

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

DESIGN AND IMPLEMENTATION OF A WORLD WIDE WEB CONFERENCE INFORMATION SYSTEM

by

Michael D. Chalfant, Jr.
Kevin M. Coats

September, 1996

Thesis Advisor:

Monique P. Fargues

Thesis Co-Advisor:

Kishore Sengupta

Approved for public release; distribution is unlimited.

19970407 018

DTIC QUALITY INSPECTION

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 1996	3. REPORT TYPE AND DATES COVERED Master's Thesis		
4. TITLE AND SUBTITLE *Design and Implementation of a World Wide Web Conference Information System		5. FUNDING NUMBERS		
6. AUTHOR(S) Michael D. Chalfant, Jr., Kevin M. Coats				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) The Asilomar Conference on Signals, Systems and Computers is a technical conference dealing in signal and image processing, communications, sensor systems, and computer hardware and software. Sponsored by the Naval Postgraduate School and San Jose State University, in cooperation with the IEEE Signal Processing Society, the conference is held annually at the Asilomar Conference Facility in Pacific Grove, California. Although the Asilomar Conference is oriented toward computers and new technology, it has yet to exploit the full capabilities of the Internet. The purpose of this thesis is to: a. Analyze the processes involved in the Asilomar Conference on Signals, Systems, & Computers, b. Improve the article submission and review process, c. Outline a target information system, d. Implement a portion of the target system. Two major portions of the target system are implemented using an IBM compatible PC: 1) the ability for authors to submit abstracts and summaries via the Internet, 2) to allow conference administrators to manage the database via the Internet. Dynamic World Wide Web pages are created using Borland Delphi as the programming base, O'Rielly's WebSite as the web server, and two Common Gateway Interface elements for Delphi recently developed by Ann Lynnworth of HREF Tools Corp. The portions implemented lay the foundation for a system that could revolutionize the way conferences are conducted by unleashing the power of the Internet.				
14. SUBJECT TERMS: World Wide Web, Relational database, Information System			15. NUMBER OF PAGES 508	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

Approved for public release; distribution is unlimited.

**DESIGN AND IMPLEMENTATION OF A
WORLD WIDE WEB
CONFERENCE INFORMATION SYSTEM**

Michael D. Chalfant, Jr.
Lieutenant, United States Navy
B.S. Business and Management, Tulane University, 1988

Kevin M. Coats
Lieutenant, United States Navy
B.S. Mathematics, University of Texas, 1986

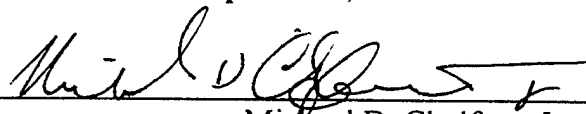
Submitted in partial fulfillment
of the requirements for the degree of

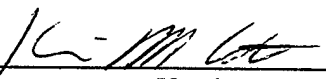
**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY
MANAGEMENT**

from the

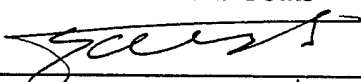
**NAVAL POSTGRADUATE SCHOOL
September, 1996**

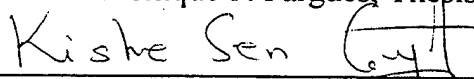
Authors:

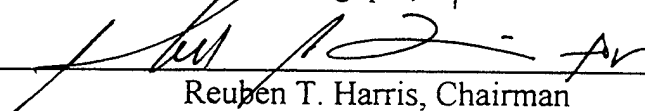

Michael D. Chalfant, Jr.


Kevin M. Coats

Approved by:


Monique P. Fargues, Thesis Advisor


Kishore Sengupta, Co-Advisor


Reuben T. Harris, Chairman
Department of Systems Management

ABSTRACT

The Asilomar Conference on Signals, Systems and Computers is a technical conference dealing in signal and image processing, communications, sensor systems, and computer hardware and software. Sponsored by the Naval Postgraduate School and San Jose State University, in cooperation with the IEEE Signal Processing Society, the conference is held annually at the Asilomar Conference Facility in Pacific Grove, California. Although the Asilomar Conference is oriented toward computers and new technology, it has yet to exploit the full capabilities of the Internet.

The purpose of this thesis is to:

- a. Analyze the processes involved in the Asilomar Conference on Signals, Systems, & Computers,
- b. Improve the article submission and review process,
- c. Outline a target information system,
- d. Implement a portion of the target system.

Two major portions of the target system are implemented using an IBM compatible PC: 1) the ability for authors to submit abstracts and summaries via the Internet, 2) to allow conference administrators to manage the database via the Internet. Dynamic World Wide Web pages are created using Borland Delphi as the programming base, O'Reilly's WebSite as the web server, and two Common Gateway Interface elements for Delphi recently developed by Ann Lynnworth of HREF Tools Corp. The portions implemented lay the foundation for a system that could revolutionize the way conferences are conducted by unleashing the power of the Internet.

TABLE OF CONTENTS

I. INTRODUCTION	1
A. OBJECTIVES	1
B. OUTLINE	1
II. BACKGROUND	3
A. CONFERENCE OVERVIEW	3
B. STRUCTURAL ORGANIZATION OF THE CONFERENCE	4
C. PROCEDURAL FLOW OF THE CONFERENCE	5
1. Attendees	5
2. Authors	6
3. Organizers	6
4. Reviewers	7
III. METHODOLOGY	9
A. BASELINE ASSESSMENT	9
1. Business Functions	9
2. Information	11
3. Applications	11
4. Technology	13
5. Survey Results	14

6.	Summary	14
B.	TARGET SYSTEM	15
1.	User Goals	16
2.	Future Possibilities	18
C.	MIGRATION TO THE TARGET	19
1.	Path to Target	19
2.	Thesis Goal	20
IV.	IMPLEMENTATION	21
A.	ISSUES	21
1.	Information Flow	21
2.	Compatibility with the Current System	22
3.	Platform Selection	22
a.	UNIX	22
b.	Desktop PC	23
c.	System Selection	23
4.	Application Selection	24
a.	Common Gateway Interface	24
b.	Database	25
c.	Web Server	25
5.	Security	26
a.	System Protection	26

b.	Database Protection	27
c.	Application Protection	28
B.	SYSTEM DESIGN	29
1.	System Overview	29
a.	System Flow Diagrams	30
2.	Article Submission Sub-system Description	34
a.	Author Submission Sub-system Introduction	34
b.	The Database Query	35
c.	Article Submission	37
d.	Confirmation Page	37
3.	Administration Sub-system Description	38
a.	Log-on and Main Menu	39
b.	Review Changes to Author Data	39
c.	Edit Author Data	41
d.	Edit Article Data	43
e.	Edit Sessions	46
f.	Edit Keywords	48
g.	Invite Articles	51
h.	Accept / Reject Articles	52
i.	Assign Articles to Sessions	54
j.	Assign Presentation Times to Articles	55
k.	Modify Passwords	56

l.	Print out a Session list	58
m.	Print out Article Data	58
n.	Print out Accept / Reject letter	59
o.	Print Abstract Catalog	61
4.	Applications	62
a.	O'Reilly & Associates' WEBSITE Web Server	62
b.	FTPD FTP Server	63
c.	Borland's PARADOX Relational Database	64
5.	System Web Pages	65
V.	CONCLUSION & RECOMMENDATIONS	115
A.	CONCLUSION	115
B.	UPGRADES TO CURRENT SYSTEM	115
C.	NEW FEATURES	116
	APPENDIX A. DELPHI TUTORIAL	119
A.	OVERVIEW	119
B.	SAMPLE APPLICATION	122
	APPENDIX B. APPLICATION TEMPLATES	137
	APPENDIX C. SALSA DATABASE DESIGN TABLES	143

APPENDIX D. SURVEY RESULTS	153
APPENDIX E. ARTICLE SUBMISSION SUB-SYSTEM CODE	159
APPENDIX F. ADMINISTRATION SUB-SYSTEM CODE	211
LIST OF REFERENCES	493
INITIAL DISTRIBUTION LIST	495

I. INTRODUCTION

The Asilomar Conference on Signals, Systems and Computers is a technical conference dealing in signal and image processing, communications, sensor systems, and computer hardware and software. Sponsored by the Naval Postgraduate School and San Jose State University, in cooperation with the IEEE Signal Processing Society, the conference is held annually at the Asilomar Conference Facility in Pacific Grove, California.

A. OBJECTIVES

The objectives of this Thesis are to:

- 1) Analyze the processes involved in Asilomar Conference on Signals, Systems, & Computers,
- 2) Improve the article submission and review process,
- 3) Outline a target information system,
- 4) Implement a portion of the target system.

B. OUTLINE

Section II covers background material such as, a description of the Asilomar Conference and its organization, the submission process for authors submitting papers,

and the article review and acceptance process. The system development process is explained in Section III. The following steps were used in our development methodology:

- 1) Develop a baseline assessment - what exists now, how is it being used, and what users would like to see improved.
- 2) Define the target system - compile the information gathered during the baseline assessment and develop a system that improves the process for all concerned.
- 3) Select a migration path - decide how to move from the existing system to the target.

Section IV discusses the implementation of the target system. This section also includes system flow diagrams and a description of every program element that comprise the target system. Conclusions and recommendations for further enhancements to the system are covered in Section V.

II. BACKGROUND

Three views will be presented to develop a detailed understanding of the present conference system: 1) an outline of the procedures required to conduct a conference; 2) a description of the structural organization of the different committees which guide the conference; and 3) an overview of the processes from the participants' perspective.

A. CONFERENCE OVERVIEW

Approximately one year before the conference convening date, the Conference publishes a "call for papers" to solicit articles for presentation. Interested individuals submit an abstract and summary of the articles they propose to present. A few months before the conference convenes, the collected submissions are reviewed for acceptance. Selected article abstracts are published in a "Final Program and Abstract Catalog." The catalog is distributed to potential attendees to generate interest in the conference and provide an idea of the topics to be covered.

The actual conference takes place over a three-day period, with four half-days of presentations. Presentations are organized by topic area into sessions, with eight sessions going on simultaneously during each half-day. Each session will have 6-8 oral presentations or up to 16 poster presentations. If the number of presentations for a given topic is insufficient to fill an entire morning/afternoon, the session may be split into two

half-sessions. On average, about seventy presentations are given each half-day. Because eight sessions run concurrently, attendees are not able to hear every article presented. Following the conference, each registered participant is mailed a set of conference proceedings containing articles presented at the conference.

