2000*, as of November 2000.

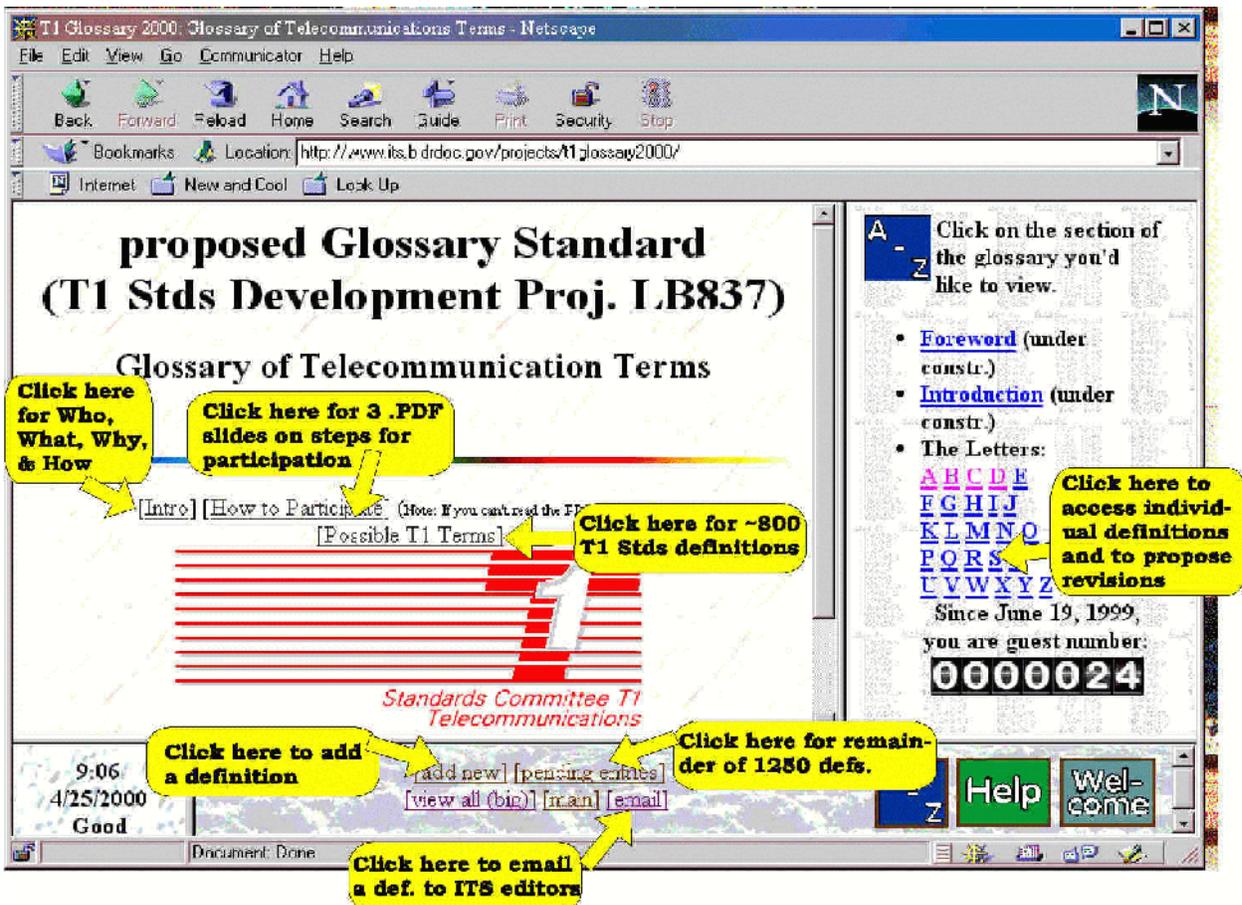


Figure 9. The several hot spots and hyperlinks on the initial *Telecom Glossary 2000* Web page.

Figure 9 is an illustration of the same *Telecom Glossary 2000* “Main” Web page with the addition of bubbles and arrows identifying and explaining seven of the several hot spots and hyperlinks that were available on that Web page when the draft of the glossary was being developed. As the work progressed, some of these seven hot spots were necessarily removed; for instance, the “[pending entries]” hot button was removed when all of the pending items were converted to final, consensus-determined form. These seven—the handiest of the hot spots for users—are (clockwise):

1. The first button, the “[Intro]” hyperlink, was removed in the final version of the glossary because it contained exactly the same information as is visible by clicking the “Welcome” button. (See Figure 10 below.)

2. Clicking the “[How to Participate]” hyperlink on the Web draft of the glossary displayed three slides in .PDF format (viewable with a PDF reader), that list the 4 numbered steps in the review and revision process that are also shown in Figure 6 above (and discussed in the text of Section 4.1). This “How to Participate” button was removed in the final version of the glossary.
3. Clicking the “[Possible T1 Terms]” hyperlink on the Web draft of the glossary provided an alphabetical list of approximately 900 definitions scanned into the draft glossary from hard copies of T1 Standards and Reports. These definitions are slightly reformatted to conform with the standard glossary presentation format, but no technical revisions were proposed in the early draft. This “Possible T1 Terms” button was removed in the final version of the glossary.
4. Clicking on any one of the letters in the screen’s far right frame presented a list of term names beginning with that letter into that frame. For instance, if one clicked on the letter “S,” all of the proposed term names and existing FED-STD-1037C term names beginning with the letter “S” would appear in the far right frame. One of the term names listed in that frame might be “security exchange.” If one clicked on “security exchange,” the definition (and icons for any graphic images, video clips, or audio clips) would appear in the large central frame of the screen.
5. The hyperlink labeled “[pending entries]” in the Web draft of the glossary allowed the user to access unresolved definitions from the original 1250 proposed new entries addressing security terms, Web/Internet terms, and photonics terms. This hyperlink was removed when the glossary was completed and all details of pending definitions were resolved.
6. The “[email]” hyperlink allowed the user direct access to the e-mail addresses for the ITS editors.
7. The “[add new]” hyperlink in the Web draft of the glossary allowed the user to propose a new definition (for a previously nonexistent term name) by clicking on this hot spot and writing an e-mail message of the proposed definition. This e-mail message was sent to the ITS editors, who then relayed the message to the members of the Ad Hoc Glossary Group. This hyperlink was removed when the glossary was completed.

By using hyperlinks 6 [“email”] or 7 [“add new”], the reviewer was able to propose revisions or additions to the glossary by submitting a proposed new entry via e-mail directly to the ITS editors.

In addition to the 7 hot buttons described above, the glossary contains a “Welcome” hot button, a “Help” hot button, and an “A - Z” hot button.

By clicking on the “Welcome” icon or the “Intro” hyperlink, the viewer can read the information shown in Figure 10 below.

ABOUT TELECOM GLOSSARY 2000

What

Under an approved T1 standards project T1A1-20, a previously developed 5800-entry, search-engine equipped, hypertext telecommunications glossary, currently on the Web at

<http://glossary.its.bldrdoc.gov/projects/telecomglossary2000>

was updated and expanded as an American National Standard (ANS). The predecessor document, FED-STD-1037C, was approved by the Federal Telecommunication Standards Committee (FTSC) and published by the General Services Administration in 1996. The new ANS (*Telecom Glossary 2000*) enhances FED-STD-1037C by adding terminology addressing new technology areas (e.g., Internet, photonics, infrastructure protection), and eliminates government-specific material and obsolete terms (e.g., "Mosaic"). The ANS adds (or references) selected terminology defined in T1-developed standards, requirements, and reports. The project was accomplished using primarily Web-based processes, including on-line revision of definitions, e-mail communication and data entry techniques, and "virtual meetings."

Available for review at the above-referenced Web site are:

- 5800-entries (in hypertext) from FED-STD-1037C.
- 1250 (alphabetically interleaved and hyperlinked) new definitions addressing:
 - Internet / Web technology
 - infrastructure protection technology
 - photonics technology
- Approximately 1000 terms and definitions selected from T1-developed standards,
-

Who

The project was accomplished by a T1A1 Telecom Glossary 2000 Ad Hoc Group comprising interested T1A1 members, contributors from other T1 groups, and members of the FTSC. . . The ITS served as editor for the glossary, and is co-hosting the Telecom Glossary 2000 Web site.



Figure 10. The essence of the text for the “[Intro]” and “Welcome” hyperlinks.



Clicking on the “Help” icon produced the text shown in Figure 11 in abbreviated form.

Help Section of T1 Glossary 2000

The typical use of this glossary is to look up a particular term or definition. The steps to do this are:

Click on the "A-Z" box (if the "A-Z" page is not already in the upper right Frame).

Click on the letter of the alphabet for the term you'd like to view. The upper right Frame should change to the listing of all the term names that begin with that letter. Use the vertical slide bar along the right edge of that Frame until you can see the definition name you'd like to view.

Click on that term name. The upper left Frame should display the definition(s) for the term you have selected. The upper right Frame should remain as the listing for the letter of the alphabet.

You can select any highlighted word within the definition to jump to that new term and definition, or you may select another definition from the list on the right.

PICT If you see this icon within a definition, that means that an illustration accompanies the text. Click on the icon and the illustration will appear in the frame where the definition was.

AUDIO If you see this icon within a definition, that means that auditory information is available to help demonstrate the definition. Click on the icon and the audio information will be downloaded to your computer. If your computer is configured to play WAV files, you will hear the sounds. If not, you will get an error message that the file cannot be played on your system. Many programs are available over the internet that will allow your system to play WAV files.

VIDEO If you see this icon within a definition, that means that video information is available to help demonstrate the definition. Click on the icon and the video information will be downloaded to your computer. If your computer or browser is configured to play AVI video files, you will see and hear the video file.

Figure 11. Essential contents of the “Help” button.

4.3.1 Selecting New Definitions to be Added

As stated earlier (Section 1.1), the glossary was revised by adding some 2300 new entries from the fields of Web/Internet terminology, security/information-infrastructure terminology, and photonics terms. The new terms were selected from the literature and from ANSI T1 standards and T1 reports.

In the selection process, each proposed new entry was subjected to the following screens:

- Does the term define systems-level concepts or equipment (not a small component)?
- Does the term define an open-systems concept (not proprietary systems or services)?
- Is the term uniquely used in telecommunications (not a word defined in a desk dictionary)?

Additionally, terms submitted from the *Code of Federal Regulations, Title 47*, and from the *NTIA Manual*, and from the *Radio Regulations*, were all checked to verify that, if included in the glossary, they retained their referenced wording verbatim. These documents are all either treaty documents or law, and their words cannot be modified by the glossary developers.

4.3.2 Standardizing the Format for an Organized review

The format for the front matter (the Foreword, Introduction, and Normative References) was designed to follow the ANSI standardized format as illustrated in ANSI X3.172-1996, *American National Standard Dictionary of Information Technology (ANSDIT)*. The formatting of the body of the glossary was designed using rules of general legibility and clarity in a Web page, as described below. The format of the Appendix of Abbreviations and Acronyms was designed as a table two columns wide and more than 100 columns deep.

Web-page design

The Web pages of the glossary are designed to use frames. They can also be used by a browser that does not display frames. The default frames format is shown in Figure 12.

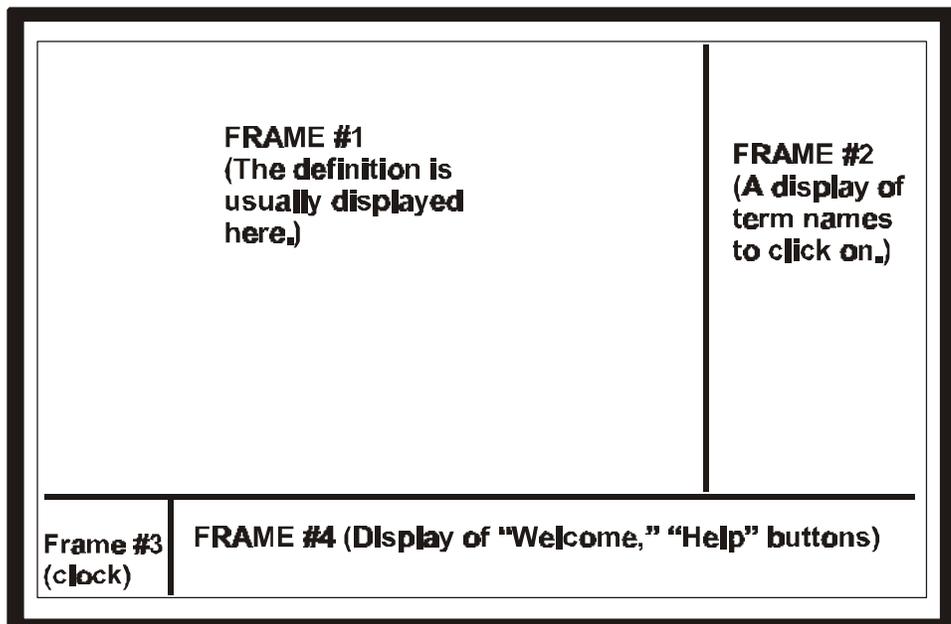


Figure 12. The configuration of frames used in the glossary Web pages.

A frame can be used to keep certain display items visible to the viewer while changing the information displayed in other frames. The frames format has the advantage that several small display areas can be addressed independently. In the seven years since the editors incorporated frames into the electronic version of FED-STD-1037C, every new version of each of the major Internet browsers has supported the use of frames. It is likely that every subsequent version will support frames as well. However, the glossary is viewable without using frames, as stated in Section 4.4 below.

The above format—coupled with a tailored search engine and hyperlinked, alphabetically arranged entries—allowed reviewers to examine the contents of the large glossary in a systematic, organized and thorough manner. Anyone can review any category of selected entries with relative ease.

4.3.3 Using the Search Engine for a Thorough Review

The specially tailored search engine was developed from an earlier search engine, described in NTIA Report 99-365 [5]. As described in that document, the search engine for the glossary provides speedy access to the entire text of definitions containing the specified character string used in the search. Previous versions of the search engine provided a list of term names only; the new enhanced search engine provides (on one page) the full text of all of the selected items. This display is usually given in Frame #1, and the clickable term names for the selected definitions are shown as a hyperlinked, weighted list in Frame #2, organized in a sequence of likely preference (depending on where the search string appeared in the definitions). The single-page presentation of the full text of all of the search-engine selected terms facilitates a much more thorough, organized review for experts using this feature.

4.3.4 Circulating Review Comments

Two types of comments were received during the 12 months spent developing this glossary:

1. Original comments (technical or editorial), and
2. Second-generation comments (comments on Reviewer comments [technical or editorial]).

To ensure full due process, all comments—whether original or second-generation—were circulated to the full e-mail reflector of the *Telecom Glossary 2000* Ad Hoc Group. In this manner, all comments received the benefit of many experts' critiques and the widest exposure for all comments.

When a reviewer saw an enhancement that needed to be made to a definition, he/she had three avenues for submitting that comment:

1. The “Add to this Definition” button that appeared on each individual definition page,
2. The “e-mail” button that appeared on the main page of the glossary, and
3. The “submit comments” button that appeared on the main page of the glossary.

Clicking on the first button (the “Add to this Definition” button) resulted in the automatic generation of e-mail to the ITS editors. Clicking on either of the other two buttons above also resulted in the same generation of e-mail. Distributing these three e-mail-generating buttons in three separate locations provided a user-friendliness that allowed the reviewer to submit comments at any point in the document without having first to use the “back” button to retreat to the main page or home page. Button #1 provided the reviewer with the ability to propose specific text (in HTML format), which could then be easily incorporated into the database if the Ad Hoc Group chose to use that text. Additionally, the operation of button #1 provided some protection for the integrity of the database; i.e., only the ITS editors were able to modify the database, and only after achieving consensus.

4.4 Web-Page Compliance with Disabilities-Access Guidelines

The *Telecom Glossary 2000* Web page was modified to comply with guidelines for promoting ease of Web-page access by the disabled. This work was done in an effort to expand opportunities for all people, including those with disabilities, to access and benefit from the glossary Web site.

Using guidelines provided by the Center for Applied Special Technology (CAST), the ITS editors made four categories of adjustments in the glossary’s Web pages:

1. Alternate names for images

The Web pages were modified so that if a hyperlinked image is clicked and the image does not appear, the descriptive file name for the image will appear. This feature is expected to benefit those individuals who have the graphics feature turned off and who are running a browser for the blind; this feature allows the user’s browser to read aloud the descriptive file name for the graphic image instead of displaying it.

2. Mouse-overs and/or fly-outs

The Web pages were modified so that when the cursor touches or slides over a hyperlinked item, that item is identified in a fly-out box prior to its being clicked. This feature has advantages for those running a browser for the blind, since such a browser can speak aloud the name for the graphic image whose hyperlink is touched by the mouse. This modification also benefits people who have turned off the graphics display on their monitors so as to speed downloading onto their computer monitor.

3. Optional use of frames to display the Web pages

The glossary Web pages were modified so that they are now capable of being accessed without using the frames feature of the browser. This modification, in keeping with popular recommendations, allows viewers to read the glossary even if their browser does not support frames.

4. Identifying colors

Where color is used to distinguish segments of a list or of selected text items, the color in question is identified by name in square brackets following its appearance. Bullets used in the glossary vary in shape as well as color. Also, good contrast was used in the glossary’s still images so that all still images can be easily read either in color or in black and white.

5. PROGRAMMING

5.1 An Overview of the Processing

The ITS editors automated three kinds of processes in developing the glossary:

1. The collection and distribution of e-mail comments to all Ad Hoc Group members
2. Interactive editing of the glossary by the ITS editors
3. The management and grooming of the large database that constitutes the glossary, including processes for generating individual Web pages, and processes for the automatic generation of more than 69,000 hyperlinks among those pages.

A diagram showing the interrelationship of these three processes is given in Figure 13.

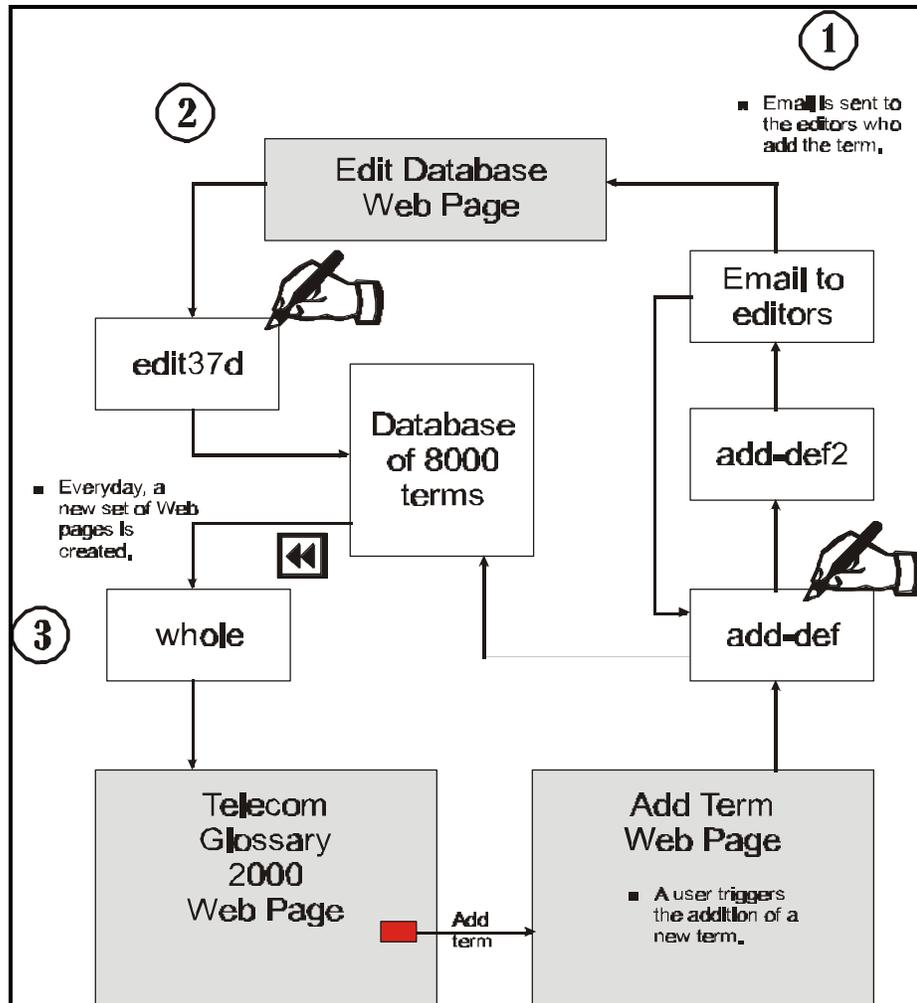


Figure 13. Flowchart of processes for adding to the glossary and grooming it.

5.2 E-mail Software

The ITS editors developed custom software for soliciting and collecting e-mail comments from all members of the ad hoc glossary group. The comments were solicited by placing hot buttons on the glossary Web pages that said [add a definition] or [send an e-mail] or [modify this definition]. These hot buttons automatically generated e-mail to the ITS editors, and then that e-mail was examined and forwarded to the Ad Hoc Group via an e-mail exploder list (or “e-mail reflector”). The dynamics of this software are illustrated in the right-hand portion (portion No. 1) of Figure 13.

The use of e-mail as the primary means of communication between business colleagues is becoming more widespread. E-mail has a myriad of advantages over conventional communication techniques. The speed of delivery—almost instantaneous, compared to conventional postal delivery—allows messages and ideas to be exchanged in minutes rather than days or weeks.

Yet, e-mail questions and suggestions and messages need not be answered immediately, unlike a telephone conversation or a face-to-face meeting. The recipient can take as much or as little time as is needed to compose the appropriate response. Similarly, the recipient of e-mail does not need to be present to receive the message. The message can be read at a later time when it might be much more convenient. Further, e-mail messages that are of no interest to the recipient can be ignored and deleted with little effort.

Like messages sent via the postal system, e-mail messages can be archived and saved for further reference. This electronic archival is even better than the regular postal mail, since the text of the saved e-mails can be electronically searched when reviewing earlier messages.

Also, nonstandard information, such as images, audio files, and video clips, can be attached to e-mail and sent to the recipient.

For these reasons, the ITS editors decided to incorporate the use of e-mail into the updating process for the glossary.

The ITS editors had to combine off-the-shelf e-mail software with custom-developed software. The off-the-shelf software consisted of an e-mail exploder to keep track of the e-mail addresses of all members of the Ad Hoc Glossary Group (membership list shown in Table 2), to duplicate any e-mail messages intended for the Ad Hoc Group, and to send one of the copies of the message to each member of the committee.

Custom e-mail software was needed to interface the glossary’s Web pages to the e-mail system. During the development process, each Web page for each definition contained a button that allowed the viewer to submit comments about the definition shown on the screen.

A series of Perl scripts guided the viewer through the process of composing comments, proposing changes, and providing rationale for the changes. The final copy of the viewer's comments was sent, as an e-mail, to the editors of the glossary.

Initially, the editors proposed that e-mail created in this way would be sent directly to the exploder list so that members could begin discussing the changes almost immediately. This idea proved unsatisfactory when members started receiving dozens of individual e-mail messages a day relating to the glossary.