B. STRUCTURAL ORGANIZATION OF THE CONFERENCE

The conference is guided by two volunteer committees; 1) the conference committee and 2) the steering committee. Both are staffed by technical experts in the various fields of signal processing. The conference committee includes the Publicity, Publication, Coordination, and Technical Program Chairpersons (Chairs). The Technical Program Chair heads the Technical Program Committee which has final responsibility for developing the technical program and publishing the Proceedings. The Technical Program Committee structure is shown in Figure 2-1. Principal tasks of the Technical Program Chair, responsible for the technical content of the conference, include determining which topics are to be included in the "Call for Papers" and presiding over the review process. Responsibility for specific topic areas is delegated down to 6-7 Technical Area Chairs (TACs). Their primary task is to review submitted articles within their area and select those best suited for presentation. Each TAC is assisted by several Session Chairs who coordinate and schedule the individual sessions. Sessions Chairs invite 3-4 papers on topics which they feel are of particular interest. These "invited"

papers are automatically selected for presentation and are not subject to review.

The Steering Committee's role is to guide the long term goals of the conference. Its main contribution is to nominate the General Chair and the other conference committee Chairs. However, the Steering Committee is not directly involved with the technical program setup, therefore it is not discussed further in this document.

C. PROCEDURAL FLOW OF THE CONFERENCE

Participants in the Conference can be divided into four roles: attendees, authors, organizers, and reviewers. Organizers conduct the required administrative functions, reviewers select articles for presentation, authors write articles and make presentations, and attendees attend the conference. Each of these roles is accompanied by a distinct set of procedures that must be completed. A particular individual may fill more than one role.

1. Attendees

Attendees begin their participation by gathering information about the upcoming conference. Information is obtained on-line via an established World Wide Web (WWW) homepage or off-line via an established mailing list. Registration is done manually via the U. S. Postal Service, sometimes referred to as snail mail (s-mail).

Bank checks and cash are the only form of payment which the conference currently accepts. Once registered, attendees will receive an "Abstract Catalog" which describes the articles to be presented. They may peruse the catalog to decide which presentations to attend. Attendees receive a copy of the conference proceedings about two months after the conference.

2. Authors

An author is someone who wishes to present recent research results. Information on topics and the submission process are obtained from the "Call for Papers." Only a title, abstract, and extended summary are required to have an article considered by the Conference Technical Program Committee. Currently, submissions are accepted via electronic mail (e-mail) and s-mail. Unless invited, the article's abstract and summary are reviewed for selection, and authors are notified by mail regarding the acceptance or rejection of their article(s). If selected, they must turn in a complete copy of the article at the conference to be used for publishing in the conference proceedings.

3. Organizers

Organizers consist of the committee members and a single administrator. They are responsible for publishing information on the WWW homepage and for sending out

information to the mailing list. Registrations and submissions are collected and sorted. After the review session, the administrator compiles and sends out accept/reject notices and the abstract catalog. After the conference, proceedings are published by the Institute of Electrical & Electronics Engineers (IEEE) Computer Society Press. Abstracts and summaries (whether accepted or rejected) are discarded two months after the conference.

4. Reviewers

The Technical Area Chairs are responsible for reviewing all articles submitted within their topic areas. All reviews are done during a single, one-day meeting. Abstracts are the primary basis for judging articles, but the extended summaries are available if more information is sought. Topic areas are generally defined through "keywords." No comments, notes, or other documentation is maintained on the reviewer's thoughts or opinions of the submissions. The Technical and General Chairs participate in the review and resolve any disputes that may arise.

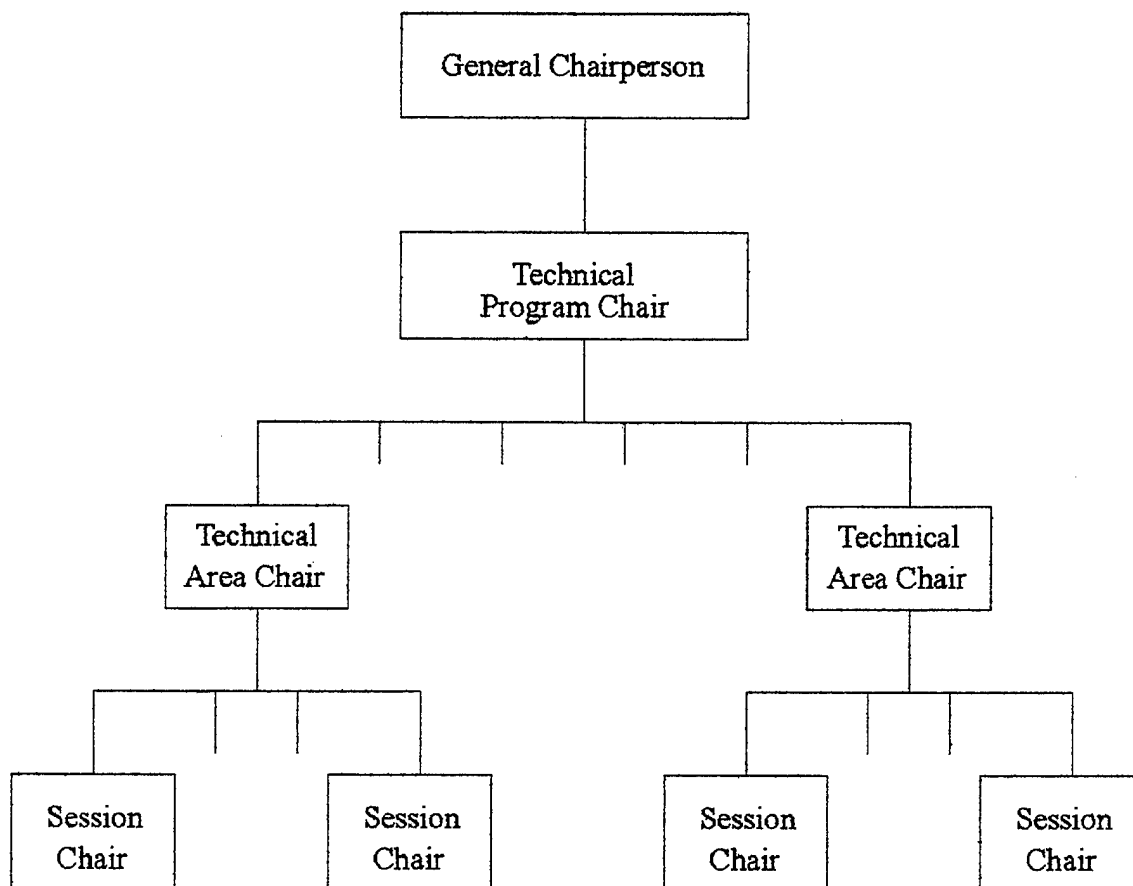


Figure 2-1. Asilomar Conference Organization.

III. METHODOLOGY

A. BASELINE ASSESSMENT

The assessment of the current Conference system includes a characterization of the organization's business functions and key processes. Data for this analysis was collected through interviews with organizers and a survey of 1995 Conference attendees.

1. Business Functions

Key business functions for the conference are: 1) attracting attendees and authors; 2) determining topic areas; 3) collecting article abstracts and summaries; 4) reviewing and selecting articles; and 5) disseminating the article's information. These functions are primarily the responsibility of the Conference Committee. Forces influencing these functions are improvements in technology, trends in research and development, and pressure on academic faculty to publish.

The Conference Registrar hired by the Conference Committee collects the article submissions. Each submission consists of an abstract (100-150 words) and an extended summary (500-1000 words). While abstracts are limited to text only, summaries may contain graphics and/or complex equations. ASCII text e-mail is sufficient for submitting abstracts, but is unable to handle the graphics/equations in a summary. Summaries may be submitted by either s-mail or e-mail in plain text ASCII format.

S-mail has no limits on content, but lacks the convenience of electronic transfer, especially at the receiving end where someone must then enter it into the Abstract Catalog. Technical Area Chairs travel to Monterey to review and select articles during a one-day, marathon review session. Abstracts and summaries are reviewed by the assigned Chair and the article is either accepted or rejected. If too many articles are accepted, the Technical Chair arbitrates a compromise. Not only is this expensive (the conference pays to fly everyone in), but the amount of time available is insufficient to thoroughly review the submissions. As a result, article selection may sometimes be based more on the reputation of the author than on the merits of the paper.

Dissemination of the articles takes place in two forms, presentation at the conference and publication in the conference proceedings. Putting on the presentations incurs the greatest effort and expense for the conference. Coordinating and paying for the resources required to bring all the players together in one location is also the most challenging problem. For attendees, the conference fee is generally trivial next to the cost of transportation, lodging, and missed work.

An abstract catalog is published and distributed prior to the conference. Still, most people who attend presentations are not familiar enough with the topic to effectively interact with the author. The most useful exchange of ideas often occurs in one-on-one interactions that take place outside the scheduled presentations.

Conference proceedings are a bulky set of documents that often end up on a bookshelf like a trophy, never to be taken down again. Most attendees are only interested in one or two articles, but their only option is to obtain the entire document.

As a result, more conferences have started to offer the proceedings in a CD-ROM format which allows readers to download only articles of interest. In addition, a large percentage of the conference fee is required to cover these printing costs.

2. Information

Dissemination of information is the primary purpose of this conference. The entire process is built around moving information from the authors (primary suppliers) to the attendees (primary customers). The flow of information throughout the conference process is as follows:

1. The conference publishes information on desired submissions,
2. Authors return information about their articles,
3. Reviewers determine whether the article is desirable for presentation,
4. Attendees come to the conference to obtain article information.

3. Applications

The analysis of applications will include processes which are not yet automated. Existing automated/on-line systems consist of an FTP server with which attendees download details about the conference (location, cost, etc.), an e-mail address to which plain text ASCII electronic submissions may be sent, and a World Wide Web homepage which provides a central point of reference. Non-automated processes include mailing

conference information to a previous-attendees mailing list, collecting article submissions through s-mail, determining topic areas, reviewing articles, publishing the hard-copy conference proceedings, and presenting the author's paper. Results of the analysis are presented graphically in Appendix B.

The current system for publicizing the conference will continue. On-line systems in place now will continue to be used to attract attendees and authors. Registration details will continue to be maintained on the WWW pages and FTP server. E-mail will be retained as an option for submitting articles (for a limited time), and for general communications. Recognizing that everyone will not have on-line access, the mailing list will also continue.

Determining topic areas is a subjective process and will not be modified. In the future, individuals may be allowed to suggest topics which interest them. However, collecting submissions is a prime candidate for re-engineering. Time and monetary costs for both the authors and the conference are too high. Authors must mail article submissions either in hard copy or electronic format. Conference administrators must collect the information, enter it into a database, collate the abstracts and summaries, and arrange and print the Abstract Catalog.

The review process is another good candidate for replacement. The time and expense to bring reviewers together to conduct the reviews may become unnecessary. By making the submitted articles available on-line, reviewers could read them whenever and wherever they desired, without having to travel to Monterey. The reviewing process would be better served by allowing reviewers more time to conduct reviews. A Decision

Support System (DSS) could eventually be developed to assist in the review and selection process.