To address this problem, the e-mail messages that were generated by the Web pages were collected and sent *only* to the editors for examination and filtering. Periodically, the collected messages, minus the unrelated and inadvertent messages, were sent by e-mail, as a single file, to the members of the exploder list for discussion.

The dynamics of the e-mail software is illustrated in step 1 (top, right-hand side) of Figure 13 above.

5.3 Interactive Editing of the Glossary

When the ITS editors received an e-mail message relating to a glossary comment, whether it was a proposed new entry, a proposed revision to an existing entry, or a comment on someone else's proposal for the glossary, the editors stepped through 6 procedures to address the comment fully:

1. They examined the source of the e-mail comment to see if it was a bona fide source.
2. They examined the comment itself to see if it was bona fide (and not an inadvertent e-mail).
3. They circulated the comment to the entire Ad Hoc Group, in keeping with due process.
4. They collected comments on that e-mail proposal.
5. They worked in e-mail discussions to achieve consensus on the proposed change for the glossary.
6. They recorded the Ad Hoc Group's consensus in the final draft of *Telecom Glossary 2000*.

5.4 Database Management Software

The bulk of the software developed for the glossary concerned its management and grooming. An overview of the tasks in the program called “whole” is shown in Figure 14. A summary of the function of each of the 12 types of tasks is given in the left-hand vertical block in the diagram.

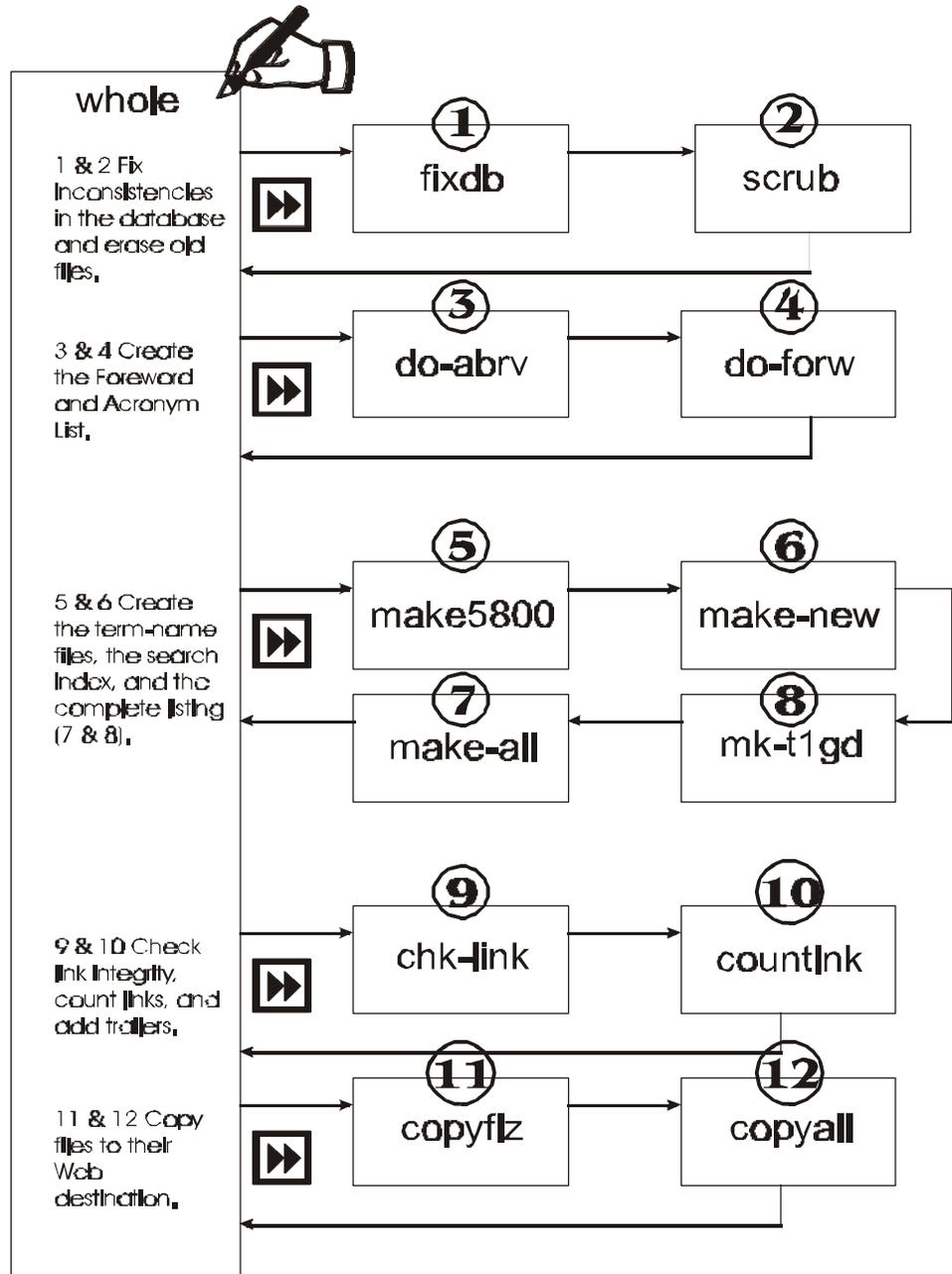


Figure 14. Flowchart of the major tasks in the program called “whole.”

A fuller description of each of the programs (“fixdb,” “scrub,” etc.) is given in Section 5.6, in the subsection corresponding to the number shown above each program’s box. The description of the function of each of the 12 programs is summarized in Table 3 below.

Table 3. A Summary of the Major Functions of Each Program in the “Whole” Program

Program No./Name	Major Function(s) of The Program
1 “fixdb”	This script fixes spacing problems (e.g., removes extra spaces, inserts missing tabs).
2 “scrub”	This script cleans the slate to remove left-over errors, erases old, out-of-date files that contained errors.
3 “do-abrv”	This script creates the abbreviations list (a table) in the <i>Telecom Glossary 2000</i> Appendix.
4 “do-forw”	This script creates the Foreword for <i>Telecom Glossary 2000</i> .
5 “make5800”	This script generates 8090 individual Web pages (one for each entry in the glossary). [Note that when the revision work began, there were only 5800 entries, hence the name of this program.]
6 “make new”	This script was not used in the final stages of glossary development. Originally, it generated a list of new entries, each of which has since been incorporated into the 8090 entries.
7 “make all”	This program creates one very large file of 8090 entries for the “view all–big” hot button.
8 “mk-51gd	This program creates the searchable index to speed the search engine operation and to create the search results.
9 “ck-link	This housekeeping program verifies that each hyperlink generated matches a file name (one of the 8090 entries in the glossary) on disk.
10 “countlnk”	This housekeeping program (which has been used to isolate problems) counts the number of links in the glossary (there are approximately 69,000 hyperlinks) and counts the number of times a given term is linked to any other entry.
11 & 12 “copyflz” & “copyall”	These two programs copy portions of the final Web site.

Most of the editing and creating of the glossary was done by a series of Perl scripts created especially for this project. These Perl scripts were developed to manage and to groom the large database that included not only text and 94 still images, but 4 video clips and 2 audio clips.

Perl is a user-supported programming language designed to manipulate large volumes of text. In fact, its name, **P**ractical **E**xtraction and **R**eport **L**anguage, implies that the language was intended for extracting and manipulating data (both text and numerical) and then creating textual reports from it. As this glossary is primarily a database of words and text, Perl was considered an appropriate language to use [5, 6].

Many of the scripts used in manipulating the glossary contain explicit references to Web address and directory structures. These locations are specific to the hard drives and Web servers at ITS, where the scripts were developed. If these scripts are copied onto other drives and other locations, the mentioned locations will need to be changed within those scripts to allow them to run properly in the new locations.

Within many of the scripts, test code remains in place. This test code was often used to halt a generation prematurely and to view intermediate results. During the creation process, this kind of code was used quite often to monitor new modules. It is deactivated but retained in the final code as an aid to future users of the scripts.

5.4.1 Levels of Database Programming Tasks

As seen in the flow charts in Figures 14 and 15, computer manipulation and updating of the glossary involves several levels of computer programs or scripts. Some scripts addressed the daily, global updating of the glossary to preserve a current copy for both archival/backup and for maintenance purposes. Other programs address fine-tuning such as minute formatting tasks that allow both clean, consistent presentation of definitions and precise, automatic selection of specific character strings to be modified.

Of the global scripts, the script called “whole” is the most important in managing glossary data. The “whole” script calls each of the other scripts to effect a thorough, global update and to perform other designated tasks.

Several large programs were triggered whenever the editors added a new entry or modified an existing entry.

5.5 Interactive Web Pages

Four scripts were developed to process the revisions to the glossary Web pages. The four scripts are: “add def”, “ad def2”, “edit 37d”, and “add-comm.prl”. Each of these scripts is discussed below.

5.5.1 Script “add-def”

This script is the primary script that is activated when a user selects the “Add Comments” button from any of the glossary Web pages. The first thing that the script does is to try to decide which user is submitting a comment.

Whenever any user views any Web page, the user’s browser automatically sends limited information about that user’s machine, such as the IP address, the browser version, and the type of operating system being used. Some of the glossary scripts keep track of this information to identify return visitors to the site.

This script “add-def” keeps a small file that stores a user’s IP address when he/she submits a comment. Additionally, the user is asked for his/her name, e-mail address, and affiliation. This information is stored in the data file and used to tag a user’s comments so that the user can be contacted if more information is needed. When a user logs on for a second time, this information is retrieved from the file and presented to the user for verification.

The script then displays the current definition for the term that the user has selected and presents an editing box, within which the user can modify the definition. When the user has finished editing the definition and selects the “Done” button, processing is passed to the “add-def2” script. The computer programming code for the “add-def” script is shown below.

```
#!/usr/local/bin/perl

# -- Print out the basic header info
print "Content-type: text/html\n\n";
print "<BODY background=\"../gifs/\$-backg2.gif\" text=\"#000000\" bgcolor=\"#ffffff\"
link=\"#FF0000\" vlink=\"#0000ff\" alink=\"#00FF00\">\n";
print "<html><title>Edit This Definition</title>\n";
print "<h1><center>Edit This Definition</center></h1>\n";
print "<hr><font color=\"blue\">\n";

# -- Read in the variables from the command line from Web page
&ReadParse;
# foreach $key (keys %in) {
# print "$key: $in{$key}<br>\n";
# }

foreach $key (keys %ENV) {
# print "$key: $ENV{$key}<br>\n";
#}

open (NAMES, "/nd/bing/files/37dnames.txt");
while ($ipad = <NAMES>) {
chop ($ipad);
```

```

$names{$i pad} = <NAMES>;
chop ($names{$i pad});
$agen{$i pad} = <NAMES>;
chop ($agen{$i pad});
}
close (NAMES);

#foreach $key (keys %names) {
# print "$key: $names{$key}\n";
# };

$rh = $ENV{REMOTE_ADDR};
$nm = $names{$rh};
$anm = $agen{$rh};

$name = $in{filename};
if ($name =~ /^$/ ) { $name="new" };

if ($name eq "new") {
    $in{newold}="new";
    $defname = "new";
} else {
    $in{newold}="old";
    $i=0;
    $name =~ s/^\//nd\bing\public_html\fs-1037d\//;
    open (HTML, $name);
    while ($line = <HTML>) {
        chop ($line);
        if ($line =~ /<body/) {next;};
        if ($line =~ /^$/) {next;};
        if ($line =~ /<title>/) {next;};
        if ($line =~ /<font size/) {last;};
        if ($line =~ /^<b>/ && $i == 0) {
            $defname = $line;
            ++$i;
            while ($defname =~ /<\b>/) {
                $defname =~ s/\. *//;
            }
            $defname =~ s/\. *>//;
        }
        $newline = $newline . $line;
    } # end while HTML
    close (HTML);

    $newline =~ s/([>])<\a>/$1/g;
    $newline =~ s/<a href="\_ \S*html\">//g;
    $newline =~ s/<a href="\t1\S*html\">//g;
    $newline =~ s/<a href="\all\S*html\">//g;
    $newline =~ s/<hr>//;
    $newline =~ s/<\b>\s*$//;
    while ($newline =~ /<p>\s*$/) {
        $newline =~ s/<p>\s*//;
    }
    print "$newline<hr>\n";
} # end if not new

print "<font color=\red\>Enter your new definition in the box below. When you are done, hit the
send button (or back up if you made no changes). Your rewrite will be included in the next rebuild
of the proposed standard and will also be sent to the Revision committee members ";
print "(to use specific formatting codes, read <font color=green><a
href=\ http://www.its.blrdoc.gov/~bing\fs-1037d\help-new.html \"
target=\new\>this<\a><\font>";

print "<FORM METHOD=\POST\" ACTION=\http://www.its.blrdoc.gov/cgi-bin/add-def2.prl \">\n";

```

```

if ($defname eq "new") {
    print "Enter a name for this definition (inside the <b>Bold</b> brackets, use lower case letters)
and then type the definition.\n";
    print "<TEXTAREA wrap=soft rows=20 cols=50 name=\"newdef\">\n";
    print "<b> name: </b> def";
    print "</TEXTAREA>\n";
} else {
    print "<INPUT NAME=\"defname\" TYPE=\"hidden\" VALUE=\"$defname\">\n";
    $newline =~ s/\\"/&quot;/g;
    print "<INPUT NAME=\"oldddef\" TYPE=\"hidden\" VALUE=\"$newline\">\n";
    print "<TEXTAREA wrap=soft rows=20 cols=50 name=\"newdef\">\n";
    print $newline;
    print "</TEXTAREA>\n";
}

    print "<INPUT NAME=\"typeof\" TYPE=\"hidden\" VALUE=\"test\">\n";
    print "<INPUT NAME=\"newold\" TYPE=\"hidden\" VALUE=\"$in{newold}\">\n";
    print "<br>Your Name: <INPUT NAME=\"init\" size=30 VALUE=\"$nm\"><br>\n";
    print "<br>Your Agency: <INPUT NAME=\"agen\" size=30 VALUE=\"$anm\">\n";
    print "<INPUT TYPE=\"submit\" VALUE=\"Send It\">\n";

```

The following Perl code was written by Steven Brenner and was extracted, with permission and without modification, from the cgi library module that he makes available at his Web site

<http://cgi-lib.berkeley.edu/1.14/cgi-lib.pl.txt>

(last accessed on 03-09-01). The permission to use the code is given within the module itself.

```

# -----
# Adapted from cgi-lib.pl by S. E. Brenner@bioc.cam.ac.uk
# Copyright 1994 Steven E. Brenner
sub ReadParse {
    local (*in) = @_ if @_;
    local ($i, $key, $val);

    if ( $ENV{'REQUEST_METHOD'} eq "GET" ) {
        $in = $ENV{'QUERY_STRING'};
    } elsif ( $ENV{'REQUEST_METHOD'} eq "POST" ) {
        read(STDIN, $in, $ENV{'CONTENT_LENGTH'});
    } else {
        # Added for command line debugging
        # Supply name/value form data as a command line argument
        # Format: name1=value1&name2=value2&...
        # (need to escape & for shell)
        # Find the first argument that's not a switch (-)
        $in = ( grep( !/^-/, @ARGV ) ) [0];
        $in =~ s/\\"/&/g;
    }

    @in = split(/&/, $in);

    foreach $i ( 0 .. $#in ) {
        # Convert plus's to spaces
        $in[$i] =~ s/\+ /g;

        # Split into key and value.
        ($key, $val) = split(=/, $in[$i], 2); # splits on the first =.

        # Convert %XX from hex numbers to alphanumeric
        $key =~ s/%(..)/pack("c", hex($1))/ge;
        $val =~ s/%(..)/pack("c", hex($1))/ge;

        # Associate key and value. \0 is the multiple separator
        $in{$key} .= "\0" if (defined($in{$key}));
        $in{$key} .= $val;
    }
}

```

```

    }
    return length($in);
}

```

5.5.2 Script “add-def2”

This script is an extension of the “add-def” script. The user’s proposed definition is parsed and formatted and shown to the user as it would appear in the glossary. The user is prompted to enter a rationale for inclusion of the proposed definition. Once this is done, the proposed new definition (or revision to an existing definition) and rationale, along with the user’s name and e-mail address, are appended to a data file on disk, and a copy of this proposal is sent to the editors of the glossary.

If the user entering a definition is one of the editors of the glossary (this is determined by examining the IP address of the user), the changes are made directly into the master database.

The code for the “add-def2” script is shown below.

```

#!/usr/local/bin/perl

# -- Print out the basic header info
print "Content-type: text/html\n\n";
print "<BODY background=\"../gifs/\"-$-backg2.gif\" text=\"#000000\" bgcolor=\"#ffffff\"
link=\"#FF0000\" vlink=\"#0000ff\" alink=\"#00FF00\">\n";
print "<html><title>This Is The Way Your Definition Will Look</title>\n";
print "<hl><center>This Is The Way Your Definition Will Look</center></hl>\n";
print "<hr><font color=\"green\">\n";

# -- Read in the variables from the command line from Web page
&ReadParse;
foreach $key (keys %in) {
    #print "$key: -${in}{$key}-<br>\n";
}

$in{'olddef'} =~ s/\\"/&quot\;/g;
$in{'newdef'} =~ s/\\"/&quot\;/g;

if (-e "/nd/bing/public_html/fs-1037d/temp-gen/stop.txt") {
    print "<font color=\"red\"><blink>\n";
    print "The database is unavailable right now</blink>. This means someone else has entered
a new definition at exactly the same time you did. Back up in your browser, wait 20 seconds,
and try again. If the problem continues, that means the database is offline for some other
reason. Contact the editors.\n";
    die;
}

if ($in{'inits'} =~ /^$/) {
    print "<font color=\"red\">\n";
    print "You did not enter your name. You need to hit the 'back' button on your browser, enter
your name in the blank, and resend the definition.\n";
    die;
}

if ($in{'edtech'} =~ /^$/ && $in{'typeof'} EQ "final") {
    print "<font color=\"red\">\n";
    print "You did not indicate whether this was a technical or editorial change. You need to
hit the 'back' button on your browser, select one of the options, and resend the
definition.\n";
    die;
}

```

```

}

if ($in{'edtech'} =~ /tech/ && $in{'comments'} =~ /^$/ && $in{'typeof'} EQ "final") {
    print "<font color=\red\>\n";
    print "You did not include any comments for this technical change. You need to hit the 'back'
button on your browser, write comments, and resend the definition.\n";
    die;
}

$_ = $in{'comments'};
$count = tr/ //;
if ($count > 52 && $in{'typeof'} EQ "final") {
    print "<font color=\red\>\n";
    print "You used more than 50 words in your comments. You need to hit the 'back' button on your
browser, reduce your comments, and resend the definition.\n";
    die;
}

if ($in{'olddef'} EQ $in{'newdef'}) {
    print "<font color=\red\>\n";
    print "You have made no changes to the definition. Nothing will be changed.";
    die;
}

$rh = $ENV{'REMOTE_ADDR'};
$nm = $in{'inits'};
$anm = $in{'agen'};

open (NAMES, "/nd/bing/files/37dnames.txt");
while ($ipad = <NAMES>) {
    chop ($ipad);
    $names{$ipad} = <NAMES>;
    $agen{$ipad} = <NAMES>;
    chop ($names{$ipad});
    chop ($agen{$ipad});
}
close (NAMES);
$names{$rh} = $nm;
$agen{$rh} = $anm;

open (NAMES, ">/nd/bing/files/37dnames.txt");
foreach $key (keys %names) {
    print NAMES "$key\n$names{$key}\n$agen{$key}\n";
}
close (NAMES);

$in{'newdef'} =~ s/\&quot\;/\//g;
$in{'newdef'} =~ s/\n//g;
$in{'newdef'} =~ s/<hr>//g;

if ($in{'newold'} eq "new") {
    $in{'defname'} = $in{'newdef'};
    $in{'defname'} =~ s/^\<b>//;
    $in{'defname'} =~ s/\n//g;
    $in{'defname'} =~ s/\.*/;
    while ($in{'defname'} =~ /\<b>/) {
        $in{'defname'} =~ s/<b>.*//;
    }
    $in{'defname'} =~ s/^\s*//;
    $in{'defname'} =~ s/\s*$//;
}

print "<h3><center>$in{'defname'}</center></h3>\n";
print "$in{'newdef'}<hr>\n";

#print "<font color=\blue\>\n";
#foreach $key (keys %ENV) {
#    print "$key: $ENV{$key}<br>\n";
#}

if ($in{'typeof'} EQ "final") {

```

```

print "<font color=\red\">\n";
print "Your definition will be reviewed. Thank you, $nm, of the $in{'agen'}.<p><font
color=\blue\">\n";

system ("date >/nd/bing/public_html/fs-1037d/temp-gen/now.txt");
open (DT, "/nd/bing/public_html/fs-1037d/temp-gen/now.txt");
$dt = <DT>;
chop ($dt);
close (DT);

if ($rh =~ /64\.126/ || $rh =~ /64\.217/) {
$i = 0;
$g = "";
system ("touch /nd/bing/public_html/fs-1037d/temp-gen/stop.txt");
open (INP, "/nd/bing/public_html/fs-1037d/temp-gen/01-37d.txt") || die "Can't open input file";
open (OUT, ">/nd/bing/public_html/fs-1037d/temp-gen/02-37d.txt") || die "Can't open output file";

while ($y = <INP>) {

if ($in{'newold'} EQ "old") {
if ($y =~ /\@\\$in{'defname'}:/) {
$y = $y . "<p>\n<hr>\n====$dt====New Suggestion by $nm of the
$in{'agen'}.<p>\n$in{'newdef'}<p>\n";
++$i;
} # end if defname

} else { # old/new
if ($i == 0 && $y =~ /\@\\$/ ) {
$f = $g;
$g = $y;
$g =~ s/\. *//;
$g =~ s/\. *@\$///g;
$outdef = $in{'newdef'};
$outdef =~ s/^<b>//;
$outdef =~ s/^\s*//;
$h = $outdef;
$h =~ s/\. *//;
$f =~ s/\&\#\#...\.//g;
$g =~ s/\&\#\#...\.//g;
$h =~ s/\&\#\#...\.//g;
$h =~ s/\. {1,3}>//g;
$g =~ s/\. {1,3}>//g;
$f =~ s/\. {1,3}>//g;
$f =~ tr/0/zero/;
$f =~ tr/1/one/;
$f =~ tr/2/two/;
$f =~ tr/3/three/;
$f =~ tr/4/four/;
$f =~ tr/5/five/;
$f =~ tr/6/six/;
$f =~ tr/7/seven/;
$f =~ tr/8/eight/;
$f =~ tr/9/nine/;
$f =~ tr/0/zero/;
$g =~ tr/1/one/;
$g =~ tr/2/two/;
$g =~ tr/3/three/;
$g =~ tr/4/four/;
$g =~ tr/5/five/;
$g =~ tr/6/six/;
$g =~ tr/7/seven/;
$g =~ tr/8/eight/;
$g =~ tr/9/nine/;
$h =~ tr/0/zero/;
$h =~ tr/1/one/;
$h =~ tr/2/two/;
$h =~ tr/3/three/;
$h =~ tr/4/four/;
$h =~ tr/5/five/;
$h =~ tr/6/six/;

```

```

$h =~ tr/7/seven/;
$h =~ tr/8/eight/;
$h =~ tr/9/nine/;

$h =~ tr/A-Z/a-z/;
$g =~ tr/A-Z/a-z/;
$f =~ tr/A-Z/a-z/;
$h =~ s/\W//g;
$g =~ s/\W//g;
$f =~ s/\W//g;
if ($f lt $h && $f ne $h && $g gt $h) {
    ++$i;
    print "<br>This definition inserted between<br>$f<br>$g<br>";
    print OUT "\n<p>\n\n\t\t<b>@\$\$outdef<p>\n";
    print OUT "===<dt>===New Suggestion by $nm of the $\in{'agen'}.<p>\n<p>\n";
} # end if between
if ($f eq $h && $g gt $h) {
    ++$i;
    print "<br>This definition <blink>exists already</blink>. Your definition has been <font
color=red><blink>abandoned</blink></font>. To recover what you wrote, back up one screen, highlight
the text and copy your new words. Find the existing term in the database, click on it, and select
'add to this def'. Edit the existing definition (or paste in your new text).";
    print OUT "\n<p>\n\n\t\t<b>$outdef<p>\n";
    print OUT "===<dt>===New Suggestion by $nm of the $\in{'agen'}.<p>\n<p>\n";
} # end if equal
} # end make sure it's a def
} # end if old/new

print OUT "$y";
} # end while

if ($i == 0) {
    $outdef = $\in{'newdef'};
    $outdef =~ s/^\<b>//;
    $outdef =~ s/^\s*//;
    print OUT "\n<p>\n\n\t\t<b>@\$\$outdef<p>\n";
    print OUT "===<dt>===New Suggestion by $nm of the $\in{'agen'}.<p>\n<p>\n";
}

close (INP, OUT);
chdir ("/nd/bing/public_html/fs-1037d/temp-gen/") || die "Can't change to that directory";

system ("mv 02-37d.txt temp.txt");
system ("mv 01-37d.txt 02-37d.txt");
system ("mv temp.txt 01-37d.txt");
system ("rm /nd/bing/public_html/fs-1037d/temp-gen/stop.txt");
} # End if only editors

if ($rh !~ /64\.126/ && $rh !~ /64\.217/) {
    system ("touch /nd/bing/public_html/fs-1037d/temp-gen/37subm.txt");
    open (OUT, ">>/nd/bing/public_html/fs-1037d/temp-gen/37subm.txt");
    print OUT "\n<p>\n\n\t\t<b>$\in{'defname'}</b><p>\n$\in{'newdef'}\n";
    print OUT "===<dt>===New Suggestion by $nm of the $\in{'agen'}.<p>\n<p>\n";
    close (OUT);
    $u = "\in{'defname'}";
    $u = "/usr/bin/mailx -s'[37D] $nm just added the definition for $u' bing,evie <
/nd/bing/public_html/fs-1037d/temp-gen/37subm.txt";
    print "<p>$u<p>";
    system ("$u");
} # End if not editor

if ($\in{'newold'} EQ "old") {
    open (NEW, ">>/nd/bing/public_html/fs-1037d/temp-gen/updat.hxx");
} else {
    open (NEW, ">>/nd/bing/public_html/fs-1037d/temp-gen/newdef.hxx");
}
print NEW "\in{'newdef'}<p>\n";
if ($\in{'edtech'} EQ "tech") {
    print NEW "===<dt>===<nm>===$\in{'comments'}<hr>\n";
} else {
    print NEW "<hr>\n";
}
}

```

```

print NEW "<hr>\n";
close (NEW);

if ($in{'edtech'} EQ "ed") {
    open (NEW, ">>/nd/bing/public_html/fs-1037d/temp-gen/edit.hxx");
    print NEW "$in{'newdef'}<hr>\n";
} else {
    open (NEW, ">>/nd/bing/public_html/fs-1037d/temp-gen/tech.hxx");
    print NEW "$in{'newdef'}<p>\n";
    print NEW "===&dt===&nm===&in{'comments'}<hr>\n";
}
close (NEW);

    open (NEW, ">>/nd/bing/public_html/fs-1037d/temp-gen/newest.txt");
    print NEW "$in{'newdef'}<p>\n\r";
    print NEW "===&dt===&nm===&in{'comments'}<hr>\n\r";
    close (NEW);
#&n = "mailx -s'[37D] &n just added a new term to pFS-1037C-R1.' rev1@green.its.blrdoc.gov <
/nd/bing/public_html/fs-1037d/temp-gen/newest.txt";
#system (&n);

} else {
    print "<font color=\"blue\">\n";
    print "If you need to change something, use the 'back' button on your browser and resubmit the
corrected copy. If this version looks correct, select whether this is an editorial or technical
change (if technical, explain the reason for the change in 50 words or less), and select the
'finalize' button;\n";
    print "<FORM METHOD=\"POST\" ACTION=\"http://www.its.blrdoc.gov/cgi-bin/add-def2.prl\">\n";
    print "<INPUT TYPE=\"radio\" NAME=\"edtech\" VALUE=ed>Editorial Change Only<br>\n";
    print "<INPUT TYPE=\"radio\" NAME=\"edtech\" VALUE=tech>Technical Changes (fill out box
below)<br>\n";
    print "<INPUT NAME=\"defname\" TYPE=\"hidden\" VALUE=\"&in{'defname'}\">\n";
    print "<TEXTAREA wrap=soft rows=6 cols=50 NAME=\"comments\"></TEXTAREA><br>\n";
    print "<INPUT NAME=\"typeof\" TYPE=\"hidden\" VALUE=final>\n";
    print "<INPUT NAME=\"inits\" TYPE=\"hidden\" VALUE=\"&in{'inits'}\">\n";
    print "<INPUT NAME=\"agen\" TYPE=\"hidden\" VALUE=\"&in{'agen'}\">\n";
    print "<INPUT NAME=\"olddef\" TYPE=\"hidden\" VALUE=\"&in{'olddef'}\">\n";
    print "<INPUT NAME=\"newold\" TYPE=\"hidden\" VALUE=\"&in{'newold'}\">\n";
    &in{'newdef'} =~ s/\\"/&quot;/g;
    print "<INPUT NAME=\"newdef\" TYPE=\"hidden\" VALUE=\"&in{'newdef'}\">\n";
    print "<INPUT TYPE=\"submit\" VALUE=\"Finalize This\">\n";
}

# -----
# Adapted from cgi-lib.pl by S.E.Brenner@bioc.cam.ac.uk
# Copyright 1994 Steven E. Brenner
sub ReadParse {
    local (*in) = @_ if @_;
    local ($i, $key, $val);

    if ( $ENV{'REQUEST_METHOD'} EQ "GET" ) {
        &n = $ENV{'QUERY_STRING'};
    } elsif ( $ENV{'REQUEST_METHOD'} EQ "POST" ) {
        read(STDIN,&n,$ENV{'CONTENT_LENGTH'});
    } else {
        # Added for command line debugging
        # Supply name/value form data as a command line argument
        # Format: name1=value1&name2=value2&...
        # (need to escape & for shell)
        # Find the first argument that's not a switch (-)
        &n = ( grep( !/^-/ , @ARGV )) [0];
        &n =~ s/\\"/&/g;
    }

    @in = split(/&/,&n);

    foreach $i (0 .. $#in) {
        # Convert plus's to spaces
        &n[$i] =~ s/\+ /g;

        # Split into key and value.

```

```

($key, $val) = split(/=/,$in[$i],2); # splits on the first =.

# Convert %XX from hex numbers to alphanumeric
$key =~ s/%(..)/pack("c",hex($1))/ge;
$val =~ s/%(..)/pack("c",hex($1))/ge;

# Associate key and value. \0 is the multiple separator
$in{$key} .= "\0" if (defined($in{$key}));
$in{$key} .= $val;
}
return length($in);
}

```

5.5.3 Script “edit37d”

This is the primary script that the ITS editors use to update the glossary database. The identity of the editor is determined by examining the IP address of the user and comparing it to the valid IP addresses associated with the ITS editors. If a match is not found, the user is denied the ability to edit. If a match is found, the editor is presented with a menu of choices to edit a definition, delete a definition, add a definition, or finalize a definition.

Newly changed or added definitions are parsed, presented to the editor for verification, and then added to the glossary database. New definitions are identified by a mark that shows them as new. In the early stages of the glossary development, viewers of the glossary could easily see which definitions were new and needed examination.

When sufficient time had elapsed for review and acceptance by the Ad Hoc Group and the community at large, the ITS editors used this script to finalize the definition. This finalizing action removed the “new” flag from the definition.

```

#!/usr/local/bin/perl

# - Read in the variables from the command line from Web page
#print "Content-type: text/html\n\n";
&ReadParse;
#foreach $key (keys %in) {
#print "$key: - $in{$key} -<br>\n";
#}
#print "<font color=\"blue\">\n";
#foreach $key (keys %ENV) {
#print "$key: $ENV{$key}<br>\n";
#}
#die;

# -- Print out the basic header info
print "Content-type: text/html\n\n";
print "<BODY background=\"../gif/$-backg2.gif\" text=\"#000000\" bgcolor=\"#ffffff\"
link=\"#FF0000\" vlink=\"#0000ff\" alink=\"#00FF00\">\n";
print "<html><title>Edit A Definition</title>\n";
print "<h3><center>Edit A Definition</h3>\n";
print "<font color=\"green\">\n";
print "<FORM METHOD=\"POST\" ACTION=\"http://www.its.blrdoc.gov/cgi-bin/edit37d.pl\">\n";
print "<table border=1><tr>\n";
print "<th colspan=2>What do you want to do with the selected term?<br>";
print "</th>";

```

```

print "<td><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"list\">List All New\n";
print "<br><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"flist\">List (Full)\n";
print "<br><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"ulist\">Restr. Old\n";
print "<br><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"l1ist\">Restr. New\n";
print "</td>\n";
print "</tr><tr>\n";
print "<td><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"edit\">Edit</td>\n";
print "<td><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"delete\">Delete\n";
print "<br><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"finalize\">Finalize</td>\n";
print "<td>Restricted to:<br><INPUT align=right NAME=\"twolet\" SIZE=6
VALUE=\"${'twolet'}\"></td>\n";
print "</tr></table><hr></center>";

# if (${'termname'} ne "") {
#   print "<INPUT NAME=\"termname\" TYPE=\"hidden\" VALUE=\"${'termname'}\">\n";
# }

if (${'dowhat'} eq "") {'${'dowhat'} = "list";}
if (${'dowhat'} eq "overr") {
  system ("rm /nd/bing/public_html/fs-1037d/temp-gen/stop.txt");
}

${'oldddef'} =~ s/\\"/&quot;/g;
${'newdef'} =~ s/\\"/&quot;/g;

if (-e "/nd/bing/public_html/fs-1037d/temp-gen/stop.txt") {
  print "<font color=\"red\"><blink>\n";
  print "The database is unavailable right now</blink>. The database is off line every
  weekday starting at 6pm MDT for about 8 hours. If you get this message at any other time,
  that means someone else has entered a new definition at exactly the same time you did. Back
  up in your browser, wait 20 seconds, and try again.\n";
  print "<table border=1><tr><td><INPUT TYPE=\"radio\" NAME=\"dowhat\"
  VALUE=\"overr\">Override the Lockout--be sure the lockout happened
  accidentally</td></tr></table>\n";
  print "<INPUT TYPE=\"submit\" VALUE=\"Ready\"> - - - - - \n";
  die;
}

$rh = $ENV{'REMOTE_ADDR'};

open (NAMES, "/nd/bing/files/37dnames.txt");
while ($ipad = <NAMES>) {
  chop ($ipad);
  $names{$ipad} = <NAMES>;
  $agen{$ipad} = <NAMES>;
  chop ($names{$ipad});
  chop ($agen{$ipad});
}
close (NAMES);

if ($rh !~ /64\.\126/ && $rh !~ /64\.\217/) {
  print "You are not one of the senior editors and are not authorized to make final edits to
  definitions. Please use the regular editing features.";
  die;
} else {
  print "<INPUT TYPE=\"submit\" VALUE=\"Ready\"> - - - - - \n";
  print "Welcome, $names{$rh}</font>\n";
  print " - - - - - <INPUT TYPE=\"submit\" VALUE=\"Ready\">\n";
  #print "<br>${'dowhat'}<br>${'termname'}";
  print "<hr>\n";
}

# -----
Si = 0;

```

```

$g = "";
system ("touch /nd/bing/public_html/fs-1037d/temp-gen/stop.txt");
open (INP, "/nd/bing/public_html/fs-1037d/temp-gen/01-37d.txt") || die "Can't open input
file";

if ($in{'dowhat'} eq "list" || $in{'dowhat'} eq "l1ist") {
    system ("touch /nd/bing/public_html/fs-1037d/temp-gen/new37d.txt");
    open (OUT, ">/nd/bing/public_html/fs-1037d/temp-gen/new37d.txt") || die "Can't open
output file";
}

if ($in{'dowhat'} eq "flist") {
    system ("touch /nd/bing/public_html/fs-1037d/temp-gen/new37d.txt");
    open (OUT, ">/nd/bing/public_html/fs-1037d/temp-gen/new37d.txt") || die "Can't open
output file";
    print "<table border=2>";
}

if ($in{'dowhat'} eq "fedit" || $in{'dowhat'} eq "rdelete") {
    system ("touch /nd/bing/public_html/fs-1037d/temp-gen/stop.txt");
    open (OUT, ">/nd/bing/public_html/fs-1037d/temp-gen/02-37d.txt") || die "Can't open
output file";
}

system ("date >/nd/bing/public_html/fs-1037d/temp-gen/now.txt");
open (DT, "/nd/bing/public_html/fs-1037d/temp-gen/now.txt");
$dt = <DT>;
chop ($dt);
close (DT);

$si = 0;
while ($y = <INP>) {

    if ($y =~ /\@\$/) {
        if ($in{'termname'} eq $g && ($in{'dowhat'} eq 'fedit' || $in{'dowhat'} eq 'rdelete'))
        {
            system (touch "/nd/bing/public_html/fs-1037d/temp-gen/log-del.txt");
            open (LOG, ">>/nd/bing/public_html/fs-1037d/temp-gen/log-del.txt") || die "Can't open
output file";
            if ($in{'dowhat'} eq "fedit") {
                $in{'newdef'} =~ s/<p>===/===/g;
                print OUT "$in{'newdef'}\n";
                print "The newer definition for <i>$g</i> has been added.";
                print LOG "***The newer definition for $g has been added by $names{$rh}\n";
                print LOG "$line\n";
                print LOG "$in{'newdef'}\n";
            } else {
                print "The definition for <i>$g</i> has been deleted.";
                print LOG "***The definition for $g has been deleted by $names{$rh}\n";
                print LOG "$line\n";
            } # end if fedit

            close (LOG);
        } else {
            print OUT "$line";
        } # end if f/r

        if ($in{'dowhat'} eq "edit" && $in{'termname'} eq $g) {
            print "<table border=1><tr><td><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"redit\">Let's
See What This Looks Like</td></tr></table><br>\n";
            print "Do not edit the line that starts with \"===\" or remove the \@$/ at the start of
the term name. <p>";
            $line =~ s/\^"/^&quot;/g;
            print "<INPUT NAME=\"termname\" TYPE=\"hidden\" VALUE=\"$g\">\n";
        }
    }
}

```

```

print "<INPUT NAME=\"ol ddef\" TYPE=\"hi dden\" VALUE=\"$line\">\n";
print "<TEXTAREA wrap=soft rows=20 cols=50 name=\"newdef\" align=center>\n";
print $line;
print "</TEXTAREA>\n";
print "<INPUT TYPE=\"submi t\" VALUE=\"Ready\">\n";
last;
}

if ( $in{' dowhat'} eq "redit" && $in{' termname'} eq $g) {
print "<table border=1><tr><td><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"fedit\">I
like what I see. Make the Change</td></tr></table>\n";
print "<INPUT NAME=\"termname\" TYPE=\"hi dden\" VALUE=\"$g\">\n";
print "<INPUT NAME=\"newdef\" TYPE=\"hi dden\" VALUE=\"$in{' newdef'}\">\n";
print "$in{' newdef'}\n";
}

if ( $in{' dowhat'} eq "finalize" && $in{' termname'} eq $g) {
print "<table border=1><tr><td><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"fedit\">I
like what I see. Make the Change</td></tr></table>\n";
print "<INPUT NAME=\"termname\" TYPE=\"hi dden\" VALUE=\"$g\">\n";
$line =~ s/===.*/;/;
$line =~ s/"/"/&quot;/g;
print "<INPUT NAME=\"newdef\" TYPE=\"hi dden\" VALUE=\"$line\">\n";
print "$line\n";
}

if ($in{' dowhat'} eq "flist") {
if ($line =~ /===/) {
$line =~ s/@\$//;
print "<tr><td><INPUT TYPE=\"radio\" NAME=\"termname\" VALUE=\"$g\">$line</td></tr>\n";
print OUT "$g\n";
}
}

if ($in{' dowhat'} eq "list") {
if ($line =~ /===/) {
print "<INPUT TYPE=\"radio\" NAME=\"termname\" VALUE=\"$g\">$g<br>\n";
print OUT "$g\n";
}
}

if ($in{' dowhat'} eq "l1ist") {
if ($line =~ /===/ && $g =~ /^$in{' twolet'}/i) {
print "<INPUT TYPE=\"radio\" NAME=\"termname\" VALUE=\"$g\">$g<br>\n";
print OUT "$g\n";
}
}

if ($in{' dowhat'} eq "ulist") {
if ($line !~ /===/ && $g =~ /^$in{' twolet'}/i) {
print "<INPUT TYPE=\"radio\" NAME=\"termname\" VALUE=\"$g\">$g<br>\n";
print OUT "$g\n";
}
}

if ($in{' dowhat'} eq "delete" && $in{' termname'} eq $g) {
print "<table border=1><tr><td><INPUT TYPE=\"radio\" NAME=\"dowhat\" VALUE=\"rdelete\">I
Really Want to Delete This</td></tr></table>\n";
print "<INPUT NAME=\"termname\" TYPE=\"hi dden\" VALUE=\"$g\">\n";
print "$line\n";
last;
}

$line = $y;

```

```

    $g = $y;
    $g =~ s/:\.*/;
    $g =~ s/.*@\$/g;
    $g =~ s/<\b>/i;
    } else { # else if @$
    $line = "$line" . "$y";
    } # end @$

} # end while

print "</table>";
if ($y !~ /===/ && ($in{'dowhat'} eq "rdelete" || $in{'dowhat'} eq "fedit")) {
    print OUT "$line\n";
}

close (INP, OUT);

# -----
    if ($in{'dowhat'} eq "fedit" || $in{'dowhat'} eq "rdelete") {
        chdir ("/nd/bing/public_html/fs-1037d/temp-gen/") || die "Can't change to that
directory";
        system ("mv 02-37d.txt temp.txt");
        system ("mv 01-37d.txt 02-37d.txt");
        system ("mv temp.txt 01-37d.txt");
    }

    system ("rm /nd/bing/public_html/fs-1037d/temp-gen/stop.txt");

# -----
# Adapted from cgi-lib.pl by S. E. Brenner@bioc.cam.ac.uk
# Copyright 1994 Steven E. Brenner
sub ReadParse {
    local (*in) = @_ if @_;
    local ($i, $key, $val);

    if ( $ENV{'REQUEST_METHOD'} EQ "GET" ) {
        $in = $ENV{'QUERY_STRING'};
    } elsif ( $ENV{'REQUEST_METHOD'} EQ "POST" ) {
        read(STDIN, $in, $ENV{'CONTENT_LENGTH'});
    } else {
        # Added for command line debugging
        # Supply name/value form data as a command line argument
        # Format: name1=value1&name2=value2&...
        # (need to escape & for shell)
        # Find the first argument that's not a switch (-)
        $in = ( grep( !/^-/ , @ARGV )) [0];
        $in =~ s/\\&/&/g;
    }

    @in = split(/&/, $in);

    foreach $i (0 .. $#in) {
        # Convert plus's to spaces
        $in[$i] =~ s/^\+/ /g;

        # Split into key and value.
        ($key, $val) = split(=/, $in[$i], 2); # splits on the first =.

        # Convert %XX from hex numbers to alphanumeric
        $key =~ s/%(..)/pack("c", hex($1))/ge;
        $val =~ s/%(..)/pack("c", hex($1))/ge;

        # Associate key and value. \0 is the multiple separator
        $in{$key} .= "\0" if (defined($in{$key}));
    }
}

```

```

        $in{$Skey} .= $val;
    }
    return length($in);
}

```

5.5.4 Script “add-comm.prl”

This script was developed to allow the users to send feedback to the editors of the glossary. It was different from the “add-def” script described earlier in that the user was not prompted for specific changes to be made to the definition. The script was developed simply to allow the presentation of generalized comments about the definition.

The script asked the user for his/her name and contact information and provided a space for the user’s comments. The script “remembered” what definition the user was viewing when the comment button was selected, and added that information to the user’s comment.

The comments were collected in a data file on disk and also e-mailed to the ITS editors of the document, who then mailed them to the entire Ad Hoc Glossary Group, where ITS served as convener.