The presentation does not need re-work, although improvements should be considered. Video broadcast and/or interactive would significantly strengthen the quality of the conference. Eventually, interactive systems may even mature to a point where it is no longer desirable to travel to distant locations for a conference. But for now, the social interactions and technical difficulties are enough to maintain the presentation as a main method of information dissemination.

Hard-copy conference proceedings would ideally be eliminated by a fully on-line system. An on-line archive of presented articles would be more efficient and cost effective. Attendees could decide whether they wish to receive a printed proceedings. Conference registration fees of those desiring a hard-copy would include printing costs. Registration fees for everyone else could be reduced by that amount.

4. Technology

The use of technology is presently limited, but there is now a strong desire to exploit the power of the Internet. All on-line services are run on a UNIX local area network (LAN) at the Naval Postgraduate School. The W. R. Church Computer Center's Visualization Lab provides the server for the Asilomar Conference WWW Home Page. In addition, a PC LAN is available in the Electrical and Computer Engineering (ECE) Department.

5. Survey Results

Attendees to the 1995 Asilomar Conference on Signals, Systems, & Computers were given the opportunity to complete a survey about their capabilities and opinions regarding an on-line conferencing system. The only respondents were individuals who presented articles at the conference. Since the initial stage of this project is geared toward the submission of articles to be considered for presentation, the narrow scope was acceptable. Views from a broader sample will be beneficial in the later stages. Of the one hundred and eighteen (118) surveys returned, only one did not support an on-line system. Most respondents have WWW access and Netscape was by far the browser of choice. Responses were very positive regarding web access to retrieve registration materials, submit article, and preview accepted abstracts. However, responders were unsure whether they would send comments to the authors about their articles. A summary of the results is included in Appendix D.

6. Summary

The Asilomar Conference on Signals, Systems, & Computers has begun the transition to the information age, but has taken only few small steps. Successfully re-engineering the processes that make up the conference will require a paradigm shift, the extent of which the members of the conference do not yet realize. Our project is viewed by most as simply an added convenience for users to be able to submit documents on-

line. However, a great deal of possibilities are being overlooked. This new system could eventually revolutionize the manner in which the conference is conducted. What is required is a willingness of the conference sponsors to explore the latest technology offered by the WWW.

Realistic measures of success of the new system are mostly subjective issues of user satisfaction. However, a real measurement will be made in cost savings, both in travel expenses for reviews and administrative savings. Measuring user satisfaction is always difficult. By placing a "Call for Papers" on the WWW homepage in addition to the usual mailing list, the widest possible dissemination of conference information is possible. Acceptance of the new system will be gauged by actual usage. A survey of respondents can be taken as well to discover how they obtained conference information.

On the review process, user satisfaction becomes even more difficult. One measure will be to count how many people review an article, on average. However, this is not completely accurate since more reviews do not automatically translate into "better" reviews. Satisfaction of the reviewers would be a good measure because they could gauge whether having a more relaxed atmosphere to do the review was helpful. They could also estimate their own ability to conduct the review.

B. TARGET SYSTEM

As stated in the introduction, part of the goal of this thesis was to develop a target system and implement a portion of it. Users' goals for the system were acquired from

the survey conducted and through meetings with conference staff members. These goals serve as the blueprint for the system design.

1. User Goals

The primary goal of the conference is to be able to collect and process article abstracts and summaries efficiently. There is no requirement for electronic submission, but the more people use the system, the greater the efficiency will be. To accomplish this task, the system must be able to:

- 1) reliably receive and number documents,
- 2) print documents,
- 3) maintain Abstracts separate from Extended Summaries,
- 4) sort incoming documents by Keywords,
- 5) send automated acknowledgments to submitter,
- 6) keep track of what is sent and received,
- 7) provide the staff access to submissions, but prevent unauthorized viewing.

Finally, maintainability is a concern. The target system for the Asilomar conference on Signals, Systems, & Computers consists of the following on-line services:

- 1) conference information,
- 2) registration and payment,
- 3) article submission,

- 4) article review,
- 5) article archiving.

1) Conference information

Information regarding the conference program, registration, submission process and local points of interest such as dining and hotels will be available on-line.

2) On-line Registration

Attendees not only receive information about the conference, they can register, reserve hotel rooms, rental cars, and dining arrangements. Everything will be purchased on-line through the convenience of electronic commerce. The system will also maintain a database of names and addresses of conference attendees.

3) On-line Submission

Authors can submit their articles and extended summaries from any Internet connection.

4) *On-line Review*

Articles are reviewed and selected on-line via the Internet with any standard web browser. Accept and reject letters are automatically printed once the reviewing process is complete.

5) *On-line Article Archive*

Articles are available for review and download over the WWW which replaces a need for hard copy proceedings.

2. Future Possibilities

The ultimate goal for the conference is a WWW system which covers every aspect of the Conference: information, registration, submission, and continual interaction on various topic areas through news groups, chat rooms, and e-mail. Eventually, on-line capabilities may replace the face-to-face conference. Articles may be submitted, reviewed, presented, and archived entirely on-line. The "conference" is no longer a physical meeting at a specific place or time. Interested individuals can submit articles and recorded presentations on a continual basis. Articles are maintained on server for access on demand. The social interactions are replicated through e-mail, news groups, and chat rooms. The exchange of information is an on-going process.

C. MIGRATION TO THE TARGET

Reliability in the system is paramount. Migration must occur with a minimum of risk to the conference, as data loss would be entirely unacceptable.

1. Path to Target

The five services discussed in Section III. D could be implemented independently, or concurrently. However, the logical serial path will be as follows:

- 1) Conference Information,
- 2) Article Submission,
- 3) Article Review,
- 4) Article Archive,
- 5) Registration and payment.

Re-engineering the review process is the highest priority because this is where the most time and money can be saved and the most value added to the system. However, on-line submissions must be accomplished first. Otherwise, administrative personnel would have to store the submissions electronically before they could be reviewed on-line. Archiving should follow submission because the on-line submission components will serve as catalog information for the archive. Registration and payment should be last simply because technology for universal electronic payment (digital cash) is still developing. On-line registration without on-line payment would only complicate the

process.

2. Thesis Goal

For the purpose of this thesis, the submission process and several administrative functions will be automated. Authors will not only submit information about their articles, they will also update their contact information maintained in the conference database. The system will rely on HTML 3.0 forms to collect the information and common gateway interfaces (CGI) to place the data into a database.

Implementation also requires the availability of several system administration functions. Necessary capabilities include maintaining passwords, accessing the database, updating the database as articles are reviewed and selected, and printing articles, accept/reject notices, and the abstract catalog. User-friendly interfaces and highly automated CGI's will be used to emphasize ease of use.

IV. IMPLEMENTATION

A. ISSUES

1. Information Flow

The current submission process is tedious for both the Asilomar Conference staff and the submitting author. Registration forms, abstracts, and extended summaries must all be sent on paper or via e-mail. The conference staff, consisting of one person at this time, must then sort through the myriad of papers and forms and enter all the information manually. Abstracts must either be retyped or scanned into the old computer system, and formatted for the abstract catalogue published at each conference.

The new on-line system will eliminate these and other hassles. All information is stored directly to a database with the exception of extended summaries which are stored as files in a separate directory. The author saves printing and mailing costs as well. The administration functions built in to the new system enables the administrator to manipulate the database from any Internet connection. The abstract catalogue is automatically formatted and can be printed with the click of a button. As more people use the system, the entire registration process becomes more timely, accurate, and simple.

2. Compatibility with the Current System

The Asilomar conference currently maintains an information-only web site. All conference registration, abstract submissions, and database functions are handled manually. The system we have created can be accessed from the current site and will automate a majority of the processes. Our system integrates the existing database and can be used in conjunction with all manual processes.

3. Platform Selection

One of the first major decisions of this project involved the selection of the hardware platform best suited to our needs. We considered two alternatives: 1) the Naval Post Graduate School's UNIX system, 2) to purchase a Personal Computer to connect to the school's Ethernet network. Factors affecting the decision included security, price, available software, and performance. The advantages and disadvantages are delineated by platform.

a. UNIX

The UNIX system offered speed and bandwidth with no monetary expenditure, but had some serious drawbacks. The UNIX system did not have a functional database installed. An effort was underway to implement the Oracle database,

however, the completion date was unknown at the time. Also at that point, the NPS system administrator was not in favor of allowing students to run Perl executables from their accounts due to security concerns. Such a limitation was unacceptable for our project. The system administrator acquiesced to a trial period for our project, but would not guarantee a permanent solution.

b. Desktop PC

The PC offered access to the latest software development, Internet access, and database programs, but at a cost. Software costs alone were expected to exceed \$500 in addition to \$3000 for hardware. The PC also had speed and bandwidth limitations.

c. System Selection

Based on the unstable commitment from the UNIX system administrator and the proliferation of 4th generation application development languages available for the PC, we elected to design our system on a PC running Windows 95. The decision was a wise one because in December of 1995, the UNIX system was compromised and security restrictions were increased as a result.

4. Application Selection

a. Common Gateway Interface

Executable programs on the Internet must follow one of a few programming protocols. Of those, the most prevalent is the Common Gateway Interface (CGI). Many different languages can support CGI functions, with ease of use varying greatly among them. Once we decided on a PC platform, the list of CGI languages narrowed. There were three major contenders: 1) Cold Fusion by Allaire, inc., [Ref. 1] 2) Delphi by Borland [Ref. 2] and, 3) DOS Perl [Ref. 3].

Cold Fusion is a brand new product, still in its first version with little documentation available. It was originally developed specifically as an interface between a database and the Internet. It came highly recommended from several classmates who had recently begun using it to develop web applications. Cold Fusion offered an easy to use interface, a language very similar to HTML, and compatibility with the latest version of Microsoft's Access database. However, it lacked powerful programming tools and the flexibility of a true application development language.

Delphi is a Windows application development tool. The programming language is essentially Pascal, but Delphi adds drag and drop programming elements, drop down lists, and a point and click interface to automate much of the coding process. Delphi offers seamless connectivity with Borland's Paradox database, which coincidentally, was the database in use at the time by conference administrators.

Unfortunately, the CGI programming elements were new, and thus, poorly documented. These elements were developed outside of the Borland corporation by Ann Lynnworth of HREF Tools Corp.[Ref. 4], so no technical support was available. Once again, classmates who had just begun using Delphi with these new elements recommended them as the solution for our system.

DOS Perl offered none of the advantages listed above and since it is not a Windows application, the power and flexibility of Windows '95 would be lost. As a result, it was eliminated as a choice early on.

Delphi was chosen as the language to be used because of the programming flexibility and the compatibility with the database already in use by the conference staff. As it turns out, Delphi's programming robustness was needed to its full extent to create the system we did.

b. Database

The choice of database was linked to our selection of a CGI language. Cold Fusion interacted seamlessly with Microsoft Access as Delphi did with Borland Paradox. Once the decision was made to use Delphi, Paradox was the logical choice.

c. Web Server

Selecting a web server was a difficult choice, as there are many products

available with a wide variety of functions and prices. From the many, two stood out as having the functionality and compatibility desired for our system. Those were Netscape HTTPD [Ref. 5] and O'Reilly's Website [Ref. 6]. Our choice was made based on word of mouth recommendations from classmates experienced in using both products. Website offered a better combination of ease of use, flexibility, performance, and price. Website was also compatible with both Cold Fusion and Delphi programming languages.

5. Security

System protection is a primary concern in this project. Authors from literally around the world will be submitting information, abstracts, and extended summaries via this system which will be used to accept or reject their papers. The database will be used to maintain mailing address lists and an archive of abstracts to be published in the conference proceedings. It is therefore imperative that all data received is accurate and secure.

a. System Protection

Physically, the system is located in the Digital Signal Processing laboratory at the Naval Postgraduate School. This area is only used by students working on thesis topics and is locked at night. The PC is password protected on boot-up and uses a password protected screen saver. While certainly not foolproof, the protection

offered will keep the average mischievous person at bay. The expected threat here is low, because of the system's location and the student's respect for the work of others. The true threat lies from Internet users notorious for attempting to hack systems for the fun of it. This issue is covered as it applies to specific components of our system.

b. Database Protection

The primary means of database protection is timely backups. With the tape backup, the entire contents of the database will be copied on a regular basis. Thus, for any catastrophic event, the data loss will be minimal. As any system designer knows, a computer is only secure if it has no connections to it and is locked in a vault that has no doors. A PC connected to the Internet has many vulnerabilities, but we have minimized those through the design of the system itself. Database functions are accessed through the administrative section of the system. Access to this area is controlled by a password form.

One problem with Internet applications is that once someone knows the name of an executable file, he may run that file by typing the path information into the URL field of his web browser. If that file exists, it will be executed. We have programmed into every executable a conditional check of a flag variable. This flag can only be set by entering the appropriate login and password combination on the first screen. Once the password has been entered correctly, a flag is set that will allow a user to access all database functions. The programs will execute without the flag being set,

however each program will only print out a warning message and return the user to the login screen. So, even if an unauthorized user discovers the names of the executables, there is no way (that we know of) to set the flag variable without the proper login and password.

The threat still exists of a brute force attack on the login and password combinations. A brute force attack is one where a hacker attempts to guess the login and passwords of legitimate users. This small threat can be minimized even more through limiting the number of users (currently two), frequent password changes, and using good password selection techniques – including upper and lower case, digits, and special characters within the password.

c. Application Protection

(1) Website Security. The denial of access to the PC's hard drive is the responsibility of the server software, Website. This program controls all Internet access functions of the system. Inherent in its design is a restriction that limits users to directories higher than the C:\Website directory. In other words, when a browser accesses the system, the lowest directory available to it is either C:\Website\Htdocs or C:\Website\cgi-win. It is not possible to access the root directory or any other directory than those stated above. Website also does not allow file uploads (i.e. a user sending a file to our computer) or modification of files on our system. Therefore, a user may look at and execute any file that we place higher in the directory structure than C:\Website,

but nothing else.

(2) **FTPD Security.** One of the requirements of the system was that authors must be able to upload Post Script files of their extended summaries. This required FTP server software be installed. This software inherently adds security risks because it grants users the ability to upload and download files as well as to view and modify the contents of a directory (i.e. delete and rename). Security controls built in to the software package allow the server administrator to restrict a user's access rights. Currently we have limited all users' rights to file upload only. Users cannot view the contents of the directory, modify its contents, or change directories. The strength of this protection lies within the strength of the program itself. We are currently using a shareware version of WFTPD [Ref. 7], a popular FTP server. By limiting the user's ability to change directories, modify directory contents, and view directory contents, the ability to corrupt another author's data is slight.

B. System Design

1. System Overview

The entire Asilomar Conference System consists of the following elements:

- 1) The Asilomar Conference information homepage
- 2) The article submission sub-system
- 3) The administration sub-system

- 4) The article review sub-system.

Prior to this thesis, only the first element existed. The second and third have been implemented and comprise the bulk of this thesis. The fourth element will be completed in a follow-on project.

a. System Flow Diagrams

The following diagrams show the relationship of: 1) the Asilomar Conference System as a whole, 2) the article submission sub-system and, 3) the administration sub-system. The information homepage and the article review sub-system were not within the scope of this thesis, and as such, are not included.

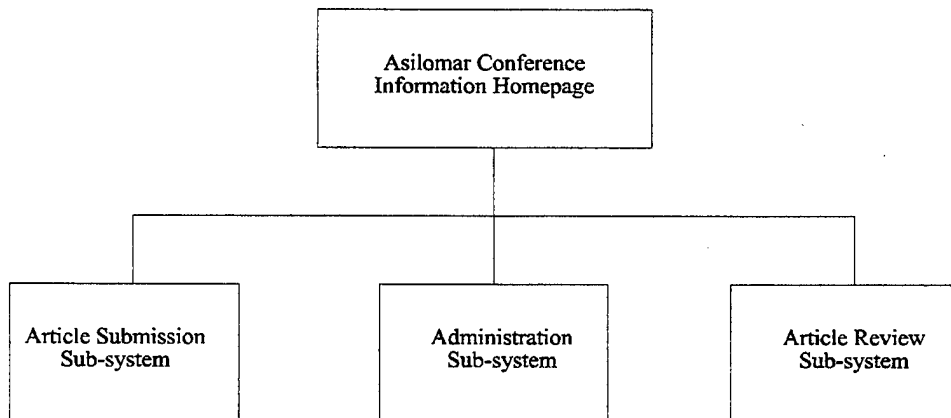


Figure 4-1. Asilomar Conference System.

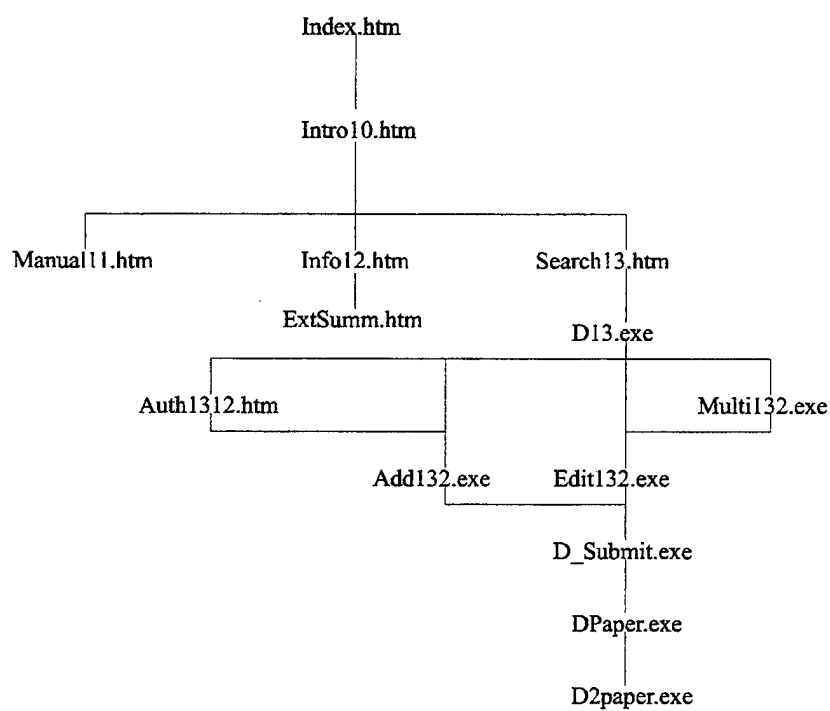


Figure 4-2. Article Submission Sub-system.

2. Article Submission Sub-system description

The Article Submission Sub-system is a combination of HTML pages and Delphi executable programs that form the user interface for the conference registration process. Images of each HTML page are at the end of this chapter and are referred to in the program descriptions below.

Via this system, users can add and update personal information as well as submit articles and paper summaries for review. Security is not a concern in this sub-system because any changes made to a user's personal information is sent to a temporary database. The update is only made permanent when reviewed by the system administrator.

a. Author submission sub-system introduction

The introduction HTML page (**Intro10.htm**) welcomes the user and offers three hypertext link options as shown in Fig. 4-4. All figures appear at the end of Section IV.

- 1) Obtain information for on-line submission (**Info12.htm**),
- 2) Obtain information for manual submission (**Manual11.htm**),
- 3) Transfer to the on-line submission process (**Search13.htm**).

(1) On-line submission information (**Info12.htm**), Fig. 4-5. This page explains the requirements and process necessary for a user to submit information with the WWW-based system. There is a hypertext link to **Extsumm.htm** (Fig. 4-6) which is an HTML page that explains what an extended summary is. In addition, this HTML page contains two links which enable a user to download a file transfer protocol (FTP) program in either Windows 3.1 or '95 format, as FTP is required for users who wish to upload an extended summary for review.

(2) Manual submission information (**Manual11.htm**) page, Fig. 4-7. This HTML page briefly outlines the steps necessary for an author to submit personal and article data via U.S. mail.

(3) Submit information on-line (**Search13.htm**), Fig. 4-8. The first step in the on-line submission process is a check of the database for the author's name. This HTML page asks the user to input his first and last name and middle initial. The HTML form then calls a Delphi executable (**D13.exe**), which conducts the database query.

b. The database query

The executable (**D13.exe**), receives the author's name and queries the **Author.db** database table. The query has three possible outcomes: 1) the author's name is not found, 2) a unique match is found, 3) multiple matches are found. For each outcome, the program will create a specific HTML form.

Outcome one: a blank HTML input form is created (Fig. 4-9) that allows

the author to input his/her personal information. Once submitted, this form calls an executable program (**Add132.exe**) which stores the information directly to the permanent author table (**Author.db**).

Outcome two: an HTML form is generated that contains all known information about the author filled in the appropriate blanks (Fig. 4-10). The author may then either accept the information as is, or make changes to it. When submitted, this form calls an executable (**Edit132.exe**) which stores changed information to a temporary database to be reviewed by the system administrator. One other possibility that had to be accounted for was when the database returned a unique entry that referred to a different user. If this is the case, the author can select a hypertext link to an HTML form (**Auth1312.htm**) which allows him/her to enter his personal information (Fig. 4-11). Once the form is filled in, **Add132.exe** is called to store the information to the permanent database, **Author.db**.