Late in the glossary-development process this script was used only when official comments on the document were required. The script could be adapted to collect a file of casual comments after the glossary goes on line permanently.

```

#!/usr/local/bin/perl

# -- Print out the basic header info
print "Content-type: text/html\n\n";
print "<BODY background=\"../gifs/\$-backg2.gif\" text=\"#000000\"
bgcolor=\"#ffffff\" link=\"#FF0000\" vlink=\"#0000ff\" alink=\"#00FF00\">\n";
print "<html><title>Submit Comments on Telecom Glossary 2000</title>\n";
print "<h1><center>Submit Comments on Telecom Glossary 2000</center></h1>\n";
print "<hr>\n";
print "<font color=red>Note:</font> Comments submitted via this form will <font
color=red><b>not</b></font> be part of the official T1 Letter-Ballot process. These
comments below will be sent to the editors of the glossary and will be reviewed on
an informal basis. <p> <b>If you wish your comments to be part of the T1 Letter-Ballot
process, you must send them directly to <a
href=\"mailto:t1ballot@atis.org,bing@its.blrdoc.gov,evie@its.blrdoc.gov?subje
ct=T1 Letter Ballot Comments on Telecom Glossary 2000\">t1ballot@atis.org</a>. </b>";
print "<hr><font color=\"blue\">\n";

# -- Read in the variables from the command line from Web page
&ReadParse;
# foreach $key (keys %in) {
# print "$key: $in{$key}<br>\n";
# }
# print "<hr>";

#foreach $key (keys %ENV) {
# print "$key: $ENV{$key}<br>\n";
# }

```

```

# print "<hr>";

open (NAMES, "/nd/bing/files/37dnames.txt");
while ($ipad = <NAMES>) {
    chop ($ipad);
    $ipad =~ s/\s*$//;
    $names{$ipad} = <NAMES>;
    chop ($names{$ipad});
    $agen{$ipad} = <NAMES>;
    chop ($agen{$ipad});
    $email{$ipad} = <NAMES>;
    chop ($email{$ipad});
}
close (NAMES);

#foreach $key (keys %names) {
# print "$key: $names{$key}<br>\n $agen{$key}<br>\n $email{$key}<br>\n";
#}
# print "<hr>";

$rh = $ENV{REMOTE_ADDR};
$nm = $names{$rh};
$anm = $agen{$rh};
$em = $email{$rh};
#print "$rh: Hello $nm of $anm<br>\n";
# print "<hr>";

if ($nm =~ /^$/) {$nm = $names{$rh} = $in{inits}}
if ($anm =~ /^$/) {$anm = $agen{$rh} = $in{agen}}
if ($em =~ /^$/) {$em = $email{$rh} = $in{email}}

open (NAMES, ">nd/bing/files/37dnames.txt");
foreach $key (keys %names) {
    print NAMES "$key\n$names{$key}\n$agen{$key}\n$email{$key}\n";
}
close (NAMES);

$name = $in{filename};
if ($name =~ /^$/) { $name="new" };

if ($name eq "new") {
    $in{newold}="new";
    $defname = "new";
} else {
    $in{newold}="old";
    $i=0;
    $name =~ s/^\//nd\bing\public_html\fs-1037d\//;
    open (HTML, $name);
    while ($line = <HTML>) {
        chop ($line);
        if ($line =~ /<body/) {next;};
        if ($line =~ /^$/) {next;};
        if ($line =~ /<title>/) {next;};
        if ($line =~ /<font size/) {last;};
        if ($line =~ /<b>/ && $i == 0) {
            $defname = $line;
            ++$i;
            while ($defname =~ /<\b>/) {
                $defname =~ s/:.*//;
            }
            $defname =~ s/. *>//;
        }
        $newline = $newline . $line;
    } # end while HTML

```

```

close (HTML);

$newline =~ s/([>])</a>/$1/g;
$newline =~ s/<a href=\"\_\S*html\">//g;
$newline =~ s/<a href=\"t1\S*html\">//g;
$newline =~ s/<a href=\"all\S*html\">//g;
$newline =~ s/<hr>//;
$newline =~ s/</b>\s*$//;
while ($newline =~ /<p>\s*$/) {
    $newline =~ s/<p>\s*//;
}
print "$newline</b></i><hr>\n";
} # end if not new

if ($in{typeof} =~ /final/) {
    system ("date >>/nd/bing/files/comments.txt");
    open (COMM ">>/nd/bing/files/comments.txt");

    print COMM "\n$nm of the $anm (Email: $enm) said, \n";
    if ($in{filename} =~ /^$/) {
        print COMM " about the Glossary in general, \n";
    } else {
        print COMM " about the term \"$in{filename}\", \n";
    }
    print COMM "\n\"$in{comments}\" \n\n----- \n";
    print "Thank you for the comments.";
    die;
}

    print "<FORM METHOD=\"POST\" ACTION=\"http://www.its.bldrdoc.gov/cgi-bin/add-
comm.prl\">\n";

    if ($in{comments} =~ /^$/) {
        if ($in{filename} =~ /^$/) {
            print "<font color=\"red\">Enter your comments on the Glossary in general and hit
the \"send\" button. If you have comments on a specific term, go to that term and
submit comments via the \"comments\" button at that term </font>";
            print "<TEXTAREA wrap=soft rows=10 cols=50 name=\"comments\">\n";
            print "</TEXTAREA>\n";
        } else {
            print "<font color=\"red\">Enter your comments on ths term and hit the \"send\"
button.</font>";
            print "<TEXTAREA wrap=soft rows=10 cols=50 name=\"comments\">\n";
            print "</TEXTAREA>\n";
        }
    } else {
        print "<font color=\"green\">Your comments are shown below. If they are OK, hit the
\"send\" button. Otherwise, use the \"Back\" button on your browser and edit
them </font>";
        print "<hr>$nm of the $anm (Email: $enm) said, <ul><li>$in{comments}</ul><hr>";
        print "<INPUT NAME=\"comments\" TYPE=\"hidden\" VALUE=\"$in{comments}\">\n";
        print "<INPUT NAME=\"typeof\" TYPE=\"hidden\" VALUE=\"final\">\n";
    }

    print "<INPUT NAME=\"filename\" TYPE=\"hidden\" VALUE=\"$in{filename}\">\n";
    print "<INPUT NAME=\"newold\" TYPE=\"hidden\" VALUE=\"$in{newold}\">\n";
    print "<br>Your Name: <INPUT NAME=\"init\" size=30 VALUE=\"$nm\">\n";
    print "<br>Your Agency: <INPUT NAME=\"agen\" size=30 VALUE=\"$anm\">\n";
    print "<br>Your Email: <INPUT NAME=\"email\" size=50 VALUE=\"$enm\">\n";
    print "<INPUT TYPE=\"submit\" VALUE=\"Send It\">\n";

# -----
# Adapted from cgi-lib.pl by S.E. Brenner@bioc.cam.ac.uk

```

```

# Copyright 1994 Steven E. Brenner
sub ReadParse {
    local (*in) = @_ if @_;
    local ($i, $key, $val);

    if ( $ENV{'REQUEST_METHOD'} eq "GET" ) {
        $in = $ENV{'QUERY_STRING'};
    } elsif ( $ENV{'REQUEST_METHOD'} eq "POST" ) {
        read(STDIN, $in, $ENV{'CONTENT_LENGTH'});
    } else {
        # Added for command line debugging
        # Supply name/value form data as a command line argument
        # Format: name1=value1&name2=value2&...
        # (need to escape & for shell)
        # Find the first argument that's not a switch (-)
        $in = ( grep( !/^-/, @ARGV ) ) [0];
        $in =~ s/\\&/&/g;
    }

    @in = split(/&/, $in);

    foreach $i ( 0 .. $#in ) {
        # Convert plus's to spaces
        $in[$i] =~ s/+/ /g;

        # Split into key and value.
        ($key, $val) = split(/=/, $in[$i], 2); # splits on the first =.

        # Convert %XX from hex numbers to alphanumeric
        $key =~ s/%(..)/pack("c", hex($1))/ge;
        $val =~ s/%(..)/pack("c", hex($1))/ge;

        # Associate key and value. \0 is the multiple separator
        $in{$key} .= "\0" if (defined($in{$key}));
        $in{$key} .= $val;
    }
    return length($in);
}

```

5.6 Daily/Nightly Scripts

The process of creating a unique Web page for each of the 8000 definitions in the glossary was handled by a series of Perl scripts that were run every night. Each script had a specific task and was executed in a precise order to create 8090 interlinked Web pages (~69,000 individual links).

Each script, listed below, should have a file extension of “pl” or “prl” when stored on a disk drive (this indicates to the system that it is a Perl script and the Perl language interpreter must be used to execute the script). For this project, the extension “prl” was used exclusively.

5.6.1 Script “whole”

The controlling script for the regeneration process is called “whole” (short for “do every step in the whole creation process”). It is a simple script that does nothing more than run every other script in a predetermined order and then log the date and time that each step was completed. This process is called the regeneration cycle. One regeneration cycle can take several hours to complete.

The scripts that ran during the regeneration cycle are detailed later in this section.

One of the important aspects of script “whole” is that the first thing it does is to create a file on the disk called “stop.txt”. The existence of this file will prevent users from modifying the database for ~10 minutes while the glossary begins its regeneration. After the initial few scripts are completed, script “whole” will remove the ban on modifications by deleting the file “stop.txt”.

Another significant aspect of this script is that it will not run if it determines that there is no reason to run. In other words, the system examines the date and time that the database file was last modified. If it determines that the database has not been modified in the last 24 hours, the “whole” script will not be run.

The system administrator may manually run the script “whole” whenever it seems necessary. Prospective users of this script are cautioned, however, *not* to run the script if another copy of the script is running (or if another copy is due to start automatically before this copy finishes).

```
#!/usr/local/bin/perl
$usedir = "/nd/bing/public_html/fs-1037d";

chdir("$usedir/temp-gen") || die "Cannot change to that directory";

system("touch stop.txt");
system("fixdb.pl");
print "\n=== 01 ===\n";
system("date >>now-end.txt");
system("scrub.pl");
system("date >now-end.txt");
system("do-forw.pl &");
print "\n=== 04 ===\n";
system("do-abrv.pl &");
print "\n=== 04a ===\n";
system("rm stop.txt");
system("make5800.pl");
print "\n=== 05 ===\n";
system("date >>now-end.txt");
system("chk-link.pl");
print "\n=== 06 ===\n";
system("date >>now-end.txt");
system("make-new.pl");
print "\n=== 07 ===\n";
system("date >>now-end.txt");
system("num-html.pl");
print "\n=== 08 ===\n";
system("mk-t1gd.pl");
print "\n=== 09 ===\n";
# system("date >>now-end.txt");
system("make-all.pl");
system("date >>now-end.txt");
system("copyflz.pl");
system("copyall.pl");
system("countlnk.pl");
print "\n=== 10 ===\n";
system("date >>now-end.txt");
```

5.6.2 Script “fixdb”

The script “fixdb” is designed to perform minor format alterations to the master database file. Only minor changes are allowed to the database file during this step to avoid accidentally corrupting the database beyond easy repair.

If, for example, it was decided that all occurrences of the time unit *second* should be changed to the abbreviation *s*, script “fixdb” could easily be modified to do that. But, perhaps several weeks later, the database administrator would discover the unfortunate flaw in that decision. That would be that the word *second* appears many times in the glossary when not referring to time (e.g., the *second* item in a list or a *second* example of something). Since several weeks likely have passed since the global change was made, restoring an old backup copy of the database would not be a useful option, because many valid changes will probably have been made to the database during the interval. The database administrator is now faced with the task of searching the database manually and restoring the non-appropriate abbreviations.

Script “fixdb” made only a few minor alterations to the database. One alteration was that it analyzed the beginning of each term and attempted to remove inconsistent spacing at the front of each term name and to replace it with a standard string of characters. Also, the characters “@\$” were moved (if necessary) to the beginning of the term name. (The string “@\$” is used to indicate the end of the current definition and the start of the next term name. This was chosen as a unique string that would be very unlikely to appear in any valid telecommunication definition.)

Script “fixdb” also moved the period at the very end of any sentence to a position inside a “bold off” code, if such a period (“.”) appeared at the end of the definition. Also, the script removed blank lines from within the database.

```
#!/usr/local/bin/perl

$usedir = "/nd/bing/public_html/fs-1037d";
chdir ("$usedir/temp-gen");
$i = 0;
open (INP, "01-37d.txt");
open (OUT, ">02-37d.txt") || die "Can't open output file";
while ($y = <INP>) {

    if ($y =~ /\@\$/) {
        ++$i;
        $j = 0;
        while ($y =~ /\^s/ || $y =~ /\^<b>/ || $y =~ /\@\$/) {
            $y =~ s/\^s//;
            $y =~ s/\^<b>//;
            $y =~ s/\@\$/;
            ++$j;
            if ($j == 10) {last;};
        }
        $y = "\t\t<b>@" . $y;
    }
}
```

```

}

$y =~ s/<i \>\s*\<\b\>\s*\<\b\> \<i \>/g;
$y =~ s/<\b\>\. \. \<\b\>/g;
$y =~ s/\( /\(/g;
$y =~ s/[ /\(/g;
$y =~ s/\s{2}/ /g;
$y =~ s/\s*\n/;
$y =~ s/^ \t\t/;
$y =~ s/&quot;\;/"/g;

$y =~ s/^\t\t<p>//;
$y =~ s/^\t\t<hr>//;
if ($y !~ /^$/ && $y !~ /^
S/) {
    print OUT "$y";
}

} # end while
system ("mv 02-37d.txt temp.txt");
system ("mv 01-37d.txt 02-37d.txt");
system ("mv temp.txt 01-37d.txt");
open (DONE, ">>now-end.txt");
print DONE "\n*** FIX DB Done ***\n";
close (DONE);

```

5.6.3 Script “scrub”

As the name implies, the script “scrub” attempted to clean up leftover files from previous and aborted regeneration attempts and prepared the system for the next generation cycle.

The first step was to copy the master database file, *01-37d.txt*, to a working copy of the database, *03-37d.txt*. This was done to minimize the loss from a catastrophic system failure while manipulating the database file or from programming errors that may accidentally destroy or erase the database file. If either of these things were to happen, the administrator could take some comfort in the fact that the master database wasn’t destroyed—only a copy of it was.

Also, by using a copy of the master database file to generate the new glossary files, the administrator was free to edit the original database with the knowledge that he/she would not be interfering with the regeneration process. The only minor downside to this system was that any changes made to the database, during a regeneration cycle, would not appear in the final glossary files until the subsequent generation cycle had started and then finished (usually the next day).

The script “scrub” then began erasing all previously generated hypertext glossary files. Since there were too many files to allow the UNIX operating system to globally delete them (it has a practical limit of being able to manipulate only a few hundred files at once), the files were deleted alphabetically in groups. The script also systematically erased hypertext files in several glossary subdirectories where they might have been created by the regeneration process.

The script “scrub” did not directly erase any files that were not generated hypertext files. That is why most master files related to the regeneration process used the file extension “.txt”.

```
#!/usr/local/bin/perl
  $usedir = "/nd/bing/public_html/fs-1037d";
  chdir ("$usedir/temp-gen") || die "Cannot change to that directory";
  system ("cp 01-37d.txt 03-37d.txt");

@_ = ("_", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z", ".", "0", "1", "9");

# ----- Delete Old Files -----
foreach (<all_*>) { unlink }

foreach $r1 (@letters) {
  foreach $r2 (@letters) {
    foreach (<_${r1}${r2}.htm*>) { unlink }
  }
  foreach (<_${r1}.htm*>) { unlink }
}

foreach (<_*>) { unlink }

chdir ("$usedir") || die "Cannot change to that directory";

# ----- Delete Old Files -----
foreach (<all_*>) { unlink }

foreach $r1 (@letters) {
  foreach $r2 (@letters) {
    foreach (<_${r1}${r2}.htm*>) { unlink }
  }
  foreach (<_${r1}.htm*>) { unlink }
}

# ----- Delete Directories -----
while ($dirname = <dir-???) {
  system ("rm -R $dirname");
}

chdir ("/green2/htdocs/projects/telecomglossary2000") || die "Cannot change to that directory";

# ----- Delete Old Files -----
while ($dirname = <dir-???) {
  system ("rm -R $dirname");
}

#foreach (<all_*>) { unlink }

#foreach $r1 (@letters) {
# foreach $r2 (@letters) {
#   foreach (<_${r1}${r2}.htm*>) { unlink }
# }
#   foreach (<_${r1}.htm*>) { unlink }
# }
```

5.6.4 Script “do-forw”

This script read in the text file “t1g-forw.txt”, which contains the basic text that will become the foreword for the glossary. The script then did some minor processing on each line and created the HTML file, “t1g-forw.html.” This file contains information that glossary users will likely read first.

The minor formatting changes included moving spaces, formatting codes, and grooming the presentation of punctuation marks. The script also contains a module to hyperlink any named definition that occurred in the foreword to its definition in the main part of the glossary. That hyperlinking module was disabled for the final draft, when it was determined that too many hyperlinks in the foreword were distracting to the reader.

```
#!/usr/local/bin/perl
$usedir = "/nd/bing/public_html/fs-1037d";
chdir ("$usedir/temp-gen") || die "Cannot change to that directory";

$sh = ". /t1g-forw";
$sh1 = "$sh" . ".txt";
$sh2 = ">" . "$sh" . ".html";
$sh3 = ">" . "$sh2";

open (TERM, ". /37d-list.txt");
while ($t1 = <TERM>) {
    chop ($t1);
    last if ($t1 =~ /^$/);
    $t1 =~ s/\(. *\)//g;
    $t1 =~ s/[. *i\ ]//g;
    $t1 =~ s/^\s*//;
    $t1 =~ s/\s*$//;
    $t1 =~ s/\s{2,}/ /g;
    $t1 =~ tr/[A-Z]/[a-z]/;
    push (@wordlist, $t1);
}

@wordlist = reverse (sort {length ($a) <=> length ($b)} @wordlist);
close (TERM);

open (DEFS, "$sh1");
open (OUT, "$sh2");
while ($d1 = <DEFS>) {
    chop ($d1);
    if ($d1 !~ /^$/) {

        $d1 =~ s/</ </g;           # Adds space before and after <> so that
        $d1 =~ s/>/> /g;         # there will be boundaries around words.
        $d1 =~ s/\s{1,}/ /g;     # Removes more than 1 spaces
        $d1 =~ s/^\s*//;         # Removes leading spaces
        $d1 =~ s/\s*$//;         # Removes lagging spaces

        foreach $t2 (@wordlist) {
            last; # Stops word linking

            $t3 = $t1 = $t2;
            if ($t1 =~ /^0/) {
                $t1 =~ s/^0//;
                $t3 = $t1;
                $t1 =~ s/0/-/g;
                $t3 =~ s/0/_/g;
            }
        }
    }
}
```

```

St3 =~ s/\s+/_/g;          # Replaces spaces with "_"
St3 =~ s/\'/_/g;          # Replaces " ' " with "_"
St3 =~ s/\/_/_/g;        # Replaces " / " with "_"

$w2 = $d1;                 # splits word and counts "<" and ">"

$_ = $w2;                 # splits word and counts "</a" and "<a"
$y = /\b$t1\b/i;         # on each side to see if word nested
$_ = $w3 = $`;           # in another link.

if ($y > 0) {             # Substitutes if match found.

$count1 = s/<a/<a/g + 1;
$_ = $w3;
$count2 = s/<\a/<\a/g + 1;

if ($count1 eq $count2) {

    $w2 =~ s@(\b)($t1)(\b)@$1<a href="_$t3.html">$2</a>$3@i;
    $d1 = "$w1" . "$w2";

} # End count matches
} # End contains phrase

} # End wordlist

$d1 =~ s/([bBiI])>\s*<([\ ])/$1><$2/g;
$d1 =~ s/<b><\i>/<\i><b>/g;
$d1 =~ s/<i><\b>/<\b><i>/g;
$d1 =~ s/<i>(\s*)/$1<i>/g;
$d1 =~ s/(\s*)<\i>/<\i> /g;
$d1 =~ s/<b>(\s*)/$1<b>/g;

$d1 =~ s/\s*<sub>/<sub>/g;
$d1 =~ s/<sub>\s*/<sub>/g;
$d1 =~ s/\s*<\sub>/<\sub>/g;
$d1 =~ s/<\sub>\s*/<\sub>/g;
$d1 =~ s/\s*<sup>/<sup>/g;
$d1 =~ s/<sup>\s*/<sup>/g;
$d1 =~ s/\s*<\sup>/<\sup>/g;
$d1 =~ s/<\sup>\s*/<\sup>/g;

$d1 =~ s/(\s*)<\b>/<\b> /g;
$d1 =~ s/<b><b>/<b>/g;
$d1 =~ s/(<\/[bBiI]>)/$1 </g;
$d1 =~ s/ {1,} / /g;
$d1 =~ s/^\s*//;
$d1 =~ s/xx/i g;          # Remove "suppress" codes

open (OUT, "Sh3");
print OUT "$d1\n";
close (OUT);
}
}

```

5.6.5 Script “do-abrv”

This script reads in the text file, “t1g-abrv.txt” and creates the hypertext file “t1g-abrv.html” from it. This hypertext file is the list of acronyms and abbreviations for the glossary. This hyperlinked list, which is presented as an appendix, is displayed as a very large table.

This script does some minor formatting to the spacing and punctuation in the hypertext document. It also creates hyperlinks to definitions within the body of the glosssary.

```
#!/usr/local/bin/perl

$usedir = "/nd/bing/public_html/fs-1037d";
chdir ("$usedir/temp-gen") || die "Cannot change to that directory";

$sh = ". /t1g-abrv";
$sh1 = "$sh" . ".txt";
$sh2 = ">" . "$sh" . ".html";
$sh3 = ">" . "$sh2";

open (DEFS, "$sh1");
open (OUT, "$sh2");
while ($d1 = <DEFS>) {
    chop ($d1);
    print "$d1\n";
    if ($d1 =~ /<b>/) {

        $d2 = "$d1";
        $d2 =~ s/<b>(.*)/$1/;
        $d2 =~ s/(.*)<\b>/$1/;
        $d2 =~ s/<.{1,3}>/g;
        $d2 =~ s/[dep.*]//;
        $d2 =~ s/&.*//;
        $d2 =~ s/{.*}//;
        $d2 =~ s/(.*)//;
        $d2 =~ s/\\//g;
        $d2 =~ s/[//g;
        $d2 =~ s/^\\s*//;
        $d2 =~ s/\\s*$//;
        $d2 =~ s/\\s+/_/g;

        $d2 =~ s/'/_/g;          # Replaces " ' " with "_"
        $d2 =~ s//_/g;          # Replaces " / " with "_"
        $d2 =~ tr/[A-Z]/[a-z]/;

        $d1 =~ s@<b>(.*)<\b@<b><a href="_$d2.html">$1</a><\b@i;
    }

    $d1 =~ s/##/<b>/g;
    $d1 =~ s/#\b#/<\b>/g;
    $d1 =~ s/\\s*<sub>/<sub>/g;
    $d1 =~ s/<sub>\\s*/<sub>/g;
    $d1 =~ s/\\s*</sub>/</sub>/g;
    $d1 =~ s/</sub>\\s*/</sub>/g;
    $d1 =~ s/\\s*<sup>/<sup>/g;
    $d1 =~ s/<sup>\\s*/<sup>/g;
    $d1 =~ s/\\s*</sup>/</sup>/g;
    $d1 =~ s/</sup>\\s*/</sup>/g;

    $d1 =~ s/(\\s*)<\b>/<\b> /g;
    $d1 =~ s/<b><b>/<b>/g;
    $d1 =~ s/(<\\/[bBi I]>)/</$1 /g;
    $d1 =~ s/ {1,} /g;
    $d1 =~ s/^\\s*//;

    open (OUT, "$sh3");
    print OUT "$d1\n";
    close (OUT);
}
}
```

5.6.6 Script “make5800”

This script is, by far, the largest script in the library. It is the heart of the regeneration process. All scripts discussed so far have prepared the system and the database for script “make5800” to create the glossary files. All the scripts that run after it, analyze and manipulate the files that script “make5800” has created. (The name of the script “make5800” derives from the computer code developed to execute these same processes for the earlier version of the glossary, Federal Standard 1037C, *Glossary of Telecommunication Terms*, which contained some 5800 entries. The current glossary, *Telecom Glossary 2000*, contains some 8090 entries, but the script name was not changed to reflect this larger database.)