Outcome three: if more than one match is found, the user is presented with a table of names from which he/she can choose a selected record by clicking on the hypertext entry (Fig. 4-12). Doing so will call an executable (**Multi132.exe**) which generates the same form as the unique entry case and performs the same functions. Once this form has been completed, **Edit132.exe** is called to store the information to the temporary database table (**Tempauth.db**).

c. Article submission

At this point, the author's information has been stored in either the permanent author table by the executable program **Add132.exe**, or into the temporary author table by **Edit132.exe**. Both of these programs, when complete, generate an information page that tells the user he/she has completed the personal information process and should proceed to the article submission by clicking the button which calls the next executable (Fig. 4-13).

The executable **D_Submit.exe** generates an HTML form that allows the user to input an abstract, title, keywords, and other relevant article information (Figs. 4-14, 4-15). When submitted, the information is redisplayed by the executable **Dpaper.exe** (Figs. 4-16, 4-17). The user is given a message stating that once this form is submitted, the information cannot be changed.

d. Confirmation Page

Once the user has confirmed the data entered into the form created by **Dpaper.exe**, a confirmation page is created by the executable **D2paper.exe** (Fig. 4-18). This final page contains a unique registration number, instructions on how to upload an article summary, and Log-in information for the FTP server.

3. Administration Sub-system description

Functions associated with maintaining the database, administering the system and handling the submissions are located in the administration sub-system. Security is a major concern within this portion of the system because of database integrity issues.

The administration sub-system is password protected, and uses only dynamically created HTML pages. In other words, any HTML form that the user sees is created by Delphi executable programs. This prevents would be hackers from accessing database management functions because each program requires an access code to create an HTML form. The access code is generated once the user has entered a correct login and password. The code is then passed from executable to executable transparent to the user. Anyone attempting to execute any of the programs without having first entered a correct login and password is given a warning message instead of the HTML form the program would normally create. The intent is not to have an air-tight system, rather to prevent the casual user and novice hacker from accessing the database. Encryption is not used, so the system is still vulnerable to sniffers gaining passwords and access codes. Future improvements to combat this vulnerability could include a one-time password system in which passwords are only valid for one session, after which, a new password must be used. Another solution is to install a firewall that limits access to the administration sub-system based on the user's location. The technology for both these improvements is currently available.

a. Log-on and main menu

The first page in the admin sub-system is an HTML page (**Admin20.htm**) that declares the purpose of the page and includes a form for entering a USERID and PASSWORD (Fig. 4-19). This is the only HTML page in the admin sub-system not created by a Delphi executable program.

A CGI executable takes the Form input (USERID and PASSWORD), verifies the user, and generates an HTML page which contains the Administration Main Menu (Fig. 4-20). The menu consists of buttons, each of which evokes a series of executables for the respective function. Failure to provide a valid password generates an HTML page which warns against unauthorized access and allows a retry.

b. Review changes to Author data

One of the functions of the Submission Sub-system is to allow an author to update his/her personal information (see chap. IV, sec. 2c). These changes are stored in a temporary table called **Tempauth.db**. This function allows the System Administrator to review the submitted changes before they become permanent. If the entries appear to have been changed maliciously, they can be rejected. Otherwise, they are accepted and become part of the permanent record. This process requires four executables:

- 1) Change Author (**AuChange.exe**)

- 2) View changes (**ViewChng.exe**)
- 3) Make the change (**MakeChng.exe**)
- 4) Do not make the change (**NoChange.exe**)

(1) Author Changes (**AuChange.exe**). This program is called from the Admin Main Menu to retrieve any data in the **Tempauth.db** table and present it in an HTML form with selected information about the Author (Fig. 4-21). The record number of the change is presented as a hypertext link to the record information. The System Admin chooses the author whose record changes they wish to review by clicking on the hypertext record number.

(2) View changes (**ViewChng.exe**). Called from the dynamically created Author Changes Table (**AuChange.exe**), this program presents the submitted changes side-by-side with the permanent data (Fig. 4-22, 4-23). If acceptable, the “Accept Changes” button calls **MakeChng.exe** to replace the permanent data. Otherwise, the “Discard Changes” button calls **NoChange.exe** which discards the changes.

(3) Make the Change (**MakeChng.exe**). This function replaces the permanent Author record with the submitted changes. The data in the **Tempauth.db** Table is discarded after the change is made. A confirmation page is sent with the option to “review more changes” which links back to **AuChange.exe**.

(4) Do not make changes (**NoChange.exe**). This function discards the changes without modifying the permanent Author record. A confirmation page is sent with the option to “review more changes” which links back to **AuChange.exe**.

*c. **Edit Author data***

This function allows the system administrator to directly access Contact Author data held in the **Author.db** table. After presenting a form for entering the first name, last name, and middle initial of the desired Author, a series of screens is presented to ensure selection of the proper record. Once the Author has been identified, the information is presented in a Form. Changes made to the Form go directly into the permanent record.

Five programs are required to implement this function.

- 1) Input Author name (**InputAu.exe**)
- 2) Search the database and show results (**AuthSrch.exe**)
- 3) Add a new author (**AuAdd.exe**)
- 4) Resolve multiple matches (**AuMulti.exe**)
- 5) Edit the Author (**AuEdit.exe**)

(1) Input Author Name (**Inputau.exe**). This program generates an HTML page with a form to submit the author name whose record will be edited (Fig. 4-24).

(2) Search the database (**AuthSrch.exe**). After searching the database for the given author name, this program branches according to how many matches were found. If no match to the submitted author name exists in the database, a blank author data form is generated (Fig. 4-25). Submitting this form calls **AuAdd.exe**. A single match generates an author data form which contains the database fields (Fig. 4-26). Submitting this form calls **AuEdit.exe**. When multiple matches to the name are found, a table containing all of the matches is generated (Fig. 4-27). Each match has a hypertext link which calls **AuMulti.exe**.

(3) Add a new Author (**AuAdd.exe**). Called by submitting the new author form, this program takes the form data and creates a new author record. A confirmation page is sent with the option to “Edit another Author” which links back to **InputAu.exe**.

(4) Multiple Matches (**AuMulti.exe**). This program is called by the hypertext links on the multiple matches table. The record number passed by the hypertext link is used to retrieve an author’s data which is presented in an HTML form for editing. Submitting this form calls **AuEdit.exe**.

(5) Edit an Author (**AuEdit.exe**). This program is called after author data has been edited and needs to be updated in the **Author.db** table (Fig. 4-26). If a unique match was found during the initial search, then **AuEdit.exe** is called by the form

generated by **AuthSrch.exe**. If there were multiple matches found, **AuEdit.exe** is called from the form generated by **AuMulti.exe**. To protect the permanent database, these changes go into a temporary table called **TempAuthor.db**. The system Administrator must review and approve the changes before the permanent database is modified.

*d. **Edit Article Data***

This function allows direct access to Article data held in the **Submissi.db** table. Three options are presented for finding a specific article record. An article may be chosen from a list of all the articles in the database, from a list of articles which meet certain user-selected criteria, or from a list of articles submitted by a certain author. Whichever option is chosen, the user is presented with a table of article titles that are hypertext links to the article record. Once the article has been identified, the information is presented in an HTML form. Changes made to the form are submitted directly into the **Submissi.db** table. The Edit Article Data function has the following components:

- 1) Edit an Article (**ArtlEdit.exe**)
- 2) Search for the Article (**ArtlSrch.exe**)
- 3) Article by Author Search (**ArtlAuth.exe**)
- 4) View the Article (**ArtlView.exe**)
- 5) Post the Article (**ArtlPost.exe**)

(1) Edit an Article (**ArtlEdit.exe**). Called from the main menu, this program creates an HTML page with three selection options (Fig. 4-28). To generate a table with all the article titles, a button links to **ArtlSrch.exe** with a hidden action flag set to "all." To specify criteria for the article, a form is presented with four fields: invited, accepted, keyword, and session. Each field has a drop-down box of choices. The option to not search on the field is available on each. Choices for "invited" and "accepted" are "yes," "no," and "do not search on this field." The "keyword" search criterion has all the keywords in the **Keyword.db** table, and "session" has all the sessions in the **Session.db** table. The submit button calls **ArtlSrch.exe** with the hidden flag set to "article" and performs a query on the selected fields. If each of the four search fields is set to "do not search on this field," the result is a table containing all article titles.

To generate a table of articles submitted by a specific author, a form is available to input the name. The submit button for this form also calls **ArtlSrch.exe**, but the hidden flag is set to "author."

(2) Search for the Article (**ArtlSrch.exe**). The action performed by this program depends on the hidden action flag sent by the calling program. A flag set to "All" generates a table containing every title in the **Submissi.db** table. Each title is a hypertext link to **ArtlView.exe** with that article's record number (Fig. 4-29). If "Article" is the flag's value, the **Submissi.db** table is queried based on the criteria ("invited", "accepted", "keyword", or "session") submitted in the previous form, and a similar table is generated. If the flag is set to "Author," a search is conducted for matches to the

submitted name. The author search is rather complicated. Author names are stored in two different locations, depending on whether they are the “contact” author for the paper, or an “other” author. First, the **Author.db** table, which contains only contact authors, is queried for matches. If there are no matches, then the **Submissi.db** table is searched for “non-contact author” matches and the results are displayed in an HTML table. If there is a contact author match, an HTML table is generated with a list of those matches so the exact author can be determined. The authors in the created HTML table are hypertext linked to **ArtlAuth.exe** so that the query will search through the non-contact authors as well. If the requested author’s name appears on the list, but belongs to a different author with the same name, a button is available which links to **ArtlAuth.exe**, and a flag set to “not a contact author” so the search will continue.

(3) Article by Author (**ArtlAuth.exe**). This program is called when **ArtlSrch.exe** finds a match in the **Author.db** table. Two queries are done, one to find the articles with contact author matches, and one for “other” authors. A table of article titles is generated with the results of each of these queries (Fig. 4-30). Each title is a hypertext link to **ArtlView.exe** with that article’s number. If the “not a contact author” flag is set, then the contact author query is not executed and only the non-contact authors are searched.

(4) View the Article (**ArtlView.exe**). Called by the hypertext link from either table of article titles generated, this program displays the selected article’s data in

a form. Using the article record number passed via the hypertext link, the article data is retrieved and presented in an HTML form. The user may then edit any information displayed. The submit button on this form calls **ArtlPost.exe**.

(5) Post the Article (**ArtlPost.exe**). Called from **ArtlView.exe**, article data is retrieved from the form and placed in the **Submissi.db** table, overwriting previous data. A confirmation page is sent with the option to "Edit another Article."

e. Edit Sessions

Sessions for identifying topic areas are maintained in the **Session.db** table. Although sessions remain constant for a given conference, they may change from year to year. Through this function, sessions may be modified, added or deleted.

- 1) Choose Session (**Sessions.exe**)
- 2) Edit a Session (**SessEdit.exe**)
- 3) Edit Confirmation (**SesEdit2.exe**)
- 4) Add a Session (**SessAdd.exe**)
- 5) Addition Confirmation (**SessAdd2.exe**)
- 6) Delete Session(s) (**SessDel.exe**)
- 7) Verification of Delete (**SessDel2.exe**)
- 8) Deletion Confirmation (**SessDel3.exe**)

(1) Choose Session (**Sessions.exe**). A table of all the current sessions is displayed (Fig. 4-31). Each session identifier is a hypertext link to **SessEdit.exe** with the session number. Options are available to “Add a Session” and “Delete a Session.”

(2) Edit a Session (**SessEdit.exe**). The current session name is displayed in a form (Fig. 4-32). Changes made in the form are submitted to **SesEdit2.exe**. The option to exit without changing also exists.

(3) Edit Confirmation (**SesEdit2.exe**). This program retrieves the form input from **SessEdit.exe** and posts the change to the **Session.db** table. An HTML confirmation page is then sent with the options to edit more sessions or return to the main admin menu page.

(4) Add a Session (**SessAdd.exe**). Called from **Sessions.exe**, four boxes are available for entering new session names (Fig. 4-33). Any number of the boxes may be used. Submitting this form calls **SessAdd2.exe**.

(5) Addition Confirmation (**SessAdd2.exe**). Form data is retrieved from **SessAdd.exe** and non-empty fields are added to the **Session.db** table. To ensure proper ordering, sessions are placed in alphabetical order. An HTML confirmation page is sent with the options to “Edit more Sessions” or “Return to the main admin page”.

(6) Delete Session(s) (**SessDel.exe**). An HTML table is presented containing all session numbers and names (Fig. 4-34). This presentation includes the session number to make it visually distinct from the “edit session” table. The session number is a hypertext link which calls **SessDel2.exe** with the record number. The option to “Delete all Sessions” is also available in the event the system administrator wishes to create a session list from scratch. Selecting “Delete all Sessions” will set a hidden flag to “DeleteAll” and is passed to **SessDel2.exe**.

(7) Verification of Delete (**SessDel2.exe**). This program retrieves the hidden flag and the session number query string. If “DeleteAll” is not found, the selected session is deleted and a confirmation HTML page sent. Otherwise, an HTML page is sent to confirm that the user wants to delete all of the sessions. Options available through form buttons are: 1) “Yes - DELETE ALL Sessions in the Database” or 2) “Cancel - do NOT delete.”

(8) Deletion Confirmation (**SessDel3.exe**). Appropriate sessions in the **Session.db** table are deleted and an HTML confirmation page sent with the option to “Edit more Sessions”.

f. Edit Keywords

Keywords for identifying topic areas are maintained in the **Keyword.db**

table. Although they remain constant for a given conference, they may change from year to year. Eight executable programs implement this function which enable the user to modify, add, or delete keywords:

- 1) Choose a Keyword (**Keywords.exe**),
- 2) Edit a Keyword (**KeyEdit.exe**),
- 3) Edit Confirmation (**KeyEdit2.exe**),
- 4) Add a Keyword (**KeyAdd.exe**),
- 5) Addition Confirmation (**KeyAdd2.exe**),
- 6) Delete Keyword(s) (**KeyDel.exe**),
- 7) Verification of Delete (**KeyDel2.exe**),
- 8) Deletion Confirmation (**KeyDel3.exe**).

(1) Choose a Keyword (**Keywords.exe**). A table of all the current keywords is displayed (Fig. 4-35). Each Keyword is a hypertext link to **KeyEdit.exe** with the keyword record number. Options are also available to “Add a Keyword” and “Delete a Keyword.”

(2) Edit a Keyword (**KeyEdit.exe**). The selected keyword is displayed in an HTML form (Fig. 4-36). Changes are made in the form and submitted to **KeyEdit2.exe**. There is also option to exit without changing.

(3) Edit Confirmation (**KeyEdit2.exe**). This executable program retrieves the form input from **KeyEdit.exe** and posts the change to the **Keyword.db** table. An HTML confirmation page is sent with the options to “Edit more Keywords” or “Return to the main admin page”.

(4) Add a Keyword (**KeyAdd.exe**). Called from **Keywords.exe**, four HTML form boxes are available for entering new keywords (Fig. 4-37). None of the boxes require entries. Submitting calls **KeyAdd2.exe**.

(5) Addition Confirmation (**KeyAdd2.exe**). Form data is retrieved from **KeyAdd.exe** and non-empty fields are added to the **Keywords.db** table. An HTML confirmation page is sent with the options to “Edit more Keywords” or “Return to the main admin page”.

(6) Delete Keyword(s) (**KeyDel.exe**). A table is presented which contains the keyword record number and name (Fig. 4-38). This presentation includes the record number to make it visually distinct from the “edit keyword” table. The keyword record number is a hypertext link which calls **KeyDel2.exe**. The option to “Delete all Keywords” is also available in the event the System Administrator wishes to create a keyword list from scratch. If “Delete all Keywords” is selected, a hidden flag is set to “DeleteAll” and passed to **KeyDel2.exe**.

(7) Verification of Delete (**KeyDel2.exe**). This program checks the hidden flag and retrieves the keyword record number. If "DeleteAll" is not found, the single keyword is deleted and a confirmation HTML page sent. Otherwise, an HTML page is sent to confirm that the user wants to delete all of the keywords. Options are available through form buttons to "Yes - DELETE ALL Keywords in the Database" or "Cancel - do NOT delete."

(8) Deletion Confirmation (**KeyDel3.exe**). Each keyword in the **Keyword.db** table is deleted and an HTML confirmation page sent with options to "Edit more Keywords" or "Return to the Main Admin Page".

g. Invite Articles

Articles invited by the Session Chairs are not subject to the normal review and selection process. Modifying the **Submissi.db** table to reflect this status is done via this function. Session Chairs will provide the system administrator with a list of invited article titles. The required programs are:

- 1) Display Titles (**Invite.exe**)
- 2) Invite Titles (**Invite2.exe**).

(1) Display Titles (**Invite.exe**). This program displays a form with a list of all the article titles in the **Submissi.db** table with a check box (Fig. 4-39). A drop-

down option box allows the administrator to either “Invite” or “NOT Invite” the selected titles.

To mark articles invited, “Invited” is selected on the drop-down box and the title(s) which are invited are checked. Submitting this form calls **Invite2.exe**. It is not necessary to mark articles as “NOT Invited” unless they were previously marked “Invited” and need to be changed.

(2) Invite Titles(**Invite2.exe**). Called by **Invite.exe**, **Invite2.exe** modifies the “Invited” field on selected articles in the **Submissi.db** table. It first takes the passed query string and parses it to determine the action (Invite or NOT Invite) and the article numbers to be changed. Then, each article is located and changed. A confirmation page is sent with the option to “Invite more Articles”.

h. Accept / Reject Articles

After the review process is complete, articles must be annotated as either “accepted” or “rejected.” This segment allows the system administrator to quickly mark the **Submissi.db** table appropriately. Changes may be made to one article, to several articles, or to all of the articles.

- 1) Select Articles (**Choose.exe**)
- 2) Accept / Reject Articles (**Choose2.exe**)

(1) Select Articles (**Choose.exe**). This program displays a form with a list of all the article titles in the **Submissi.db** table with a check box (Fig. 4-40). A drop-down option box allows the administrator to either “accept,” “accept then reject,” “reject,” or “reject then accept” the selected titles.

The multiple choices provided will simplify this process because the Administrator will not have to specifically mark every article. When a list exists with more rejections than acceptances, checking off accepted articles and the “accept then reject” option will set the selected titles as accepted and automatically set the rest as rejected. If there are fewer rejected articles than accepted, then the rejected article titles are selected and “reject then accept” will reject those and accept the rest.

To mark articles individually, either the “accept” or “reject” option is selected on the drop-down box and the title(s) checked off are modified as appropriate. “Accept” or “reject” will only modify the selected articles and will not affect other articles. Submitting this form calls **Choose2.exe**.

(2) Accept / Reject Articles(**Choose2.exe**). The query string is retrieved and parsed to obtain the action and the titles. A case statement uses the action passed to modify the necessary article fields in the SUBMISSION Table. Titles are placed on a string list and individually processes. When the action is to “accept then reject” or “reject then accept,” then all articles are set to the second action and then the string list is processed to change the selected articles.

i. Assign Articles to Sessions

Once articles are accepted, they must be assigned to a session and ordered within a session. The following programs assign articles and order them within a session.

- 1) Assign Articles to Session (**Assign.exe**)
- 2) Order Articles in Session (**Assign2.exe**)
- 3) Confirmation (**Assign3.exe**)

(1) Assign Articles (**Assign.exe**). This program displays a form with a list of all the article titles in the **Submissi.db** table with a check box (Fig. 4-41). A drop-down option box allows the administrator to select the session name. The session is selected and then each of the titles which belong in that session are checked. Submitting the form calls **Assign2.exe**.

(2) Order Articles (**Assign2.exe**). The query string is retrieved and parsed to obtain the session record number and the list of titles. Article titles assigned to the selected session are displayed next to a form box which is used to assign an order to each article. Each box has a default number inserted when it is displayed. Once the numbers are changed to reflect the proper order of presentation, this form is submitted to **Assign3.exe**.

(3) Confirmation (**Assign3.exe**). This program retrieves the query string and parses it to obtain the session number and a list of titles and a list of title orders. The string list of titles and the string of title orders are matched so that the first title goes with the first order. Title record numbers are used to locate a title's record. The session and order are then recorded. An HTML confirmation page is sent which lists the titles and their order.

j. Assign Presentation Times to Articles

Prior to printing the Abstract Catalog, presentation times must be included in the article information. Three programs are used to assign presentation times to each article.

- 1) Choose Session (**Times.exe**),
- 2) Assign Presentation Times (**Times2.exe**),
- 3) Confirmation (**Times3.exe**).

(1) Choose Session (**Times.exe**). An HTML table of all the current sessions is displayed (Fig. 4-42). Each session identifier is a hypertext link to **Times2.exe** with the session record number.

(2) Assign Times (**Times2.exe**). Each article title assigned to the requested session is retrieved and presented with its order number and an HTML form

box to enter the presentation time (Fig. 4-43). After the times are entered, the form is submitted to **Times3.exe**.

(3) Confirmation (**Times3.exe**). The query string is retrieved and parsed to obtain a string list of titles and times. The list is processed such that the title number is used to locate the record and then the time is placed in the database. An HTML confirmation page is sent which again displays the titles with their order and time.

*k. **Modify Passwords***

System administrator and reviewer identities and passwords are maintained with this segment. System administrators and reviewers are referred to as “users.” Users can be modified, added, and deleted. Note that only system administrators have access to the Administration sub-system. Reviewers will only have access to the Review sub-system.