The script is complex, so the descriptions of its code will be distributed throughout the listing. The initialization functions are begun with the following lines of code:

```
#!/usr/local/bin/perl
$usedir = "/nd/bing/public_html/fs-1037d";
chdir("$usedir/temp-gen") || die "Cannot change to that directory";
system("cp 01-37d.txt 03-37d.txt");

@letters =
("_", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v",
"w", "x", "y", "z", ".", "0");
```

These next few lines of code will open up two files on the hard drive, “37dnew1.txt” and “37dnew2.txt”. These two files contain a listing of new terms for the glossary and new T1 terms, respectively. Each term name in each file is added to an internal array and later used to create colored tags to indicate to the reader of the glossary when a term is a new term or from T1 sources. These tags appear in the final version of the glossary.

```
# ----- Make ALLNew List -----
open (ALLN, "37dnew1.txt");
while ($i = <ALLN>) {
  chop($i);
  if (length($i) < 3) {++$ln};
  if ($ln > 0 && $i =~ /:/) {
    $ln = 0;
    $i =~ s/:.*/g;
    $i =~ s/^\s*/;
    push (@new1, $i);
  }
}
close (ALLN);

open (ALLN, "37dnew2.txt");
while ($i = <ALLN>) {
  chop($i);
  if (length($i) < 3) {++$ln};
  if ($ln > 0 && $i =~ /:/) {
    $ln = 0;
    $i =~ s/:.*/g;
    $i =~ s/^\s*/;
    push (@new2, $i);
  }
}
```

```

}
}
close (ALLN);

```

The following code creates the tag line that will eventually be appended to the end of every HTML file in the glossary. It currently contains the date the glossary was last generated and a reminder to the viewer as to where the references can be found. The tag line can be modified to include almost any information the user wishes.

```

# ----- Date Code -----
system ("date >now.txt");
open (DT, "now.txt");
while ($dt = <DT>) {
  chop ($dt);
  $tf = "</b><hr>\n <font size = \"-1\">\n This <a href=\"_html.html\" target =
\"main1\">HTML</a> version of <a href=\"t1g2k.html\" target=\"_top\">Tel ecom Glossary
2K</a> was last generated on " . $dt. <a href=\"t1g-forw.html#refs\">References</a>
can be found in the Foreword.<hr>";
}
close (DT);

# ----- Testing -----
if (0) {
  $a = "_";
  $b = "$a" . ".html";
  while (-e "$b") {
    unlink ("$b");
    $a = "$a" . "-x";
    $b = "$a" . ".html";
  } die; }

```

The following code creates the alphabetically arranged index files from which the user will select terms for viewing. An index file is generated for each letter of the alphabet. Each index is displayed, within the glossary, in the right-most frame of the user's browser. The text immediately below creates the header information for each index file. Some special coding is required for the letters A and Z.

Later, as each hypertext file is created from the terms found in the database, the associated term name is appended to the corresponding index file for the appropriate letter of the alphabet.

```

# ----- Create New "All" Files -----
foreach $file (@letters) {
  # print "$file = ";
  if ($file1 =~ /[a-z]/) {
    $d6 = "all_" . "$file1" . ".html";
    open (LET, ">$d6");
    # print "$d6\n";
    print LET "<title>T1 Glossary 2000</title>\n";
    print LET "<body background = \"gifs\\/$-backg4.gif\">\n";
    print LET "<body text = \"#000000\" link = \"#0000FF\" vlink = \"545454\">";
    print LET "<center>\n";
    print LET "<img src=\"gifs\\let_$file1.gif\" alt = \"Letter $file1\">";
    print LET "<a href=\"t1g-indx.html\">[A-Z]</a><br></center>\n";
    print LET "<base target = \"main1\">\n";
    print LET "<hr>\n";
    print LET "<center>\n";
  }
}

```

```

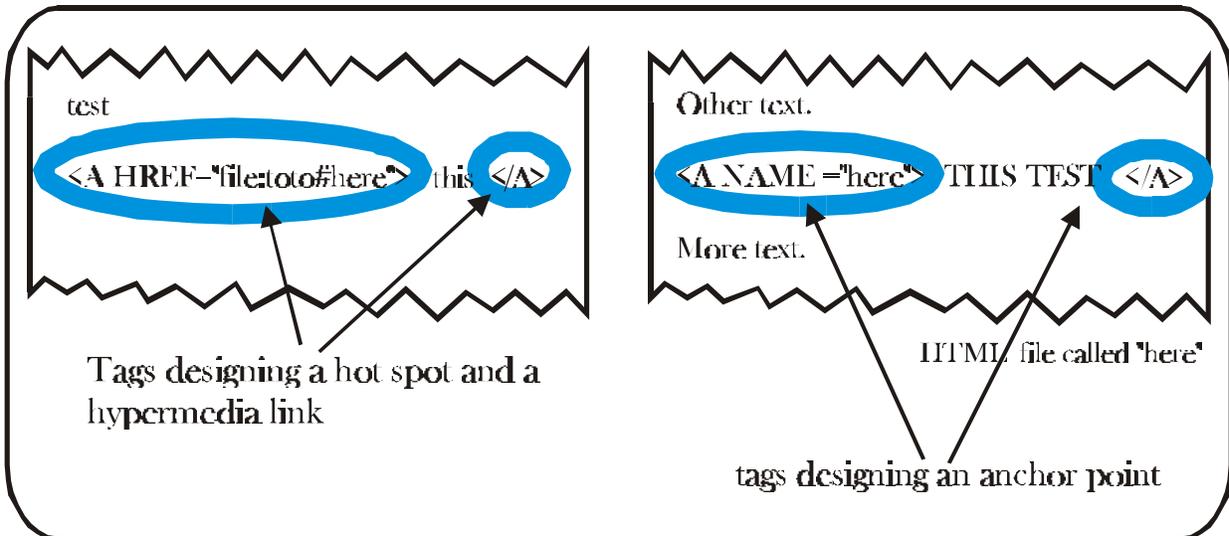
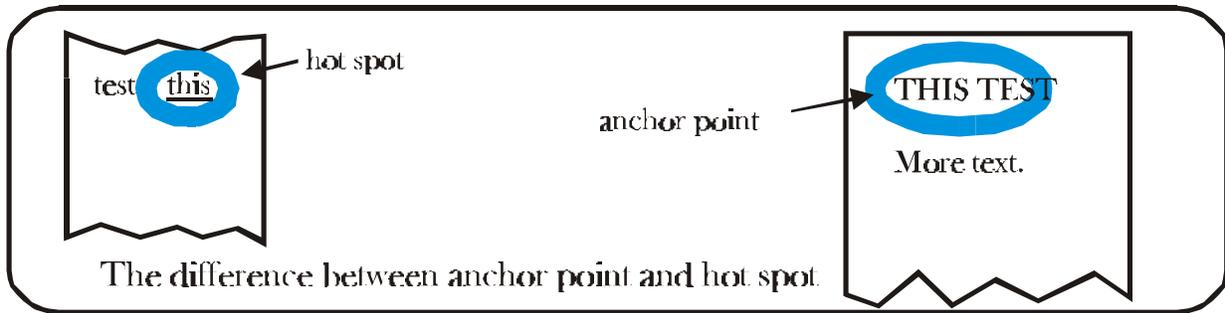
# ===== Special for A and Z =====
  if ($file1 !~ /a/) {
    $ _ = $file2;
    tr/a-z/A-Z/;
    print LET "<a href=\"all_$file2.html\" target=\"main2\">Back</a> to $ _, ";
  }

  if ($file1 !~ /z/) {
    $ _ = $file;
    tr/a-z/A-Z/;
    print LET "<a href=\"all_$file.html\" target=\"main2\">Forward</a> to $ _";
  }

  print LET "<font size=-3><hr>\n<ul>\n";
  print LET "<font color=black><li type=disc> = Term unchanged from previous
glossary</font>\n";
  print LET "<font color=red><li type=circle> = New term for this
glossary</font>\n";
  print LET "<font color=blue><li type=square> = New T1 term for this
glossary</font>\n";
  print LET "</ul><hr></font></center>\n";
  print LET "<ul>\n";
close (LET);
} # end if alphabetic
$file2 = $file1;
$file1 = $file;
} # ----- End "All" -----

```

The next section of code opens the anchors-list file and reads each term name into an internal array. Figure 15 illustrates the concepts of anchors and hot spots.



When a user clicks on the hot spot, the toto file is fetched and displayed, starting from the anchor point whose name is "here".

Figure 15. The relationship of anchors and hot spots.

The anchors list is an important aspect of the glossary, since it indicates to script "make5800" which term names are to be hyperlinked and which are not. Some term names should not be linked each time they occur. For example, the term *branch* is defined in the glossary as a part of a telephone network. Unfortunately, in almost every case in this glossary where that term is used, it is not used in that sense (it is usually used as a "branch of science"). Therefore, it would be inappropriate to link such a usage to the definition indicating that it is part of a telephone network. To prevent such

inappropriate linking from happening, the term *branch* has been manually removed from the anchors list. Nearly 100 words and phrases were removed from the anchors list for similar reasons.

Once the anchors list is imported into the array, the array is sorted according to the number of characters in each search phrase. This is done to prevent double links (i.e., links within other links) from being created. Later in the script, a module will search for occurrences of each phrase and build a hyperlink when it finds any. It has elements to determine if it is trying to build a link inside an already linked phrase. If so, it aborts the current attempt and moves on to the next term in the list. By starting the search with the longest phrases in the acronym list, the smaller phrases contained within the larger ones will automatically be ignored.

```
# ----- Import Term List -----
open (TERM, "./37d-list2.txt");
while ($t1 = <TERM>) {
  chop ($t1);
  last if ($t1 =~ /^$/);
  #$t1 =~ s/\(.*\)//g;
  #$t1 =~ s/\[.*\]/g;
  $t1 =~ s/^\s*//;
  $t1 =~ s/\s*$//;
  $t1 =~ s/\s{2,}/ /g;
  $t1 =~ tr/[A-Z]/[a-z]/;
  push (@wordlist, $t1);
}

@wordlist = reverse (sort {length ($a) <=> length ($b)} @wordlist);
close (TERM);

# foreach (@wordlist) {print "$_\n";} die;
```

This part of the script begins the actual processing of the 8090 definitions in the glossary. Two output files are opened. One will become a single, very large file (downloadable in two parts) that contains all of the definitions, properly formatted for viewing. (Most viewers will not want to regularly use this large hypertext file, since it can often take ~15 minutes to download. But, it can be used to print a 600-page document containing all of the terms and definitions in alphabetical order with their accompanying equations and figures and in formatted fonts, but without any hyperlinks.)

The second file created will be a simple list of all the term names in the glossary. This list can be used for later debugging of this and other scripts. This list can also be used as a quick reference to identify terms that may or may not be included in the glossary, or it can be used as a new anchors list (see Section 5.7.2) for future iterations of this script.

The final step in this part of the script is to set up a loop that reads in each line of code from the master database file (see Section 5.7.1). Each line is then parsed to follow a standard output format.

```
# ----- Open Main Database -----
open (ALLD, ">./allddefs.html");
print ALLD "<table width=90%><tr><td>\n";
```

```

open (NLIST, ">./37d-list4.txt");
print NLIST "-- List ---\n";
open (DEFS, ". /03-37d.txt");
OUT: while ($d1 = <DEFS>) {
  chop ($d1);
  if ($d1 !~ /\^s*$/ && $d1 !~ /\^s*<p>\s*$/) {
    $d1 = $d1 . " "; # adds space to compensate for SRT
  }
}

```

As each line of text is read from the master database, some pre-processing is required to ensure that the format of the text is uniform. For example, if a sentence ended with a bolded phrase, some editors would place the sentence-ending period inside the final “bold-off” code and others would place it outside. The following section of the script attempted to locate these inconsistencies and, in this example, to place the period uniformly in front of the “bold-off” code.

Approximately ten similar inconsistencies are addressed by this section of the script.

```

# ----- Fixes Coding Oddities -----

$d1 =~ s/\^s*//; # Removes leading spaces
$d1 =~ s/<\b>\. \. <\b>/gi; # Swithes when period after Bold off
$d1 =~ s/<\b>: \: <\b>/gi; # Swithes when colon after Bold off
$d1 =~ s/<\b>; \; <\b>/gi; # Swithes when semi after Bold off
$d1 =~ s/<\b>\, \, <\b>/gi; # Swithes when comma after Bold off
$d1 =~ s/\@\$<b>\@\$/; # Adds bold code
$d1 =~ s/\@\$<b>/<b>\@\$/; # Makes sure Bold is before @$
$d1 =~ s/<b>\s*<b>/<b>/g; # Removes one Bold if now are two
$d1 =~ s/<\b>\s*<b>/ /g; # Concats when Bold Off/Bold On
$d1 =~ s/<b><\i>/<i><b>/gi; # Swithes when Bold comes first

$d1 =~ s/\s{1,}/ /g; # Removes more than 1 spaces
$d1 =~ s/</ </g; # Adds space before and after <> so that
$d1 =~ s/>/> /g; # there will be boundaries around words.
$d1 =~ s/<b>\s*\@\$<b>\@\$/; # Removes space between Bold and @$

```

This next section will address the case where an automatic hyperlink will not be created by the script (due to its name having been removed from the anchors list (Section 5.7.2), but the editor wants to force a link to be created in this specific case. The reasons why a term name might not be in the anchors list are described earlier. In this situation, the editor should edit the master database and put the two characters, “yy,” at the start of any single-word term name that occurs within a definition (the string, “yy,” was selected for its unlikeliness to appear anywhere else in the glossary as a naturally occurring string).

When this script finds the characters, “yy,” at the start of a word, it will blindly build a hyperlink to a definition with that word as its name. Care needs to be taken to ensure that such a definition file exists. The script “chk-link” (Section 5.6.7) can help to determine when a hyperlink has been built to a non-existing file.

```

while ($d1 =~ /yy/) {
  #print "$d1\n";
  $d8 = $d1;
  $d8 =~ s/. *byy(\w*)\W */$1/;
  $d8 =~ tr/A-Z/a-z/;
}

```

```

    $d1 =~ s/(.*)yy($d8)(.*)/$1<a href = "_$d8.html">$2</a>$3/i;
    #print "$d8\n$d1\n";
    #if ($d8 eq "trunk") {die;}
}

```

This part of the script will try to determine the term name for the current definition being analyzed. First, it looks to see if the character string “@\$” is in the current line of the master database (the string “@\$” was chosen as an indicator because of its unlikeliness of being a naturally occurring string in the database).

If that character string is found in the current line, the script considers that that line is the first line of a new definition. The script processes this line to determine the term name and format it in a uniform manner (removing the colon, making it all lower case, removing punctuation, etc.).

Once the term name is determined, that name is written to disk as part of a new anchors list (Section 5.7.2).

```

# ----- Attempts to Isolate and Standardize Term Name -----
if ($d1 =~ /\@$/) {
    $d2 = $d1;          # Makes copy
    $d2 =~ s/:.*//;    # Removes all after colon

    $d2 =~ s/yy//g;    # Removes all "force" codes
    $d2 =~ s/([\^nN])xx/$1/g; # Removes all "supress" codes
    $d2 =~ s/\@$/g;    # Removes @$ codes
    $d2 =~ s/\s*<\i>\s*- /<\i>- /g; # Fixes problems with ital in mid term name
    $w8 = $d2;
    $w8 =~ s/<[bB]>//g;
    $w8 =~ s/<\/[bB]>//g;
    $d2 =~ tr/[A-Z]/[a-z]/; # Makes all lower case
    $d2 =~ s/\(.*\)//;
    $d2 =~ s/[. *\]//;
    $d2 =~ s/ <.>//g;
    $d2 =~ s/ <...>//g;
    $d2 =~ s/ <\/...>//g;
    $d2 =~ s/<img.*\ ">//g; # Replace specials with nothing (was "_")
    $d2 =~ s/<.*>//g;
    $d2 =~ s/\&.*;//g; # Replace specials with nothing (was "_")
    $d2 =~ s/\/_//g; # Replace "/" with "_"
    $d2 =~ s/^\s*//; # Remove leading spaces
    $d2 =~ s/\s*$//; # Remove lagging spaces
    $d2 =~ s/\s{2,}/ /g; # Remove double spaces

    print "$d2\n";
    if (length($d2) == 3) {print NLIST "$d2\n"};

} # End if @$
# next OUT; # Ends processing to just generate anchors list.

```

This next portion of the script determines the first letter of the term name. It is used to determine which of the alphabetical lists the term will appear in. Many of the term names in the glossary begin with numerals or symbols. The editors of the glossary decided that these term names should appear

in the alphabetical lists as if those numerals and symbols were spelled out. In other words, a definition named “6-coding” would appear after “sit-down strapping” and before “sizzle wire,” and the term name “S phasing” would appear after the term “old data” and before the term “1-sided encryption.”

When the script discovers a non-alphabetic character at the start of a term name, the spelled-out word is inserted instead. This insertion does not appear in the final presentation of the term name within the hypertext pages.

```
# ----- Finds First Letter of Term Name -----
$D6 = $D2;          # Whittle it down to first letter
$D6 =~ s/^\s*//;
$D6 =~ s/^(.).*/$1/;

    if ($D6 eq "_") {$D6 = "s"}
    if ($D6 eq "1") {$D6 = "o"}
    if ($D6 eq "0") {$D6 = "z"}
    if ($D6 eq "9") {$D6 = "n"}
    if ($D6 eq "\.") {
        $D6 = $D2;          # Whittle it down to second letter
        $D6 =~ s/^\s*//;
        $D6 =~ s/^\.(.).*/$1/;
    }
if ($D6 NE $D9) {
    open (NOW, ">>now-end.txt");
    print NOW "Letter -$D6- (was -$D9-)\n";
    print NOW " $D2\n";
    close (NOW);
    $D9 = $D6;
    system ("date >>now-end.txt");
}

# if ($D9 eq "b") {print ALLD "</td></td></table>\n"; close (ALLD); close (NLIST);
die};          # Kills process at letter B (for testing)
# if ($D6 eq "a" || $D6 eq "b" || $D6 eq "c" || $D6 eq "d" || $D6 eq "e" || $D6
eq "f" || $D6 eq "g") { next OUT; } # Skips a and b.
```

This portion of the script looks for unusual or incorrect formatting within the definition and attempts to reformat it. For example, if a space is found to be the first character within an italicized phrase, the space is moved outside the italicized characters.

```
# ----- Fixes Oddities in Term Definition -----
$D1 =~ s/ </>/g;
$D1 =~ s/> />/g;
$D1 =~ s/ {1,} / /g;
$D1 =~ s/> (\W)\>/$1/g; # Removes single space after bracket.
$D1 =~ s/ <\i> -/<\i>/g; # Fixes problems with ital in mid term name
$D1 =~ s/([bBiI])>\s*([\/])/>$1<$2/g;
$D1 =~ s/<b><\i>/<\i><b>/g;
$D1 =~ s/<i><\b>/<\b><i>/g;
$D1 =~ s/<i>(\s*)/$1<i>/g;
$D1 =~ s/(\s*)<\i>/<\i> /g;
$D1 =~ s/<b>(\s*)/$1<b>/g;

$D1 =~ s/\s*<sub>/<sub>/g;
$D1 =~ s/<sub>\s*/<sub>/g;
$D1 =~ s/\s*<\sub>/<\sub>/g;
# $D1 =~ s/<\sub>\s*/<\sub>/g;
```

```

    $d1 =~ s/\s*<sup>/<sup>/g;
    $d1 =~ s/<sup>\s*/<sup>/g;
    $d1 =~ s/\s*</sup>/</sup>/g;
#   $d1 =~ s/</sup>\s*/</sup>/g;

    $d1 =~ s/(\s*)</b>/</b> /g;
    $d1 =~ s/<b><b>/<b>/g;
    $d1 =~ s/(<\/[bBiI]>)</$1 </g;
    $d1 =~ s/\></> \</g;      # Adds space between adjacent brackets.
    $d1 =~ s/\s*</i>\s*-/</i>-/g; # Fixes problems with ital in mid term name
    $d1 =~ s/^\s*//;
    $d1 =~ s/\( \(/g;          # Removes space after opening bracket
    $d1 =~ s/ \)/g;          # Removes space before closing bracket
    $d1 =~ s/\&quot;\;/g;      # Puts quotes in

# while ($d1 =~ /\[^\<\/] /) {
#   $d3 = $d1;
#   $d3 =~ s/.\*\[([^\<\/])/$1/;
#   $d3 =~ s/([^\>])\].*/$1/;
#   $d1 =~ s/$d3/<a href="t1g-intr\.html#refs\">$d3</a>/;
#   print "$d1\n$d3\n";
# }

    $d8 = $d1;

} # End if blank line or just <p>

# ----- Replaces Special Exemptions -----

    $d1 =~ s/(n<\i>\s*-ary code) ([^: ])/<a href="_n_ary_code\.html\">$1</a>$2/;
        $d1     =~ s/(bandwidth\s*#\&149;\s*distance product) ([^: ])/<a
href="_bandwidthdistance_product\.html\">$1</a>$2/;
        $d1     =~ s/(bandwidth\s*#\&149;\s*length product) ([^: ])/<a
href="_bandwidthlength_product\.html\">$1</a>$2/;

```

This next portion of the script is probably the most critical. It creates all the hyperlinks within the 8090 glossary definitions. It takes the potential list of anchors (7000+ term names) and searches each line of the 8090 definitions for each name in the anchors list. If a match is found, a hyperlink is created to the found definition.