- 1) Selection options (**PassWds.exe**),
- 2) Modify Password (**PwdMod.exe**),
- 3) Add new user (**PwdAdd.exe**),
- 4) Addition confirmation (**PwdAdd2.exe**),
- 5) Delete a user (**PwdDel.exe**),
- 6) Deletion confirmation (**PwdDel2.exe**).

(1) Selection options (**PassWds.exe**). This program sends an HTML instruction/option page which is divided into sections for System Administrators and Reviewers (Fig. 4-44). A form is displayed which allows system administrators to modify passwords. New passwords must be entered twice for verification. Submitting this form calls **PwdMod.exe**. Additional options are to add (**PwdAdd.exe**) and delete (**PwdDel.exe**) users.

(2) Modify Password (**PwdMod.exe**). This program retrieves the form data, including an identifier for whether the ID is for a system administrator or a reviewer. It then retrieves the old password from the password table (**Admin.db**) and authenticates the request. If the old password is valid, and the two new passwords match, then the old password is replaced with the new. An HTML confirmation page is sent with the option to "Modify more Passwords".

(3) Add new user (**PwdAdd.exe**). This program creates an HTML page with a form to enter a new system administrator or reviewer ID and password (Fig. 4-45). The password must be entered twice to be valid. Submitting the form calls **PwdAdd2.exe**.

(4) Addition confirmation (**PwdAdd2.exe**). Creates an HTML page which confirms the new ID and offers the option to "Modify more Passwords".

(5) Delete a user/reviewer (**PwdDel.exe**). An HTML page is created with a form for entering the system administrator or reviewer ID and password to be deleted (Fig. 4-46). Submitting the form calls **PwdDel2.exe**.

(6) Deletion confirmation (**PwdDel2.exe**). This program creates an HTML page which confirms the deletion and offers the option to “Modify more Passwords”.

l. Print out a Session list

This single program (**P_Sess.exe**) displays a list of the current sessions on the screen (Fig.4-47). The display is suitable for viewing or printing out using the Web browser’s print screen function.

m. Print out Article data

This function prints out information about an article. It is intended for use with the current review process. This information will be printed, along with the Extended Summary, and given to the reviewer. Options allow information about a single article to be displayed on the screen or printed to a file.

- 1) Select Article or All (**P_Artl.exe**),
- 2) Print Article to screen (**P_Artl2.exe**),

3) Print Articles to a file (**P_Artfil.exe**).

(1) Select Article or All (**P_Artl.exe**). An HTML table is displayed which contains every article title in the **Submissi.db** table as a hypertext link to **P_Artl2.exe** (Fig. 4-48). There is also a hypertext link which provides the option to print article information for every article to a file.

(2) Print Article (**P_Artl2.exe**). This program first looks for an article record number which would be passed if the user selected an article title hypertext link. If no record number is passed, then the user selected the “Print to a file” hypertext link. In this case, an HTML page is created by **P_Artl2.exe** which allows the user to enter a name for file in which all article information will be stored. Submitting that form calls **P_Artfil.exe**. If **P_Artl2.exe** is passed an article record number, then the record is located and retrieved. Information is then displayed in HTML format suitable for printing with the Web browser’s print screen function (Fig. 4-49).

(3) Print Articles to file (**P_Artfil.exe**). This program retrieves all articles from the **Submissi.db** table. The necessary information is then printed to a file with the user given name. The file is always given an extension of **.art** and is placed in the C:\My Files\Thesis directory available for FTP download by the system administrator.

n. Print out Accept / Reject letter

After review, articles are marked as either accepted or rejected. This function generates the accept/reject letters to be mailed to submitters. Letters can be generated for all the articles at once (to a file only), or for individual letters (to the screen or to a file).

- 1) Instructions and options (**P_Accept.exe**),
- 2) Print all to a file (**P_AccAll.exe**),
- 3) Select Article (**P_Accep2.exe**),
- 4) Choose screen or file (**P_Accep3.exe**),
- 5) Print to screen (**P_AccScr.exe**),
- 6) Print to file (**P_AccFil.exe**).

(1) Instructions / Options (**P_Accept.exe**). An HTML page is created which describes the three options available and provides links to each (Fig. 4-50).

(2) Print all to a file (**P_AccAll.exe**). This program generates an accept or reject letter, as designated by the “accept” field in the **Submissi.db** table, for every article in the database. The user has the opportunity to provide a name for the output file. The file is always given an extension of **.let** and is placed in the **C:\Website\Download** directory for FTP download by the system administrator.

(3) Select Article (**P_Accep2.exe**). This program is called when the user wishes to print a single letter. An HTML table is generated with every title as a hypertext

link to **P_Accep3.exe**.

(4) Choose screen or file (**P_Accep3.exe**). The option is given to print a letter to either the screen or to a file. The article record number is passed to the appropriate program (**P_AccScr.exe** or **P_AccFile.exe**)

(5) Print to screen (**P_AccScr.exe**). The letter is generated to the screen in an HTML page. The page is suitable for printing by the user's Web browser.

(6) Print to file (**P_AccFile.exe**). The letter is generated to a file which the user names. The file is always given an extension of **.let** and is placed in the **C:\Website\Download** directory for FTP download by the user.

*o. **Print Abstract Catalog***

This function organizes the articles, in order by session, and prints them in the Abstract Catalog format to a file.

- 1) Print Initiation (**P_AbsCat.exe**),
- 2) Print Confirmation (**Prabsca3.exe**).

(1) Print Initiation (**P_AbsCat.exe**). Utilizing the **Sessions.db**, **Submissions.db**, and **Author.db** tables, this function sorts the articles by session and

prints the appropriate data to a file in the format of the Abstract Catalog. The user has the opportunity to provide a name for the output file (Fig. 4-51). The file is always given an extension of **.cat** and is placed in the C:\Website\Download\ directory for FTP download by the user.

(2) Print Confirmation (**Prabsca3.exe**). Confirming the successful completion of the print job, the name of the file is again displayed and a link is provided which automatically downloads the file to the system administrator's host.

4. Applications

Three applications are required to operate the WWW Conference Information System, O'Reilly & Associates' WEBSITE web server, FTPD FTP Server, and Borland's PARADOX Relational Database. Following is a brief discussion of the specific implementations of these programs.

a. O'Reilly & Associates' WEBSITE Web Server

The WEBSITE [Ref. 6] software is responsible for maintaining the connection between the PC hardware and the Internet. The PC has been assigned a static IP address (131.120.20.70) and maintains a continuous Internet connection via the ECE department's local area network. When an Internet user accesses the PC's address,

WEBSITE automatically displays **Index.html** in the C:\Website\Htdocs directory.

WEBSITE then displays other HTML pages or executes Delphi programs as implemented within the system.

All HTML pages (other than **Index.html**) are stored in the C:\Website\Htdocs\Prototype directory. All Delphi executables are stored in the C:\Website\cgi-win directory as required by WEBSITE. The programming code for each executable is stored in the C:\Delphi\Bin directory. The queries conducted throughout the system are initiated by WEBSITE passing standard query language (SQL) to Borland's Paradox which maintains the system database and is discussed in part c.

WEBSITE is also responsible for limiting a user's access. By default, WEBSITE limits a user's access to no lower than the C:\Website\Htdocs directory. This protects the root directory from unauthorized access. WEBSITE also handles FTP downloads for users. Any file that has the extension ".exe" or ".html" and resides in a sub-directory of C:\Website\Htdocs can be downloaded by any Internet user. This is an important consideration when placing files on the hard drive. File uploads (i.e. loading files to the PC) are handled strictly by the FTPD FTP server discussed in the next section.

b. FTPD FTP Server

The FTPD FTP Server is shareware which handles file uploads to the system [Ref. 7]. It was installed to allow users to upload Extended Summaries which contain graphics and equations not suitable for ASCII text that HTML forms are limited

to. Allowing uploads can be risky because anything can be uploaded. System administrators are warned not to execute any executable (.exe) programs found in the upload directory. Any such program found should be deleted immediately. The only legitimate uploads are post script (.ps) Extended Summaries.

Security of uploaded files is maintained by limiting users to one directory and disabling the ability to list directory contents. Thus a user may upload his/her own file, but will not be able to determine the contents of the upload directory. FTPD has not been registered. All attempts to contact the company have failed. Thus, no technical support is available. This is an area for future improvement. More powerful and flexible FTP servers are commercially available.

c. Borland's PARADOX Relational Database

Implementation of the PARADOX database for this system mirrored the one in use by conference administrators. Seven tables were constructed to maintain data on the Contact Authors, Submissions, Sessions, Keywords, and system passwords. The tables were initially designed using SALSA [Ref. 10], an object-oriented database design tool. SALSA generated reports and a brief description of the SALSA program are included in Appendix C.

A relational database stores not only data, but also maintains relationships between different tables. As mentioned above, the database we implemented contains seven separate tables.

- 1) **Author.db** contains personal information on each contact author.
- 2) **Submissi.db** contains information about an author's paper.
- 3) **TempAuth.db** temporarily stores changes to an author's personal information until it is verified by conference personnel.
- 4) **Session.db** stores session mnemonics that indicate when a particular session will occur.
- 5) **Keyword.db** maintains a list of keywords authors may use to classify their papers.
- 6) **Admin.db** stores user ID's and passwords for system administrators.
- 7) **Review.db** stores user ID's and passwords for article reviewers.

By using separate tables to store this information, an author can submit more than one paper without causing problems for the database. Each contact author is given a unique identifier that *relates* him/her to each of his/her submissions. Similarly, an author's submissions are *related* to keywords, and a session number. Thus queries can be performed on either an author or on a submission and all information from any table can be retrieved. The names and contents of each database table is listed in Appendix C.

5. System Web Pages

The following pages are screen captures of the web pages created by the Asilomar Conference System. The pages were viewed with Netscape Navigator Gold 3.0 [Ref. 8]

and captured with Corel Draw Capture.