This is the part of the script that causes the whole process to take more than half a day to run. It must check an average of 50 words in each of 8000 definitions for the occurrence of any of 7000 term names. That is approximately 3 billion comparisons that need to take place.

```

# ----- Builds Links in Definitions -----

print "$d1\n";
foreach $t2 (@wordlist) {
# last; # Stops linking if not remmed out (faster compile)

    $t3 = $t1 = $t2;
    $t1 =~ s/\(\/\//g;
    $t1 =~ s/\/\//g;
    $t1 =~ s/\/\//g;
    $t1 =~ s/\/\//g;
    $t1 =~ s/\/\//g;
    if ($t1 =~ /^0/) {
        $t1 =~ s/^0//;
    }
}

```

```

    St3 = St1;
    St1 =~ s/0/-/g;
    St3 =~ s/0/_/g;
}

St3 =~ s/\(.*\)//g;
St3 =~ s/[.*\]//g;
St3 =~ s/\s*$//;          # Removes trailing spaces
St3 =~ s/\s+/_/g;        # Replaces spaces with "_"
St3 =~ s/'/_/g;          # Replaces " ' " with "_"
St3 =~ s/\/_/g;          # Replaces " / " with "_"

if ($d2 ne $t1) {

if ($d1 =~ /\@\$/) {
    $ _ = $d1;                # splits word and considers only
    /: /;                    # phrases after the ":"
    $w1 = "$`" . "$&";
    $w2 = $';                # splits word and counts "<" and ">"
} else {
    $w2 = $d1;
    $w1 = "";
}

# Below is section that will not find second occurrence of phrase
# in a line:

$ _ = $w2;                    # splits word and counts "</a" and "<a"
$y = /\b$t1\b/i;              # on each side to see if word nested
$ _ = $w3 = $`;              # in another link.

# if ($t1 =~ /WRU/i && $d1 =~ /WRU/i) {
# $d1 =~ s@(\b)($t1)(\b)@$1 <a href="_$t3.html">$2</a> $3@i;
# print "$d1\n$t1 -- $t3\n";
# if ($d1 =~ /$t1/i) { print "yes\n"; }
# die;
# }

if ($y > 0) {                  # Substitutes if match found.

$count1 = s/<a/<a/g + 1;
    $ _ = $w3;
    $count2 = s/<\a/<\a/g + 1;
# print "$d1\n$t1 -- $count1 -- $count2\n";
if ($count1 eq $count2) {

    $w2 =~ s@(\b)($t1)(\b)@$1 <a href="_$t3.html">$2</a> $3@i;

    if ($w2 =~ /internet\.html\`\>Internet\`</) { # Special for "Internet"
        $w2 =~ s/_internet\>I/_internet-x>I/;
        print "$w2\n";
        die;
    }

    $d1 = "$w1" . "$w2";

} # End count matches
} # End contains phrase

} # End word not same as target
} # End wordlist

```

The following lines are special programming to compensate for the use of reserved characters within the text of definitions. For example, the character “[” is used by Perl to indicate grouped variables or constants. Normally, when Perl finds this character in a text string, it interprets it to mean that a group of variables or constants is beginning. Unfortunately, this character is used within the glossary in its ordinary, English usage. The special programming in this section integrates the two different kinds of usage for that character and others. It also addresses some spacing issues around those special characters.

```
# if ($d1 =~ /WRU/) {
#   print "$d1\nSt1\nSt3\n";
#   die;
# }

$d1 =~ s/ / /g; # Removes double space.

$d1 =~ s/([\nN])xx/$1/g; # Removes all "supress" codes
$d1 =~ s/^\> \-^\>\- /g; # Removes single space before - .
$d1 =~ s/\- \<^\- \</g; # Removes single space before - .
$d1 =~ s/\( (\W)^\($1/g; # Removes single space after bracket.
$d1 =~ s/\[ (\W)^\[$1/g; # Removes single space after bracket.
$d1 =~ s/\{ (\W)^\{$1/g; # Removes single space after bracket.
$d1 =~ s/(\W) \)/$1\)/g; # Removes single space after bracket.
$d1 =~ s/(\W) \]/$1\]/g; # Removes single space after bracket.
$d1 =~ s/(\W) \}/$1\}/g; # Removes single space after bracket.
$d1 =~ s/^\>\s*\.\ ^\>\. /g; # Removes space after bracket and before ..
$d1 =~ s/^\> \:^\>\: /g; # Removes single space after bracket and before ..
$d1 =~ s/^\> \;^\>\; /g; # Removes single space after bracket and before ..
$d1 =~ s/^\> \,^\>\, /g; # Removes single space after bracket and before ..

$d1 =~ s/ / /g; # Removes double space.
```

The following portion addresses graphical two-dimensional images, video clips, and audio clips. In the raw database file, a referenced image is represented by a series of characters similar to “#ifilename#”. The “#i” indicates to the script that the characters immediately following it, up to the trailing “#” character, are the file name of the image (in this case, it is “filename”). The script creates the appropriate coding to display a hyperlink to the referenced image, with a small icon informing the user of the link. When viewing a definition, the reader must click on the icon to view the image (this may be a slight inconvenience to the reader, but it significantly decreases the download time for those readers with relatively slow Internet connections).

Similarly, video clips are represented by the string, “#vfilename#” and audio clips are represented as “#afilename#”.

In-line images are represented by the string, “#ffilename.” These images are displayed, directly to the user, as the definition is shown on the screen and are only used, in this glossary, for displaying equations.

```
# ----- Replaces Special Images -----
$d1 =~ s/#f(..)/<p><img src=\"equats\"/t1g_$.gif\" alt=\"Equation $1\"><p>/g;
$d8 =~ s/#f(..)/<p><img src=\"equats\"/t1g_$.gif\" alt=\"Equation $1\"><p>/g;
# Replaces #fxx with an Equation
```

```

    Sd1 =~ s/#a(.*)#<p><a href="images\/$1.wav"><\a>/g;
    Sd8 =~ s/#a(.*)#<p><a href="images\/$1.wav"><\a>/g;
    # Replaces #axx# with an Audio Image

    Sd1 =~ s/#v(.*)#<p><a href="images\/$1.avi"><\a>/g;
    Sd8 =~ s/#v(.*)#<p><a href="images\/$1.avi"><\a>/g;
    # Replaces #vxx# with an Video Image

    Sd1 =~ s/#i(.*)#<p><a href="images\/$1.gif"><\a>/g;
    Sd8 =~ s/#i(.*)#<p>/g;
    # Replaces #ixx# with an Image

```

Once each definition has had its hyperlinks created, the definition is written out as its own Web page. The name of its Web page is related to its name, usually by removing spaces and unprintable characters and replacing each with an underscore character (“_”).

Some basic header information is prepended to the Web page to format the information in a consistent manner.

```

# ----- Printing Definition to File -----
# ===== See if Line is a Continuation =====
if ($d1 !~ /\@\$/) { # Check to see if line has "@$" in it. If not,
    $d4 = ">" . $d3; # add to previous file.
    open (OUT, $d4);
    print ALLD "$d8\n";
    print OUT "$d1\n";
    close (OUT);
} else { # Make a new file if not a continuation

    Sd1 =~ s/\@\$//g;
    Sd3 = "_" . "$d2" . ".html";

    Sd3 =~ s/\*/star/g;
    Sd3 =~ s/\/\_/g;
    Sd3 =~ s/\s{2,}/ /g;
    Sd3 =~ s/\s*\.\html/\.\html /;
    Sd3 =~ s/\s{1,}/_/g;
    Sd3 =~ s/'/_/g;

    while (-e $d3) { # Make a new file name for dup names
        Sd3 =~ s/(.*) (\.\html)\/$1-x$2/;
    }

    Sd3 = ">" . "$d3";

# ----- Print Definition -----
open (OUT, $d3);
print ALLD "</b></i></a><p>\n";
print OUT "<title>Definition: $d2</title>\n\n";
print OUT "<body background = \"gifs\/\$-backg.gif\">";
print OUT "<body text = \"#000000\" link = \"#0000FF\" vlink = \"545454\">";
print OUT "<h1><center>$w8</center></h1><hr>\n";
print OUT $d1;
print ALLD "$d8\n";

```

```

close (OUT);

$d6 = ">>all_" . "$d6" . ".html";
$d7 = $d3;
$d7 =~ s/^>//;
$d7 =~ s/\s{2,}/ /g;

open (LET, $d6);
$sr5 = 0;
foreach (@new1) {
    $sw8 =~ s/^\s*//;
    if ($_ eq $sw8) {
        $sr5 = 1;
    }
}

$sr6 = 0;
foreach (@new2) {
    $sw8 =~ s/^\s*//;
    if ($_ eq $sw8) {
        $sr6 = 1;
    }
}

if ($sr5 == 1) {
    print LET "<font color=red><li type=\"circle\">";
}
if ($sr6 == 1) {
    print LET "<font color=blue><li type=\"square\">";
}
if ( ($sr6 != 1) && ($sr5 != 1) ) {
    print LET "<font color=black><li type=\"disc\">";
}

print LET "</font>";
print LET "<a href=\"\$d7\">$sw8</a>\n";
close (LET);

} # End of make new term file
} # End of list of defs

close (NLIST);
print ALLD "</td></tr></table>\n";
close (ALLD);

```

After all the processing is complete, every Web page that was created has trailing information added to complement the starting information previously added to the top of the file. For example, where lists were created within a file, the HTML code for ending the list was appended to the end.

```

# ----- Add Trailing HTML Code to Term Lists -----
foreach $r2 (@letters) {
    foreach $file (<_ $r2\.html>) {
        &addtrail;
    }
} # end r2

foreach $r1 (@letters) {
    foreach $r2 (@letters) {
        foreach $file (<_ $r1$r2*.html>) {
            &addtrail;
        }
    } # end r2
} # end r1

```

```

}          # End r1

open (DONE, ">>now-end.txt");
print DONE "\n*** M5800 Done ***\n";
close (DONE);

# -----
sub addtrail {
open (TEMP, ">>$file");
print TEMP "$tf";
print TEMP "$dt";
print TEMP "\n<FORMMETHOD=\"POST\" ACTION=\"http://www.its.blrdoc.gov/cgi-bin/add-
comm prl\">";
print TEMP "<INPUT TYPE=\"hidden\" NAME=\"filename\" VALUE=\"$file\">";
print TEMP "<INPUT TYPE=\"submit\" VALUE=\"Send Comments on this Def\">";
close (TEMP);
system ("mv $file ..")
}          # End subroutine

foreach $file (<*html>) {
open (TEMP, ">>$file");
print TEMP "$tf";
close (TEMP);
system ("mv $file ..")          # End file
}          # End DT

```

5.6.7 Script “chk-link”

This script checks all of the hyperlinks within all 8000+ Web pages created earlier to see which links lead to existing Web pages and which are “broken.”

The first thing the script does is to list all of the Web pages that exist in the glossary subdirectory. This list is read into an array in the script. Then, one at a time, each of those Web pages is opened up and read into a temporary variable. That variable is examined for the existence of a hyperlink. If one is found, the array is checked for the existence of that filename. If one is found in the array, the link is reported as

“successful” and written to a log file on disk. If not, that fact is also written to the log file.

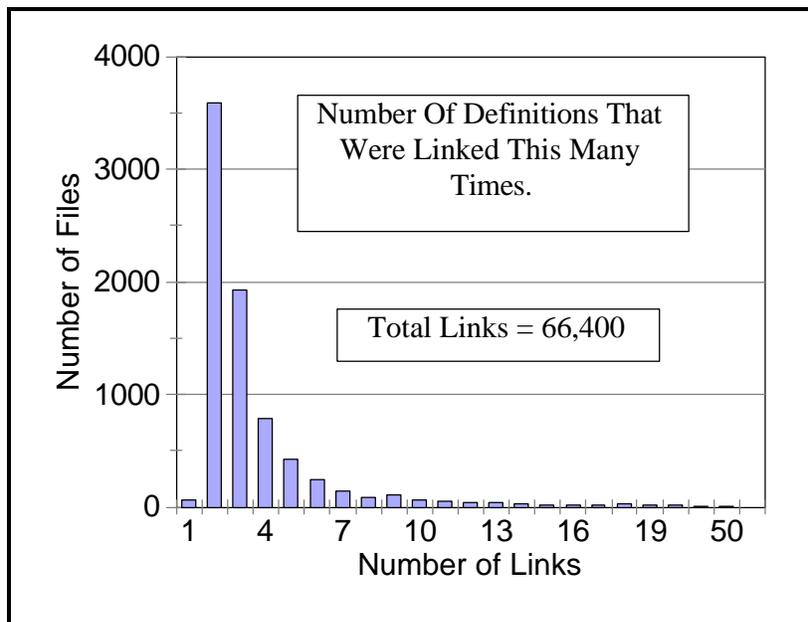


Figure 16. The number of times that some definitions were linked.

For record keeping purposes, each time a successful link is found, a counter is incremented for that named link. The chart in this section shows that certain words were linked far more than others. Term names like “system” and “signal” were linked more than 600 times within the document, while some, like “digital loopback” and “dehop” only had one link to them (in the index). The accompanying Figure 16 (above) and Table 4 (below) show the top ten most linked terms.

Table 4. The Top 10 Most Linked Terms

Term Name	Number of Links to It
signal	698
data	660
system	646
information	641
network	495
transmission	489
time	441
infosec	430
communications	413
user	363

Below is the coding for the script that does this counting.

```
#!/usr/local/bin/perl
  $usedir = "/nd/bing/public_html/fs-1037d";
  chdir ("$usedir") || die "Cannot change to that directory";
  $nlc = 0;
  $nlb = 0;
  unlink (". /logfile.txt");

  # system ("cp ./temp-gen/*.html .");
  system ("ls -pR >list.txt");

  $cdir = "";
  open (FILES, ". /list.txt");
  open (FF, ">temp-gen/log-link.txt");
  while ($sname = <FILES>) {
```

```

chop ($sname);
if ($sname !~ /\$/ && $sname =~ /\S/) {
if ($sname =~ /^\. (.*)\:/) {
    $cdir = $sname;
    $cdir =~ s/^\.\.//;
    $cdir =~ s/^\:/;
    $cdir = "$cdir" . "\/";
} else {
    $sname = "$cdir" . "$sname";
    push (@nrm, "$sname");
}
}
}
close (FILES);

foreach $sname (@nrm) {
if ($sname =~ /\.htm/ || $sname =~ /\.html/) {
print FF "\n$sname\n";
++$htmfiles;
open (THIS, $sname);
while ($line = <THIS>) {
    chop ($line);
    while ($line =~ /href=/) {
        ++$linkschecked;
        $tmp = $line;
        $tmp =~ s/.*<a href=\"//;
        $tmp =~ s/\. *//;
        # $tmp =~ s/\///;
        $tmp =~ s/\#.*//;

        while ($tmp =~ /\"/) {
            $tmp =~ s/\".*//;
        }
        print FF " >> $tmp\n";
        if (-e "$tmp") {} else {
            ++$badlinks;
            print FF " **** Bad Link ***\n";
        }
        $line =~ s/(.*)href=\"/$1/;
    }
}
close (THIS);
}
}
print FF "\n$htmfiles HTML Files Checked";
print FF "\n$linkschecked Links Examined";
print FF "\n$badlinks Bad Links Found\n";

close (FF);

```

5.6.8 Script “make-all”

This script compiles all definitions into one or two very large alphabetical files that can be viewed by clicking the “Files” hot spot in the bottom Frame (Frame #4) on the glossary Web page.

This script works much the same as the “make5800” script described earlier, except that the

definitions that would have been written to separate, small Web pages, are all written to one or two very large Web pages. All images and figures are in-line, rather than being presented as icons that the user needs to click on. No hyperlinking is done within the file.

This service is provided for users who might want to print the entire 600+ pages of the glossary to their own printer or who might want to view the entire document at once.

```
#!/usr/local/bin/perl
  $usedir = "/nd/bing/public_html/fs-1037d";
  chdir ("$usedir/temp-gen") || die "Cannot change to that directory";

@l e t t e r s =
("_", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z", "0", ".");

open (TERM, ". /37d-list.txt");
while ($t1 = <TERM>) {
  chop ($t1);
  last if ($t1 =~ /^$/);
  $t1 =~ s/\(. *\)//g;
  $t1 =~ s/[. *i\]//g;
  $t1 =~ s/^\s*//;
  $t1 =~ s/\s*$//;
  $t1 =~ s/{2,}/ /g;
  $t1 =~ tr/[A-Z]/[a-z]/;
  push (@wordlist, $t1);
}

@wordlist = reverse (sort {length ($a) <=> length ($b)} @wordlist);
close (TERM);

# foreach (@wordlist) {print "$_ \n";} die;

open (ALLD, ">$usedir/alldefs.html");
print "<table width=90%><r><td>\n";

open (DEFS, ". /01-37d.txt");
while ($d1 = <DEFS>) {
  chop ($d1);
  if ($d1 !~ /\s*$/ && $d1 !~ /\s*\<p>\s*$/) {
    $d1 = $d1 . " "; # adds space to compensate for SRT

    $d1 =~ s/#f(..)/<p><center></center><p>/g;
    # Replaces #fxx with an Equation
    $d1 =~ s/#i(.*)#//g;
    # Replaces #ixx# with an Image

    $d1 =~ s/yy//g;
    $d1 =~ s/^\s*//; # Removes leading spaces
    $d1 =~ s/<\b>\: \: <\b>/; # Moves bold off to after colon
    $d1 =~ s/@\$/<b>@\$/; # Adds bold code
    $d1 =~ s/@\$/<b>/<b>@\$/; # Makes sure Bold is before @$
    $d1 =~ s/<b>\s*<b>/<b>/g; # Removes one Bold if now are two
    $d1 =~ s/<\b>\s*<b>/ /g; # Concats when Bold Off/Bold On

    $d1 =~ s/\s{1,}/ /g; # Removes more than 1 spaces
    $d1 =~ s/</ </g; # Adds space before and after <> so that
    $d1 =~ s/>/> /g; # there will be boundaries around words.
    $d1 =~ s/<b>\s*\$/<b>@\$/; # Removes space between Bold and @$
  }
}
```

```

if ($d1 =~ /\@\$/) {
  $d2 = $d1;          # Makes copy
  $d2 =~ s/:.*//;    # Removes all after colon

  $d2 =~ s/yy//g;    # Removes all "force" codes
  $d2 =~ s/([\^nN])xx/$1/g; # Removes all "supress" codes
  $d2 =~ s/\@\$//g;  # Removes @$ codes
  $d2 =~ s/\<s*\<i>\s*-/ <i>-/g; # Fixes problems with ital in mid term name
  $w8 = $d2;
  $w8 =~ s/<[bB]>//g;
  $w8 =~ s/<\/[bB]>//g;
  $d2 =~ tr/[A-Z]/[a-z]/; # Makes all lower case
  $d2 =~ s/\(. *\)\//;
  $d2 =~ s/\[. *\]\//;
  $d2 =~ s/ <.>//g;
  $d2 =~ s/ <...>//g;
  $d2 =~ s/ <\/...>//g;
  $d2 =~ s/ <\/.>//g;
  $d2 =~ s/<img.*\>//g; # Replace specials with nothing (was "_")
  $d2 =~ s/<.*>//g;
  $d2 =~ s/\&.*//g;    # Replace specials with nothing (was "_")
  $d2 =~ s/\/\//g;    # Replace "/" with "_"
  $d2 =~ s/\^s*//;    # Remove leading spaces
  $d2 =~ s/\s*$//;    # Remove lagging spaces
  $d2 =~ s/\s{2,}/ /g; # Remove double spaces
} # End if @$

$d6 = $d2;          # Whittle it down to first letter
$d6 =~ s/^\s*//;
$d6 =~ s/^(.)*$/1/;

  if ($d6 eq "_") {$d6 = "s"}
  if ($d6 eq "1") {$d6 = "o"}
  if ($d6 eq "0") {$d6 = "z"}

foreach $t2 (@wordlist) {
  last;          # Stops word linking (for faster tests)
  $t3 = $t1 = $t2;
  if ($t1 =~ /^0/) {
    $t1 =~ s/^0//;
    $t3 = $t1;
    $t1 =~ s/0/-/g;
    $t3 =~ s/0/_/g;
  }

  $t3 =~ s/\s+/_/g;    # Replaces spaces with "_"
  $t3 =~ s/\'/_/g;    # Replaces "' " with "_"
  $t3 =~ s/\/_/g;    # Replaces " / " with "_"

if ($d2 ne $t1) {

if ($d1 =~ /\@\$/) {
  $_ = $d1;          # splits word and considers only
  /:/;              # phrases after the ":"
  $w1 = "$`" . "$&";
  $w2 = $';          # splits word and counts "<" and ">"
} else {
  $w2 = $d1;
  $w1 = "";
}

$_ = $w2;          # splits word and counts "</a" and "<a"
$y = /\b$t1\b/i;   # on each side to see if word nested

```

```

    $ _ = $w3 = $`;          # in another link.

if ($y > 0) {                # Substitutes if match found.

    $count1 = s/<a/<a/g + 1;
    $ _ = $w3;
    $count2 = s/<\/a/<\/a/g + 1;

if ($count1 eq $count2) {

    $w2 =~ s@(\b)($t1)(\b)@$1 <a href="_$t3.html">$2</a> $3@i;
    $d1 = "$w1" . "$w2";

} # End count matches
} # End contains phrase

} # End word not same as target
} # End wordlist

$d1 =~ s/ </>/g;
$d1 =~ s/> />/g;
$d1 =~ s/ {1,} / /g;
$d1 =~ s/ <\/i> -<\/i>/g; # Fixes problems with ital in mid term name
$d1 =~ s/([bBiI])>\s*<([\/])/$1><$2/g;
$d1 =~ s/<b><\/i><\/i><b>/g;
$d1 =~ s/<i><\/b><\/b><i>/g;
$d1 =~ s/<i>(\s*)/$1<i>/g;
$d1 =~ s/(\s*)<\/i><\/i> /g;
$d1 =~ s/<b>(\s*)/$1<b>/g;

$d1 =~ s/\s*<sub>/<sub>/g;
$d1 =~ s/<sub>\s*/<sub>/g;
$d1 =~ s/\s*<\/sub>/<\/sub>/g;
# $d1 =~ s/<\/sub>\s*/<\/sub>/g;
$d1 =~ s/\s*<sup>/<sup>/g;
$d1 =~ s/<sup>\s*/<sup>/g;
$d1 =~ s/\s*<\/sup>/<\/sup>/g;
# $d1 =~ s/<\/sup>\s*/<\/sup>/g;

$d1 =~ s/(\s*)<\/b><\/b> /g;
$d1 =~ s/<b><b>/<b>/g;
$d1 =~ s/(<\/[bBiI]>)/$1 </g;
$d1 =~ s/\s*<i>\s*-<\/i>/g; # Fixes problems with ital in mid term name
$d1 =~ s/^\s*//;
$d1 =~ s/([\^nN])xx/$1/g; # Removes all "supress" codes

} # End if blank line or just <p>

print "$d2\n";
if ($d1 !~ /\@$/ ) { # Check to see if line has "@$" in it. If not,
    print ALLD "$d1\n";
}
else {                # Make a new file

print ALLD "</b><i></a><p>\n";
$d1 =~ s/\@$/ /g;
$d3 = "_" . "$d2" . ".html";

$d3 =~ s/\*/star/g;
$d3 =~ s/\/\_/g;
$d3 =~ s/\s{2,} / /g;
$d3 =~ s/\s*\.\html/\.\html /;
$d3 =~ s/\s{1,} /_/g;
$d3 =~ s/' /_/g;

```


almost all formatting codes stripped away and has reference codes added at the start of each term name.

The search engine opens this index file and begins looking for words that match the specified search parameters. When it finds one, it records the reference code for the definition within which it found the match. The list of matching reference codes is used by the search engine to show the user a list of term names matching the search request.

```
#!/usr/local/bin/perl

# -- Setting base variables
$base = "http://www.its.bldrdoc.gov/projects/t1glossary2000";
$ibase = "/nd/bing/public_html/pub";
$dir = "/nd/bing/public_html/fs-1037d";
chdir("$dir") || die "Cannot change to that directory";

open (FSDB, ">$ibase/t1gd.txt");
#print FSDB "@@\n";

# -- Cycle through each directory on the hard drive in the selected folder
@letters = ("_", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z", ".", "0", "1", "9");

foreach $r1 (@letters) {
  foreach $r2 (@letters) {

    # -- And each HTML file
    EACHFILE: foreach $d2name (<_{$r1}{$r2}*.html>) {
      open (TERM, "$d2name");
      print "$d2name\n";

      # -- For each file, set defaults
      $isadef = 0; $linex = "*** error ***"; $i = 0;
      # -- and then examine each line
      while ($line = <TERM>) {
        ++$i;
        #print "$i";
        chop ($line);
        #print "$line\n";
        if ($i == 1 && $line !~ /title>Def/) {next EACHFILE;}
        if ($line =~ /^<body/) {next;}
        if ($line =~ /<font/) {next EACHFILE;}

        $line =~ s/\i>.([]\+\:\:\/)/\i>$1/i;
        $liney = $line; # Removes bracketted things
        while ($liney =~ /</ && $liney =~ />/) {
          $liney =~ s/(.*)<.*>(.*)/$1$2/;
        }
        # print "$liney\n";
      }

      # -- Determine the hit definition's Name and HTML
      if ($line =~ /^<b>.*\:. *</b>/) {
        $linex = $line;
        while ($linex =~ /\: /) {
          $linex =~ s/^(^.*)\:.*$/1/;
        } # End While
        $linex =~ s/<b>//g;
        $linex =~ s/src="\.\.\/src="\$ibase/g;
        $isadef = 1;
      }
    }
  }
}
```

```

        print FSDB "@@\\href=\"\$base/\$d2name\" target=\"main1\">$line</a>\n";
    } # end if Def

    # -- Push hits onto stack
    if ($isadef == 1) {
        print FSDB "$line\n";
    }

} # end while

} # End foreach subdirectory
}
}

close (FSDB);

```

5.6.11 Other Miscellaneous Scripts

Other miscellaneous scripts were used in developing the glossary database and Web pages. Some of these scripts were designed primarily for housekeeping, for backup, or for integrity protection, and some for addressing internal Web-design decisions. These miscellaneous scripts are not detailed in this report.

5.6.12 Script “copyall”

This script copies the remaining files from the internal server to the external Web server.

```

#!/usr/local/bin/perl
$usedir = "/nd/bing/public_html/fs-1037d";
chdir ("$usedir") || die "Cannot change to that directory";

# $i = 0;
# open (DONE, "temp-gen/now-end.txt");
# while ($n = <DONE>) {
#   print "$n\n";
#   if ($n =~ /M5800/) {++$i;}
# }
# close (DONE);

if ($i >= 1) {
#system ("cp ind* /green2/htdocs/projects/telecomglossary2000");
#system ("cp t1* /green2/htdocs/projects/telecomglossary2000");
#system ("cp all_* /green2/htdocs/projects/telecomglossary2000");
#system ("cp *html /green2/htdocs/projects/telecomglossary2000");
#system ("cp .ht* /green2/htdocs/projects/telecomglossary2000");
system ("cp 37d-list.txt /green2/htdocs/projects/telecomglossary2000");
#system ("cp -R dir-??? /green2/htdocs/projects/telecomglossary2000");
system ("cp -R equats /green2/htdocs/projects/telecomglossary2000");
system ("cp -R gifs /green2/htdocs/projects/telecomglossary2000");
system ("cp -R images /green2/htdocs/projects/telecomglossary2000");
system ("cp -R other /green2/htdocs/projects/telecomglossary2000");
system ("cp -R symbols /green2/htdocs/projects/telecomglossary2000");
open (DONE, ">>temp-gen/now-end.txt");
print DONE "\n*** COPY Done ***\n";
close (DONE);
#} else {

```

```

    system ("mail bing < /nd/bing/public_html/fs-1037d/temp-gen/now-end.txt");
    #print "Si\n"; die;
#}

```

5.6.13 Script “countlnk”

This script counts the number of links found within the created Web pages and compiles them into a simple report. This report can be examined to find pages that are not linked anywhere, or pages that, for some reason, are linked too many times.

```

#!/usr/local/bin/perl
$usedir = "/nd/bing/public_html/fs-1037d";
chdir ("$usedir") || die "Cannot change to that directory";
unlink (". /temp-gen/countfil.txt");

# system ("cp ./temp-gen/*.html .");
system ("ls -pR >list.txt");

open (FF, ">./temp-gen/countfil.txt");
print "-- Out File Opened --\n";

$ج = 0;
open (FILES, ". /list.txt");
while ($سname = <FILES>) {
    chop ($سname);
    if ($سname =~ /\.html/) {
        ++$ج; print FF "- $سname\n";
        open (HTML, ". /$سname");
        while ($سline = <HTML>) {
            chop ($سline);
            $سi=0;
            while ($سline =~ /\\"_/ && $سline =~ /\.html/) {
                ++$سi; if ($سi eq 100) {last;}
                $سg = $سline;
            # print "$سg\n";
                $سg =~ s/(\\"S*.html).*/$س1/;
            # print "$سg\n";
                $سg =~ s/.*\\"//;

                ++$سrdz{$سg};
                $سline =~ s/$سg//;
            # print "$سline\n";

                print FF "+ $سrdz{$سg}   $سg\n";
            } #end while
        } #end while
        close (HTML);
    # print "--- close HTML ---\n\n";
    } #end if
} #end while
close (FILES);
print FF "--- close LIST ---\n";

foreach $سkey (sort { $سrdz{$سa} <=> $سrdz{$سb} } keys %سrdz) {
    print "$سrdz{$سkey}   $سkey\n";
    ++$سcntz{$سrdz{$سkey}};
    print FF "$سrdz{$سkey}   $سkey\n";
}

```

```

print FF "--- Overall counts ---\n";
foreach $key (sort {$a <=> $b} keys %cntz) {
  print FF "$key hits occurred $cntz{$key} times.\n";
}
close (FF);

```



5.7 Files

5.7.1 The Database File

This file is the heart of the glossary. All 8000+ definitions are contained in this database file. The listing in this chapter only shows approximately 10 of the terms (culled from the letters A and Z). A complete listing would take more than 600 pages. All 8000+ definitions can be accessed in two long, alphabetically arranged files by clicking on the “Files” icon (reproduced at right). Clicking on this icon also presented yet another hyperlink allowing the user to send e-mail to the ITS editors.

Each definition can be identified, by the editor or a script, by the existence of the characters “@\$” at the beginning. The end of a definition can be identified by the start of a new definition (i.e., a “@\$” symbol appears) at the end of the file.

Within each definition, HTML codes can be seen. Any HTML codes that exist within this database will pass through the compilation scripts unchanged.

```

      <b>@$a: </b> <i>Abbreviation for</i> <b>atto</b> (10<sup>-18</sup>).
<i>See</i> <b>metric system </b>
      <b>@$abandoned call: </b> A call in which the call originator disconnects
or cancels the call after a connection has been made, but before the call is
established. <p>
      <b>@$abandonment: </b> Network replacement of a connect signal with an on-
hook signal (network) prior to receiving a CI (customer installation) answer signal;
abandonment is the only way to end an unanswered call attempt. [T1.405-1989] [T1
Gloss '90]
      <b>@$abbreviated address: </b> An address that has fewer
characters than the full address, usually for special communications and
other services or for certain users. <i>Note: </i> Examples of abbreviated
addresses are (a) a four-digit telephone number for a user calling another
user connected to the same switching exchange, and (b) message addresses
that have only the addressee name and station code or number. [After Weik '96]<p>
....[Letters A to Z]....

      <b>@$zip-cord: </b> In optical communications, a two-fiber cable consisting
essentially of two single-fiber cables having their jackets conjoined by a strip of
jacket material. <i>Note 1: </i> This name is borrowed from electrical terminology
referring to lamp cord. As with lamp cord, optical zip-cord may be easily furcated
by slitting or tearing the two jackets apart, permitting the installation of optical
connectors. <i>Note 2: </i> Zip-cord cables include both loose-buffer and tight-buffer
designs. [FAA]#i zip4c#
      <b>@$zone: </b><i> See <b></i>communications zone, Fresnel zone, skip
zone. <p>
      <b>@$zone of influence: </b> <b></b> The area surrounding a power station
that is raised in potential above a remote or true grounding (earthing) point.

```

[T1. 318- 2000]

zone of protection: The space adjacent to a lightning protection system that is substantially immune to direct lightning strokes. [T1. 318-2000]

zone of silence: *Synonym skip zone.*

Z Time: *Abbreviation for Zulu xxTime.*
See Coordinated Universal Time.

Zulu Time (Z): *Synonym Coordinated Universal Time.*
Formerly a synonym for Greenwich Mean Time.

5.7.2 The Anchors List

Below is a small sample of the anchors list. The full list contains more than 7000 term names. The list is used to generate hyperlinks within other definitions. Only words that appear in this list are linked. If, for example, there was a defined word that the editors did not want linked (like the word “A”, for example), that word was removed from this list (as described in Section 5.6.6), even though the word “a” is defined (“a” is defined as an abbreviation for “atto,” a metric measure denoting 10⁻¹⁸).

```
abandoned call
abandonment
abbreviated address
abbreviated address calling
abbreviated dialing
abort
abrasive
absolute address
absolute delay
...
zone
zone of influence
zone of protection
zone of silence
zooming
z time
zulu time
```

5.7.3 The Log File

This log file recorded the length of time that most of the scripts required to complete their respective functions. This file proved useful for discovering where the bottlenecks or errors occurred during processing.

```
Wed Nov  8 17:19:59 MST 2000
Letter -a- (was --)
a
Wed Nov  8 17:20:16 MST 2000
Letter -b- (was -a-)
b
Wed Nov  8 18:49:30 MST 2000
Letter -c- (was -b-)
cable
Wed Nov  8 19:52:23 MST 2000
Letter -d- (was -c-)
```

d*
Wed Nov 8 22:01:53 MST 2000
Letter -e- (was -d-)
e-
Wed Nov 8 22:53:20 MST 2000
Letter -f- (was -e-)
facet erosion
Wed Nov 8 23:26:20 MST 2000
Letter -g- (was -f-)
gain
Wed Nov 8 23:58:16 MST 2000
Letter -h- (was -g-)
hack
Thu Nov 9 00:10:27 MST 2000
Letter -i- (was -h-)
iana
Thu Nov 9 00:22:41 MST 2000
Letter -j- (was -i-)
jabber
Thu Nov 9 01:08:37 MST 2000
Letter -k- (was -j-)
k
Thu Nov 9 01:11:40 MST 2000
Letter -l- (was -k-)
label
Thu Nov 9 01:17:25 MST 2000
Letter -m- (was -l-)
mac
Thu Nov 9 01:39:48 MST 2000
Letter -n- (was -m-)
na
Thu Nov 9 02:30:22 MST 2000
Letter -o- (was -n-)
object
Thu Nov 9 02:54:28 MST 2000
Letter -p- (was -o-)
pabx
Thu Nov 9 03:16:10 MST 2000
Letter -q- (was -p-)
qa
Thu Nov 9 04:03:19 MST 2000
Letter -r- (was -q-)
raceway
Thu Nov 9 04:07:29 MST 2000
Letter -s- (was -r-)
s
Thu Nov 9 04:45:27 MST 2000
Letter -t- (was -s-)
t
Thu Nov 9 05:51:54 MST 2000
Letter -u- (was -t-)
udp
Thu Nov 9 06:30:54 MST 2000
Letter -v- (was -u-)
vacant code
Thu Nov 9 06:40:38 MST 2000
Letter -w- (was -v-)
wafer
Thu Nov 9 06:50:50 MST 2000
Letter -x- (was -w-)
x-bits
Thu Nov 9 07:00:37 MST 2000
Letter -y- (was -x-)
yagi antenna

Thu Nov 9 07:01:37 MST 2000
Letter -z- (was -y-)
z
Thu Nov 9 07:01:58 MST 2000

. . .

6. EXECUTION

The programs and scripts used to collect e-mail definitions and to generate, organize, and groom the >8000 Web pages for *Telecom Glossary 2000* are detailed and explained in Section 5 above to provide prospective users with specific information both for re-creating the glossary and to illustrate how the specific scripts and use of the e-mail reflector promoted due process in “electronic meetings.”

Execution of the scripts provided significant efficiencies in the glossary-development processes. Although computer code for creating all of the 69,000 hyperlinks could have been inserted manually, that procedure would have been too slow to be useful. Use of the Perl scripts allowed these 69,000 links to be generated and checked in a single 15-hour time segment.

The programs and scripts for the glossary were executed (usually) every night on an ITS mainframe computer. Initially, the programs and scripts required more than 20 hours to complete all tasks. This schedule was not useful, since changes made to the glossary were not viewable for the greater part of every day. The ITS programmer trimmed tasks and moved files to reduce the daily run time for revision and regeneration to approximately 16 hours. Execution for the later versions of the programs began each afternoon at 4 p.m., MDT, and was completed the next morning by approximately 8:00 a.m. In the final-draft stages following the November 2000 ballot, the regeneration programs were executed only on those evenings when the ITS editors had introduced a change into the body of the glossary.

7. RESULTS AND CONCLUSIONS

This report addresses the “electronic-meeting” techniques (and the computer programs generated to effect those techniques) which were used to develop ANS T1.523-2001, *Telecom Glossary 2000*. The report also identifies, in Section 3, the advantages of speed, economy, and accuracy which those techniques fostered both for the developers and for the glossary’s user community at large. The report also describes the contents of the glossary itself—its fields of emphasis, its enhancements, and its user community.

The final product of the electronic meetings is a Web-based, hyperlinked, search-engine equipped telecommunications glossary with approximately 8090 Web pages, a hyperlinked appendix of abbreviations, 94 drawings, 4 video clips, 2 audio clips, and 69,000 (mostly internal) hyperlinks. This glossary used FED-STD-1037C, *Glossary of Telecommunication Terms*, as its baseline, and—after removing any obsolete terms—added approximately 2300 new terms from the literature and from ANSI T1 Standards and T1 Reports. The new terms addressed the fields of (a) the Internet/Web, (b) security and infrastructure protection, and (c) photonics. Definitions for all of these terms were reviewed by an Ad Hoc Group of experts in the T1 community. The work was completed in just under 12 months, principally using e-mail discussions and “electronic meetings” to reach consensus on the definitions.

Use of Perl scripts contributed to the very rapid speed with which the very large glossary database was updated, groomed, formatted, revised, and linked. Use of electronic meetings for updating the glossary had the advantages of speed, efficiency/economy, wider participation of experts, wider accessibility, and accuracy (as discussed in Section 3). The economies and speed of the revision using electronic means have signaled the option for future revisions (or supplements) at more frequent intervals than the mandatory 5-year cycle. These more frequent revisions can keep the glossary more current in the dynamic world of telecommunications and can enhance its usefulness to telecommunications users, vendors, and service providers in the world marketplace, where the standard is key to interoperability and key to understanding the global market.

8. REFERENCES

- [1] L. Garcia, “A new role for government in standard setting?” *StandardView*, vol. 1, no. 1, pp. 4-7, 1993.
- [2] W. Ingram and E. Gray, “A search engine for Federal Standard 1037C,” NTIA Report 99-365, March 1999.
- [3] W. Ingram and E. Gray, “A Federal Standard on electronic media,” NTIA Report 98-350, Aug. 1998.
- [4] E. Gray and W. Ingram, “A hypertext glossary: More than print with bells and whistles,” in *Proceedings of the 1998 STC Region 7 Conference, The Medium Formerly Known as Print*, Denver, CO, Oct. 1998.
- [5] R. L. Schwartz, *Learning Perl*, Sebastopol, CA: O’Reilly & Associates, Inc., 1993.
- [6] L. Wall and R. L. Schwartz, *Programming Perl*, Sebastopol, CA: O’Reilly & Associates, Inc., 1991.

APPENDIX A: HISTORY OF THE GLOSSARY

A.1. Historical Background

Use of standardized vocabulary for telecommunications terminology assists in ensuring reliability and interoperability of telecommunications products and services. The benefits to the users of strong, accepted standards include (a) the control of product quality, which promotes trade, (b) the interoperability of equipment and services, which promotes communication and productivity, and (c) the support of interdependent national economies, which reduces the cost of doing business by amplifying the benefits of economies of scale while reducing the need for retooling and retrofitting [1].

To secure these benefits for the U.S. telecommunications community in the world market and to assist in satisfying the Federal Government's mission of enhancing reliability and ensuring interoperability of the Government's resources, the Institute for Telecommunication Sciences (ITS)—under the sponsorship of the National Communications System (NCS)—has led the development of a telecommunications glossary and its revisions for the last 25 years.

A1.1 History of the Glossary

This report describes the e-mail and computerized processes used for the recent updating of FED-STD-1037C (1996) to *Telecom Glossary 2000*, which was adopted as an American National Standard in February 2001. The previous editions of this popular glossary span 25 years.

From 1974 to 1996, ITS has held a leadership role in developing and updating the 1996 Federal Standard 1037 (FED-STD-1037C), *Glossary of Telecommunication Terms*, and its predecessor documents. (See Figure 1.) The 1037C Glossary is available on the Web at

<http://glossary.its.blrdoc.gov/fs-1037/>

The popular FED-STD 1037C was updated using exclusively electronic means: the ITS editors created interactive Web pages and e-mail communications paths to unite the geographically dispersed committee of experts and to collect their definitions, reviews, and technical comments in an efficient, timely manner. The result of this effort is a new American National Standard, a glossary of telecommunication terms called *Telecom Glossary 2000*. The new American National Standard is slated to be available free of charge to the public on the ATIS (Alliance for Telecommunications Industry Solutions)/T1 Web site as well as on the ITS Web site

<http://www.its.blrdoc.gov/projects/telecomglossary2000> .

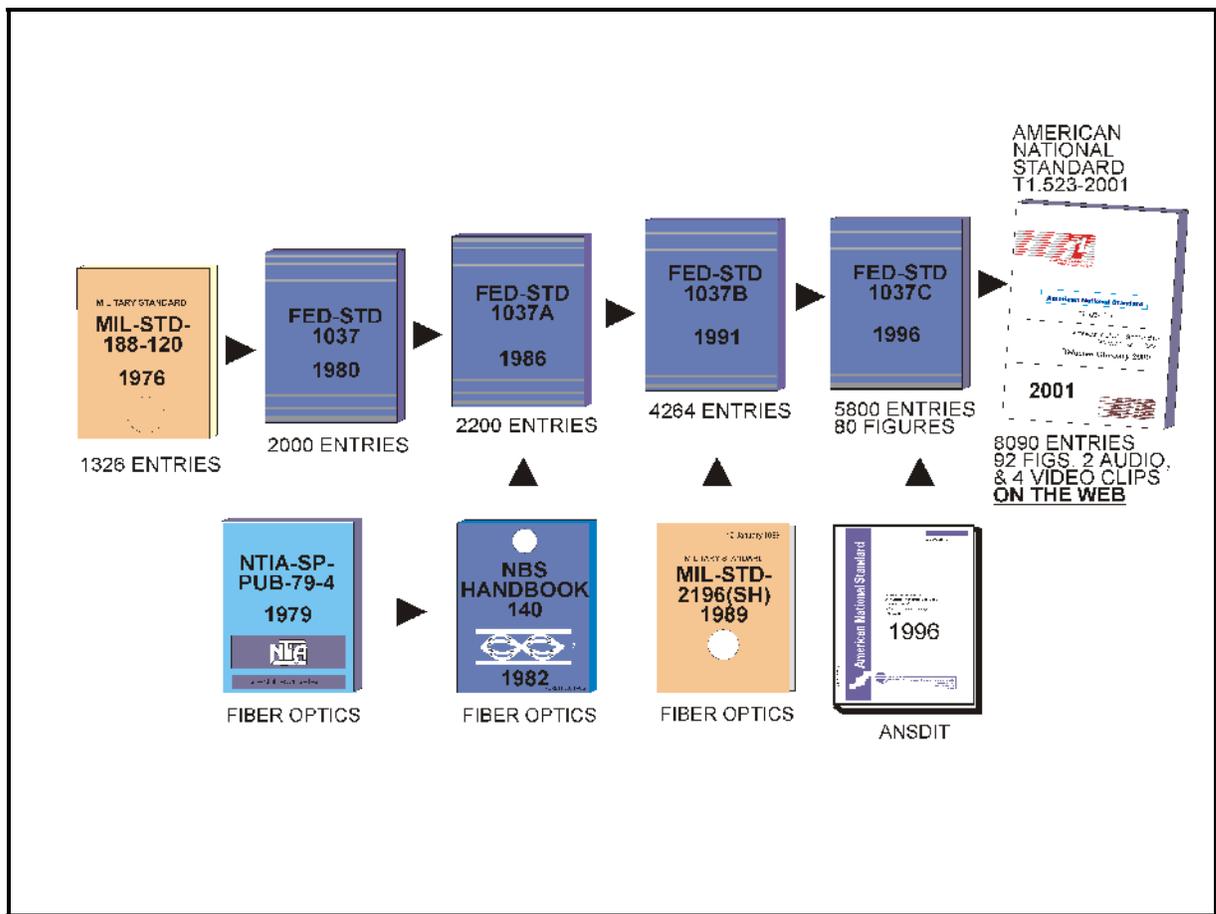


Figure A-1. History of the development of the ANS T1.523-2001 glossary.

With foresight, the glossary’s sponsors recognized the need to update the 1996 edition to advance the interoperability, reliability, and operational effectiveness of telecommunications equipment and services by standardizing telecommunications terminology used in technical specifications. They particularly wished to accommodate new Web/Internet terminology, and to address security and infrastructure-protection terms. The sponsors and editors, as members of the Federal Government, remain committed to the use of voluntary industry standards whenever such standards exist or can be developed. Further, the developers worked to retain the glossary’s wide acceptance, credibility, and usefulness, by preserving its status as a standard (see Sections 1.2 and 1.3 above), rather than changing it to a Federal Telecommunications Recommendation.

The ITS editors spoke with representatives of three different standards-development organizations (SDO’s), to select one for release and promulgation of the new standard. The editors selected the T1 standards community for co-hosting the work of revising the glossary.

The advantages of selecting this group were that ITS has some leadership responsibilities in the T1A1 Technical Subcommittee (TSC) on Performance and Signal Processing and, therefore, can provide an energizing force and a useful level of control in guiding the revision of the document. The contributions of Committee T1 participants expanded the revised standard's content and consensus base, and its publication as a T1-developed American National Standard maximized its benefit to the U.S. telecommunications industry.

APPENDIX B: REFERENCES FOR *TELECOM GLOSSARY 2000*

B.1 Source References for the Glossary

The terms and accompanying definitions contained in *Telecom Glossary 2000* are drawn from authoritative sources such as Federal Standard 1037C, *Glossary of Telecommunication Terms*, 1996; T1 Committee Standards and T1 Technical Subcommittee Standards and Reports; documents from the International Telecommunication Union; documents from the International Organization for Standardization, and documents from the American National Standards Institute, and numerous authoritative dictionaries and lexicons. Numerous U.S. Government publications (*e.g.*, the *Code of Federal Regulations, Title 47*) were also drawn upon for this and earlier versions of the standard.

The editors of this standard have rewritten many definitions as deemed necessary either to reflect technology advances or to make definitions, which were phrased in specialized terminology, more understandable to a broader audience.

In those cases where definitions were taken verbatim from copyrighted sources, the editors received *written* permission from the copyright holder to use the definition. The definitions from 47 CFR, from the *NTIA Manual*, from the *Radio Regulations*, and from other treaty documents were used verbatim.

Definitions with no source citation following them are either (a) from FED-STD-1037C, or (b) from a body of definitions proposed and reviewed by industry standards workers in the course of developing the draft *Telecom Glossary 2000*.

The full list of Normative References that is now part of the Foreword for *Telecom Glossary 2000* was called the *Legend* in earlier editions of the glossary. The complete list of Normative References is given below in this Appendix, in Tables B-1 through B-4.

Table B.1. Key to Literature Source Citations for *Telecom Glossary 2000* References

Citation	Meaning/Full Title
[] (no source citation)	These definitions are from two major sources: (a) FED-STD-1037C 1996, <i>Glossary of Telecommunication Terms</i> , and (b) a body of definitions proposed and reviewed by industry standards workers.
After...	Definitions with a source citation “[After...]” (as in “[After Weik '89]”) are the responsibility of the Ad Hoc Group editing the glossary rather than the responsibility of the authors of the cited material.
2196	Definitions extracted verbatim from MIL-STD-2196 (SH), <i>Glossary, Fiber Optics</i> (Jan. 1989). The citation “[After 2196]” indicates editing of the source definition. All such editing is the responsibility of the editors of this T1 standard and not the cited document.
2382-35	Committee Draft 12/15/98 of ISO/IEC 2382-35 on networking.
2382-8	ISO/IEC 2382-8, <i>Information Technology - Vocabulary: Control, integrity, and security</i> , 1998.
ANSDIT	<i>American National Standard Dictionary for Information Technology</i> (ANSDIT), ANSI X3.197-1996, and the draft of the Millennial Edition of ANSDIT 2000.
Bahorsky	Bahorsky, R. (ed.) (1998), <i>Official Internet Dictionary</i> , Government Institutes. Used with written permission of the holder of the copyright.
CC	Common Criteria for Information Technology (IT), Security Evaluation, August 1999, CCIMB-909-031. (Inspiration from this source resulted in definitions that were rewritten to fit the <i>Telecom Glossary 2000</i> format and purpose.)
CCITT/CCIR or ITU-T/ITU-R	Recommendations and other documents from the ITU-T and ITU-R, formerly the Consultative Committee for Telegraph and Telephone, and the Consultative Committee for Radio.
47CFR	Definitions extracted verbatim from the <i>Code of Federal Regulations, Sec. 47, Telecommunications</i> (rev. Oct. 1, 1987, Oct. 1988 & ff.).
47 CFR 36-A	Definitions extracted verbatim from the <i>Code of Federal Regulations, Ch. 1, Sec. 47, Part 36, Appendix-Glossary</i> , Oct. 1, 1999 edition.
FAA	Beaty, James S., <i>Glossary of Optical Communication Terms</i> , April 1991, DOT/FAA/CT-TN91-9, U.S. Department of Transportation, Federal Aviation Administration, Technical Center, Atlantic City

Citation	Meaning/Full Title
	International Airport, NJ.
FRMR	<i>Federal Information Resources Management Regulation 201-13-103.</i>
FCC	<i>FCC Glossary of Telecommunications Terms</i> , on the Web at http://www.fcc.gov/cib/handbook.html [accessed August 2000].
FED-STD-1045A	<i>Telecommunications: HF Radio Automatic Link Establishment</i> (1993).
FP	Definitions extracted from Federal Information Processing Standard Publication (FIPS PUB) 11-3, <i>American National Standard Dictionary for Information Systems</i> , which adopts ANSI X3N-1990 (with unchanged title), and which is superseded by ANS X3.172, 1996, <i>American National Standard Dictionary of Information Technology</i> (ANSDIT).
IEC	IEC (1999), <i>TC Newsletter</i> , No. 6, February. [IEC, 3 rue de Varembé, P.O. Box 131, 1211 Geneva 20, Switzerland.] (Modified by the Ad Hoc Group.)
IEEE-100-1992	<i>The New IEEE Standard Dictionary of Electrical and Electronics Terms</i> (1992), 5th Edition, IEEE, Inc., 345 East 47th Street, New York, NY 10017-2394.
INFOSEC-99 or NIS	<i>National Information Systems Security (INFOSEC) Glossary</i> , NSTISSI No. 4009, January 1999 (Revision 1).
ITU-T or ITU-R	Recommendations and other documents from the ITU-T or ITU-R, formerly the Consultative Committee for Telegraph and Telephone, and the Consultative Committee for Radio.
JP1 or JP 1-02	Telecommunications definitions extracted from Joint Chiefs of Staff Publication No. 1-02, <i>Department of Defense Dictionary of Military and Associated Terms</i> (1999), available on the Web at http://huachuca-dcd.army.mil/nsto/lexicon/lexicon.htm [Most recent access: Oct. 2, 2000.]
JSB	Beaty, James S., U.S. Department of Transportation, Federal Aviation Administration, William J. Hughes Technical Center, Atlantic City International Airport, NJ 08405. E-mail communication to the Ad Hoc Glossary Group.
Mattila	Mattila, S. (1998), <i>Secure Electronic Transaction Terms</i> , e-mail to the Ad Hoc Glossary Group. [Used with written permission of the holder of the copyright.]
MS	<i>Microsoft Press Computer Dictionary</i> , 3rd ed. 1997 [Redmond, WA: Microsoft Press. ISBN 1-57231-743-4, 704 pp.] [Substantially modified by the Ad Hoc Group.]
NECA/FCC-5	National Exchange Carrier Association, Inc., Tariff No. 5 filed with the FCC on March 9, 2000.

Citation	Meaning/Full Title
NTIA	Definitions extracted verbatim from the <i>NTIA Manual of Regulations and Procedures for Federal Radio Frequency Management</i> .
NIS	The pre-1999 edition of INFOSEC-99. (See reference for INFOSEC-99 above.)
Pacific Newsbytes	<i>Pacific Newsbytes</i> Vol. 3, No. 10, Dec. 1998, & Vol. 4, No. 1, Jan. 1999.
RR	Definitions extracted verbatim from the <i>International Radio Regulations</i> , Malaga-Torremolinos (Oct. 1984, ref. 1985).
Silicon	Silicon Graphics (1998), <i>Film & Video Glossary</i> , 39 pp. [Inspiration from this glossary generated definitions that were significantly modified by the Ad Hoc Group to conform to the purpose of <i>Telecom Glossary 2000</i> .]
SMPTE	Society of Motion Picture and Television Engineers, a 60-page glossary, Proposed SMPTE Engineering Guideline, Annotated Glossary of Essential Terms for Electronic Production (1998). [Inspiration from this glossary generated definitions that were significantly modified by the Ad Hoc Group to suit the purposes of <i>Telecom Glossary 2000</i> .]
Sullivan	Sullivan, T.F.P. (ed.), <i>Official Telecommunications Dictionary</i> (1997), Government Institutes.
T1	Draft of <i>ANSI T1A1.5 Glossary</i> , Document #T1A1.5/94-137, July 29, 1994. S. Wolf (ed.). (For a complete bibliography of T1 Standards and Reports used in this glossary, see the ANSI T1 References List below.) NOTE: The T1-cited terms defined herein are as used in the standards developed by Committee T1. The same terms may be used in a different context with a different meaning.
After Weik '89	Use of this source citation acknowledges that the cited information is from <i>Communications Standard Dictionary</i> , 2nd ed., Dr. M. Weik, 1989 (Van Nostrand Reinhold Co., New York, NY), with the written permission of the holders of the copyright. These definitions are sometimes verbatim, but in most cases have been abbreviated or edited.
Weik	Weik, M. H., <i>Communications Standard Dictionary</i> , 1997, Chapman & Hall. Used with the written permission of the holder of the copyright.
X. ...	Definitions cited "[X. ...]" or "[After X. ...]" were obtained from the X9 Accredited Standards Committee. The definitions have been rewritten for the broader audience. The X9 documents used in this glossary are listed below.

Note 1: Appreciation is proffered to the Telecommunications Industry Association for permission to include definitions from ANSI/EIA/TIA-440A, *Fiber Optics Terminology* (1988). These entries are not source-cited herein because of extensive editorial rewriting by the editors of this document in the interest of making the specialized terminology more understandable to a broad audience. Note 2: Appreciation is extended to working group T1A1.5 for providing their draft glossary of terms and definitions relating to video-quality degradation. Note 3: Appreciation is extended to working group T1X1 for providing their glossary of terms and definitions relating to synchronization and timing.

ANSI T1 References List

Table B-2. Numerical List of ANSI T1 Standards Cited in the Glossary

Standard No., Date	Title of the ANSI T1 Standard (“R” = reaffirmed)
T1A1.2-2000 – T1A1/2	<i>Proposed Definitions for Baseline Document, T1A1.2/2000-002</i>
T1.101-1999	<i>Synchronization Interface Standard</i>
T1.104-1991 (R1997)	<i>Exchange-Interexchange Carrier Interfaces—Individual Channel Signaling Protocols</i>
T1.105.03-1994	<i>Synchronous Optical Network (SONET)—Jitter at Network Interfaces</i>
T1.105.09-1996	<i>Synchronized Optical Network (SONET) Network Element Timing and Synchronization</i>
T1.110-1999	<i>Signalling System No. 7 (SS7)—General Information</i>
T1.118-1992 (R1997)	<i>Signalling System No. 7 (SS7)—Intermediate Signalling Network Identification (ISNI)</i>
T1.204-1992 (R1997)	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Lower-Layer Protocols for Interfaces between Operations Systems and Network Elements</i>
T1.206-1994	<i>Digital Exchanges and PBXs—Digital Circuit Loopback Test Line with NxDSO Capability</i>
T1.207-2000	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Terminating Test Line Access and Capabilities</i>
T1.209-1998	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Network Tones and Announcements</i>
T1.210-1989 and 1993 (R1999)	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Principles of Functions, Architectures, and Protocols for Interfaces between Operations Systems and Network Elements</i>
T1.213-1990	<i>Coded Identification of Equipment Entities of the North American</i>

Standard No., Date	Title of the ANSI T1 Standard (“R” = reaffirmed)
(R1996)	<i>Telecommunications System for Information Exchange</i>
T1.215-1990 and 1994	<i>Operations, Administration, Maintenance and Provisioning (OAM&P)—Fault Management Messages for Interfaces between Operations Systems and Network Elements</i>
T1.216-1990 and 1998	<i>Signalling System Number 7 (SS7)—Operations, Maintenance and Administration Part (OAM&P)</i>
T1.217-1991 (R1998)	<i>Integrated Services Digital Network (ISDN) Management—Primary Rate Physical Layer</i>
T1.218-1991 (R1999)	<i>Integrated Services Digital Network (ISDN) Management—Data-Link and Network Layers</i>
T1.222-1991 (replaced by T1.251-1996)	<i>Identification of Exchange Carrier Company Names and Codes for the North American Telecommunications System</i>
T1.223.1991 (R1997)	<i>Information Interchange-Structure and Representation of Network Channel (NC) and Network Channel Interface (NCI) Codes for the North American Telecommunications System</i>
T1.224-1992 (withdrawn)	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Protocols for Interfaces between Operations Systems in Different Jurisdictions</i>
T1.226-1992	<i>Operations, Administration, Maintenance and Provisioning (OAM&P)—Management of Functions for Signalling System No. 7 (SS7) Network Interconnections</i>
T1.230-1994 (R1999)	<i>Telecommunications Credit Card and Billed Number Screening Validation Message Component</i>
T1.231-1997	<i>Digital Hierarchy—Layer 1 In-Service Digital Transmission Performance Monitoring</i>
T1.232-1996	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—G Interface Specification for Use with the Telecommunications Management Network (TMN)</i>
T1.233-1993 (R1999)	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Security Framework for Telecommunications Management Network (TMN) Interfaces</i>
T1.234-1993	<i>Signalling System Number 7 (SS7)—MTP Levels 2 and 3 Compatibility Testing</i>
T1.235-1993	<i>Signalling System Number 7 (SS7)—SCCP Class 0 Compatibility Testing</i>
T1.236-1993	<i>Signalling System Number 7 (SS7)—ISDN User Part Compatibility Testing</i>
T1.238-1994 and 1997	<i>Information Interchange—Structure for the Identification of Telecommunications Facilities for the North American Telecommunications System</i>

Standard No., Date	Title of the ANSI T1 Standard (“R” = reaffirmed)
T1.243-1995 (R1999)	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Baseline Security Requirements for the Telecommunications Management Network</i>
T1.244-1995	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Interface Standards for Personal Communications Services</i>
T1.245-1997	<i>Directory Service for Telecommunications Management Network (TMN) and Synchronous Optical Network (SONET)</i>
T1.246-1998 and 1999	<i>Operations, Administration, Maintenance and Provisioning (OAM&P)—Information Model and Services for Interfaces between Operations Systems Across Jurisdictional Boundaries to Support Configuration Management Customer Account Record Exchange (CARE)</i>
T1.250-1996	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Extension to Generic Network Information Model for Interfaces Between Operations Systems and Network Elements to Support Configuration Management-Analog and Narrowband ISDN Customer Service Provisioning</i>
T1.251-1996	<i>Identification of Telecommunications Service Provider Codes for the North American Telecommunications System</i>
T1.253-1999	<i>Information Interchange—Code Description and Codes for the Identification of Location Entities for the North American Telecommunications System</i>
T1.255-1997	<i>In-Service, Non-Intrusive Measurement Device (INMD)—Methodology for Applying INMD Measurements to Customer Opinion Models</i>
T1.256-1999	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Model for Interface Across Jurisdictional Boundaries to Support Electronic Access Service Ordering: Inquiry Function</i>
T1.263-1998	<i>Operations, Administration, Maintenance, and Provisioning (OAM&P)—Model for Interfaces Across Jurisdictional Boundaries to Support Service-Level Connection Management</i>
T1.266-2000	<i>Information Interchange—Structure for the Identification of Telecommunications Circuits for the North American Telecommunications System</i>
T1.305-1990 (withdrawn)	<i>Digital Processing of Program Audio Signals-Algorithm for 15-kHz Audio at 384 kbit/s using 14/11 Bit Coding</i>
T1.308-1996	<i>Central Office Equipment—Electrostatic Discharge Immunity Requirements</i>
T1.309-1990 (withdrawn)	<i>Digital Circuit Multiplication Equipment (DCME)—Interface Functional and Performance Specification</i>

Standard No., Date	Title of the ANSI T1 Standard (“R” = reaffirmed)
T1.310-1991 (withdrawn)	<i>Digital Processing of Voice-Band Signals—Algorithms for 5-, 4-, 3-, and 2-bit/Sample Embedded Adaptive Differential Pulse Code Modulation (ADPCM)</i>
T1.312-1991 (R1997)	<i>Voice Packetization—Packetized Voice Protocol</i>
T1.313-1997	<i>Electrical Protection for Telecommunications Central Offices and Similar Type Facilities</i>
T1.316-1997	<i>Electrical Protection of Telecommunications Outside Plant</i>
T1.317-1993 (R1999)	<i>Uniform Language for Accessing Power Plants—Human-Machine Language</i>
T1.318-1994, 2000	<i>Electrical Protection Applied to Telecommunications Network Plant at Entrances to Customer Structures or Buildings</i>
T1.320-1994	<i>Above-Baseline Electrical Protection for Designated Telecommunications Central Offices and Similar-Type Facilities against High-Altitude Electromagnetic Pulse (HEMP)</i>
T1.321-1995	<i>Electrical Protection for Network Operator-Type Equipment Positions</i>
T1.328-1995	<i>Protection of Telecommunications Links from Physical Stress and Radiation Effects and Associated Requirements for DC Power Systems (A Baseline Standard)</i>
T1.329-1995	<i>Network Equipment—Earthquake Resistance</i>
T1.330-1997	<i>Valve-Regulated Lead-Acid Batteries Used in the Telecommunications Environment</i>
T1.401.03-1998	<i>Network-to-Customer Installation Interfaces—Analog Voicegrade Switched Access Lines with Calling Number Delivery, Calling Name Delivery or Visual Message-Waiting Indicator Features</i>
T1.407-1997	<i>Network to Customer Installation Interfaces—Analog Voicegrade Special Access Lines Using Customer-Installation-Provided Loop-Start Supervision</i>
T1.408-1990 (replaced by T1.403.01- 1999)	<i>Integrated Services Digital Network (ISDN) Primary Rate—Customer Installation Metallic Interfaces Layer 1 Specification</i>
T1.409-1996	<i>Network to Customer Installation Interfaces—Analog Voicegrade Special Access Lines Using E&M Signaling</i>
T1.414-1998	<i>Network to Customer Interfaces—Enhanced 911 Analog Voicegrade PSAP Access Using Loop Reverse-Battery Signaling</i>
T1.503-1996	<i>Network Performance Parameters for Dedicated Digital Services—Definitions and Measurements</i>
T1.507-1996	<i>Network Performance Parameters for Circuit-Switched Digital Services—Definitions and Measurements</i>

Standard No., Date	Title of the ANSI T1 Standard (“R” = reaffirmed)
T1.508-1998	<i>Loss Plan for Evolving Digital Networks</i>
T1.509-1995 (R1999)	<i>Packetized Circuit Multiplication Equipment—Interface Specification</i>
T1.514-1995	<i>Network Performance Parameters and Objectives for Dedicated Digital Services—SONET Bit Rates</i>
T1.601-1988 (R1999)	<i>Integrated Services Digital Network (ISDN)—Basic Access Interface for Use on Metallic Loops for Application on the Network Side of the NT (Layer 1 Specification)</i>
T1.615-1992 (R1999)	<i>Digital Subscriber Signalling System No. 1 (DSS1)—Layer 3 Overview</i>
T1.620-1991 (R1997)	<i>Integrated Services Digital Network (ISDN)—Circuit-Mode Bearer Service Category Description</i>
T1.625-1993 (R1999)	<i>Integrated Services Digital Network (ISDN)—Calling Line Identification Presentation and Restriction Supplementary Services</i>
T1.627-1993 (R1999)	<i>Broadband ISDN—ATM Layer Functionality and Specification</i>
T1.628-1993	<i>Routing, Bridging, and Transfer of Emergency Service Calls (RBTEESC)</i>
T1.629-1999	<i>Broadband ISDN---ATM Adaptation Layer 3/4 Common Part—Functions and Specification</i>
T1.630-1999	<i>Broadband ISDN—ATM Adaptation Layer for Constant Bit Rate Services Functionality and Specification</i>
T1.640-1996	<i>Broadband ISDN Network Node Interfaces and Inter-Network Interfaces—Rates and Formats Specifications</i>
T1.646-1995 (supercedes T1.624-1993)	<i>Broadband ISDN—Physical Layer Specification for User-Network Interfaces Including DS1/ATM</i>
T1.647-1995	<i>Integrated Services Digital Network (ISDN)—Conference Calling Supplementary Service</i>
T1.649-1995	<i>B-ISDN Cell Relay Service Description</i>
T1.653-1996	<i>Integrated Services Digital Network (ISDN)—Call Park Supplementary Service</i>
T1.661-1997	<i>Signalling System Number 7 (SS7)—Release to Pivot (RTP)</i>
T1.667-1999	<i>Intelligent Network</i>
T1.701-1994 (R1999)	<i>Universal Personal Telecommunication (UPT)—Service Description (Service Set One)</i>
T1.702-1995	<i>Personal Communications Terminology</i>
T1.707-1998	<i>Requirements for a User Identity Module (UIM) for Personal Communications Services (PCS)</i>
T1.708-1998	<i>PCS 1900 Service Provider Number Portability</i>
T1.711-1999	<i>Number Portability for PCS 1900 Short Message Service and Other</i>

Standard No., Date	Title of the ANSI T1 Standard (“R” = reaffirmed)
	<i>Services</i>
T1.800.01-1995	<i>Visual Telephone Systems and Terminal Equipment Using Digital Channels up to 1920 kbit/s</i>
T1.800.06-1995	<i>Multipoint Control Units for Audiovisual Systems Using Digital Channels up to 1920 kbit/s</i>
T1.800.08-1995	<i>Multimedia Communications and Performance</i>
T1.801.02-1996	<i>Digital Transport of Video Teleconferencing/Video Telephony Signals— Performance Terms, Definitions, and Examples</i>
T1.801.04-1997	<i>Multimedia Communications Delay, Synchronization, and Frame Rate</i>
T1.X1	Definitions provided by T1X1 from their experts on synchronization and timing.

Table B-3. Committee T1 Technical Reports Cited in the Glossary.

T1 Report No. –Date	T1 Report Title
T1-Rpt 11-1991	<i>A Technical Report on Switched Exchange-Access Network Traffic Availability Performance</i>
T1-Rpt 16-1992	<i>A Technical Report on DS3 Transport for Contribution Application of System M-NTSC Television Signals Analog Interface and Performance Objectives</i>
T1-Rpt 17-1993	<i>A Technical Report on the Effects of SONET on Payload Output Jitter</i>
T1-Rpt 19-1993	<i>A Technical Report on Wireless Personal Communications: Transmission Performance Guidelines for Speech and Voiceband Data</i>
T1-Rpt 22-1993	<i>A Technical Report on Privacy and Authentication Objectives for Wireless Access to Personal Communications</i>
T1-Rpt 24-1993	<i>A Technical Report on Network Survivability Performance</i>
T1-Rpt 25-1993	<i>A Technical Report on Test Patterns for DS1 Circuits</i>
T1-Rpt 28-1994	<i>A Technical Report on High-Bit-Rate Digital Subscriber Lines (HDSL)</i>
T1-Rpt 30-1994	<i>A Technical Report on UPT Numbering and Addressing in World Zone 1</i>
T1-Rpt 31-1994	<i>A Technical Report on Delay and Echo Performance Guidelines for SONET Self-Healing Rings During Diversity Routing</i>
T1-Rpt 34-1994	<i>A Technical Report on Network Capabilities, Architectures, and Interfaces for Personal Communications</i>

T1 Report No. -Date	T1 Report Title
T1-Rpt 37-1994	<i>A Technical Report on Line Break Detection (LBD)</i>
T1-Rpt 38-1994	<i>A Technical Report on Analysis of FCC-Reportable Service Outage Data</i>
T1-Rpt 39-1995	<i>A Technical Report on Termination Devices for Digital Data at 64 kbit/s and Subrates</i>
T1-Rpt 41-1995	<i>A Technical Report on UPT Routing for the Networks Served by the North American Numbering Plan</i>
T1-Rpt 44-1995	<i>A Technical Report on Security Management Guidelines to Support PCS</i>
T1-Rpt 46-1996	<i>A Technical Report on Transmission Performance Standards Issues Regarding the Introduction of ATM Technology into Networks Supporting Voiceband Services</i>

Table B-4. Other References Cited in the Glossary

X9 References

- X9.5 – Financial Institution Numbering System, 1994
- X9.8 – Personal Identification Number (PIN) Management and Security, 1995
- X9.9 – Financial Institution Message Authentication, 1994
- X9.15 – Specifications for Financial Message Exchange Between Card Acceptor and Acquirer, 1996
- X9.17 – Financial Institution Key Management, 1995
- X9.19 – Financial Institution Retail Message Authentication, 1996
- X9.23 – Encryption of Wholesale Financial Messages
- X9.24 – Financial Services Retail Key Management
- X9.26 – Financial Institution Sign-On Authentication for Wholesale Financial Services
- X9.30-1 and 30-2 – Public Key Cryptography Using Irreversible Algorithms, 1997
- X9.31 – Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry, 1998
- X9.49 – Secure Remote Access to Financial Services for the Financial Industry, 1998
- X9.52 – Triple Data Encryption Algorithm Modes of Operation, 1998
- X9.57 – Public Key Cryptography for the Financial Services Industry, Certificate Management, 1997

Web References

<http://webopedia.internet.com>

[accessed August 17, 2000]

http://home.netscape.com/newsref/std/cookie_spec.html

[accessed August 17, 2000]

http://www.x9.org/subcomms/x9f/documents/public/doc_x9f.html

[accessed October 27, 2000]