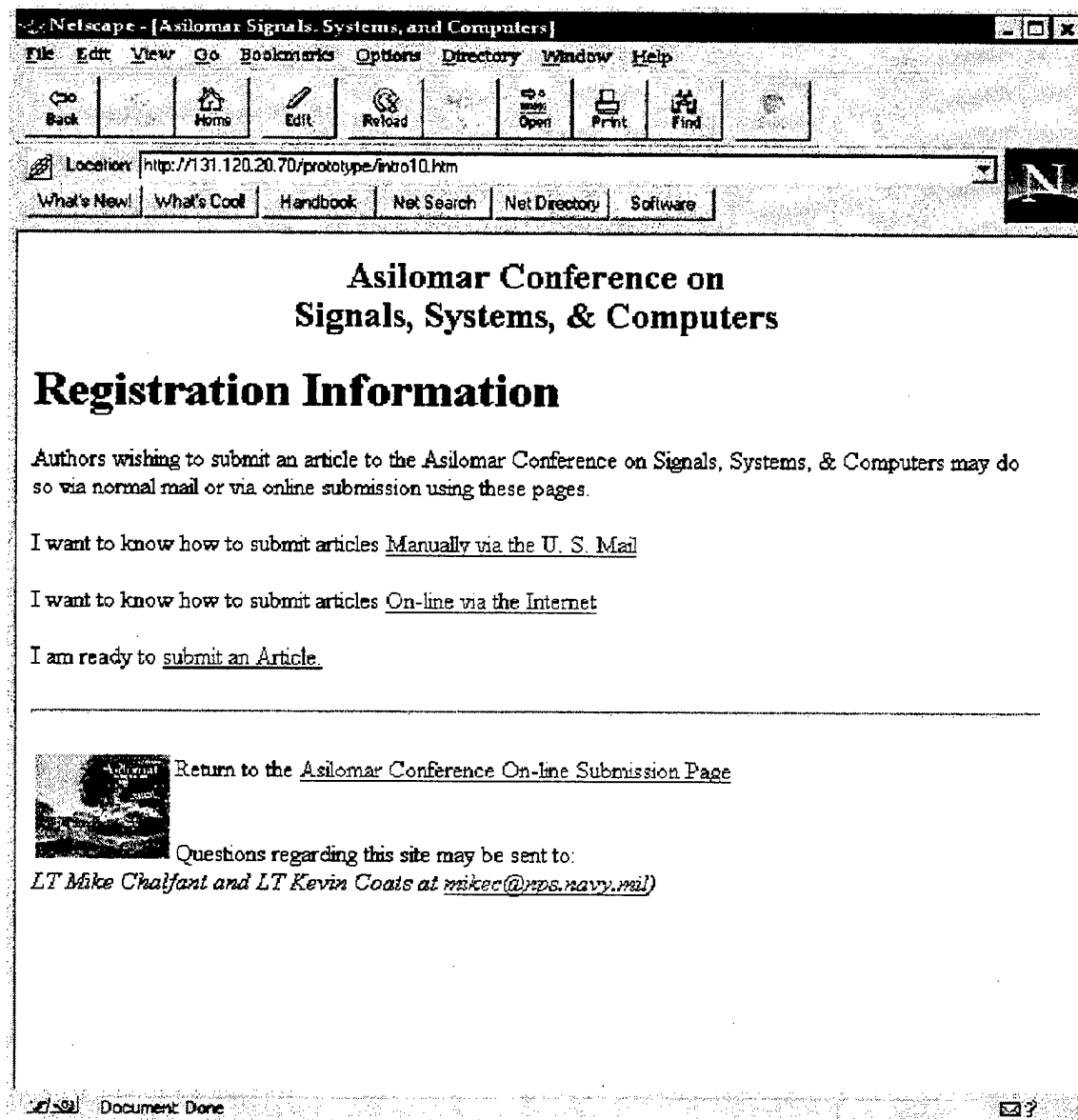


Figure 4-4. Intro10.htm.

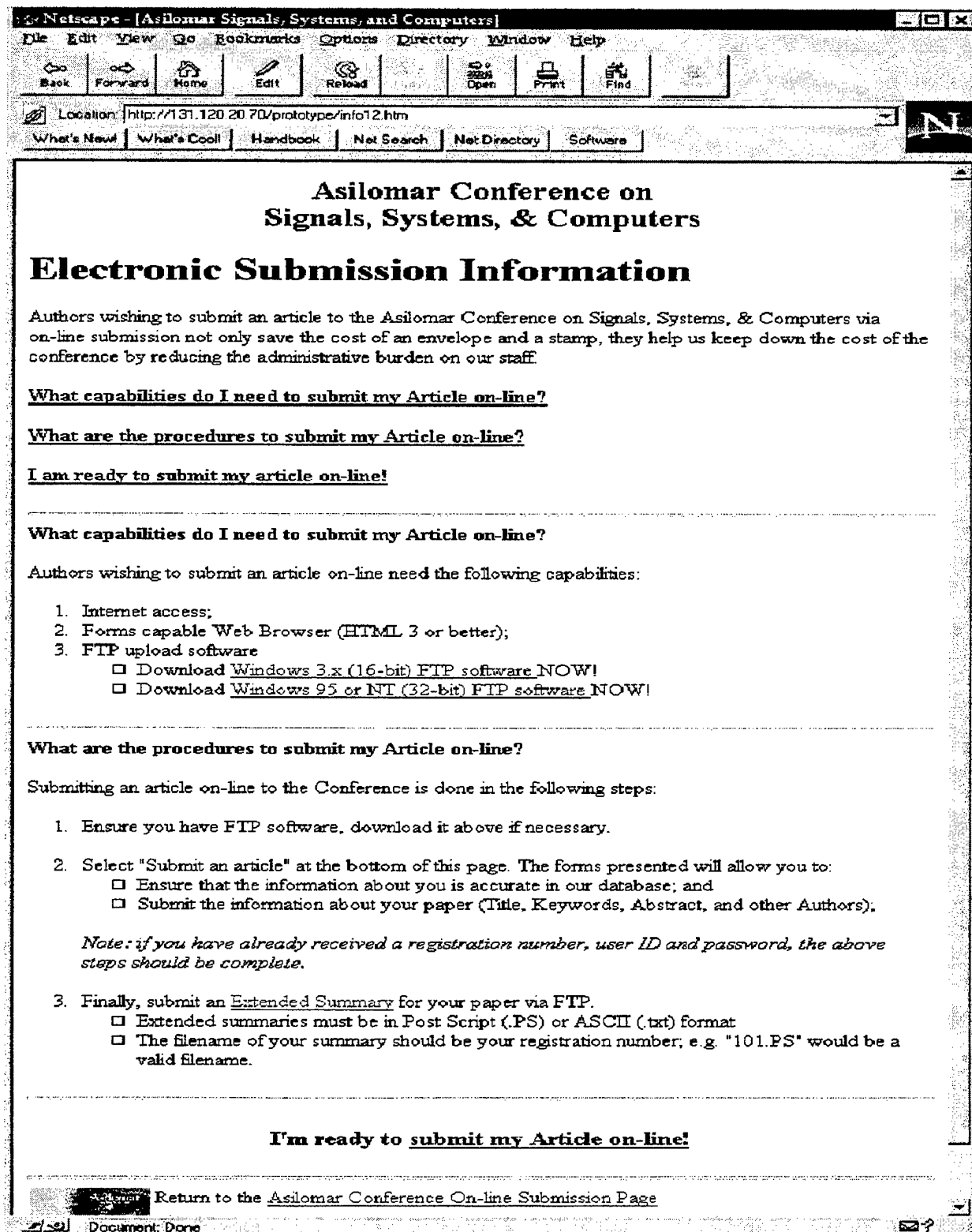


Figure 4-5. Info12.htm.

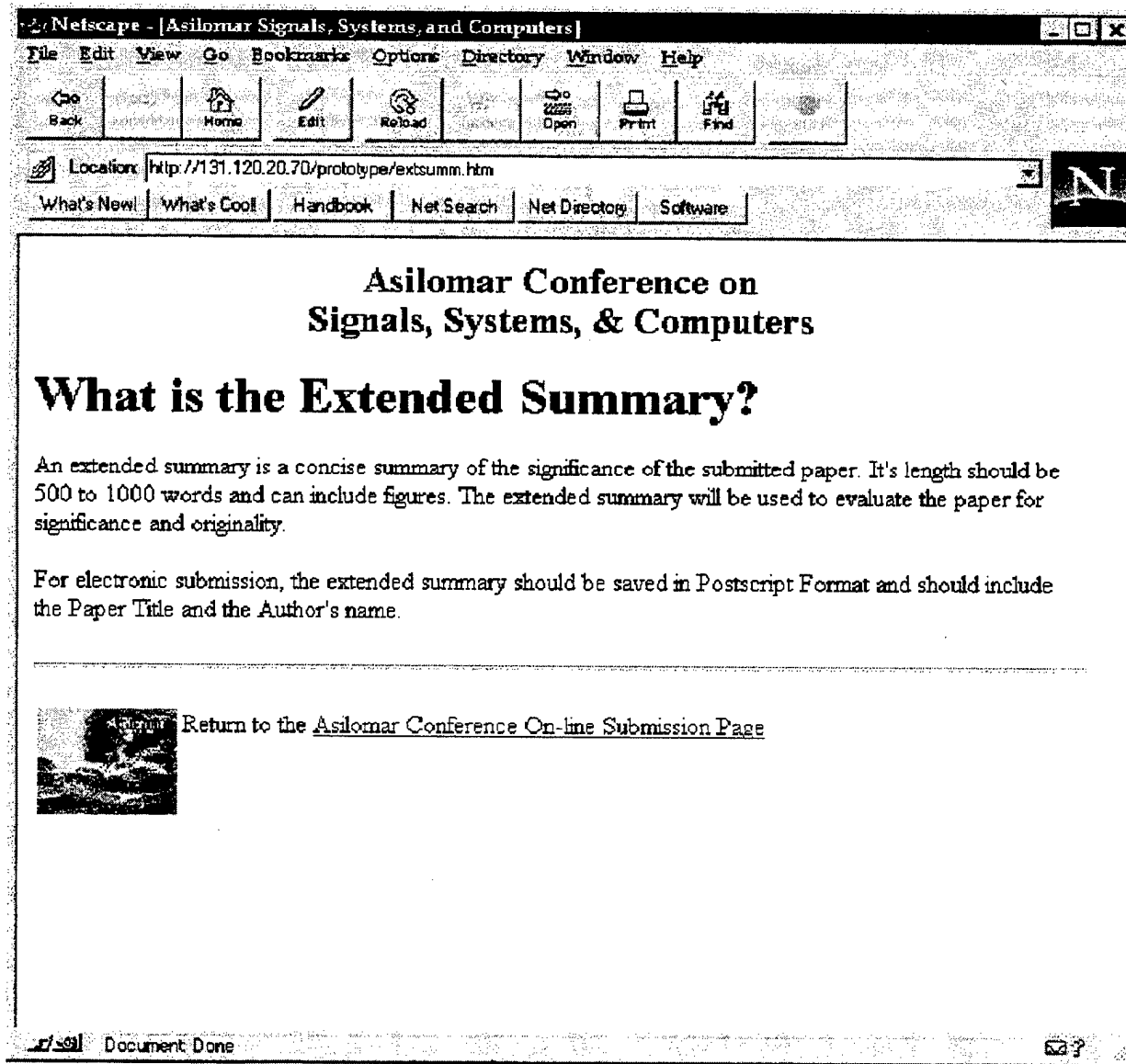


Figure 4-6. Extsumm.htm.

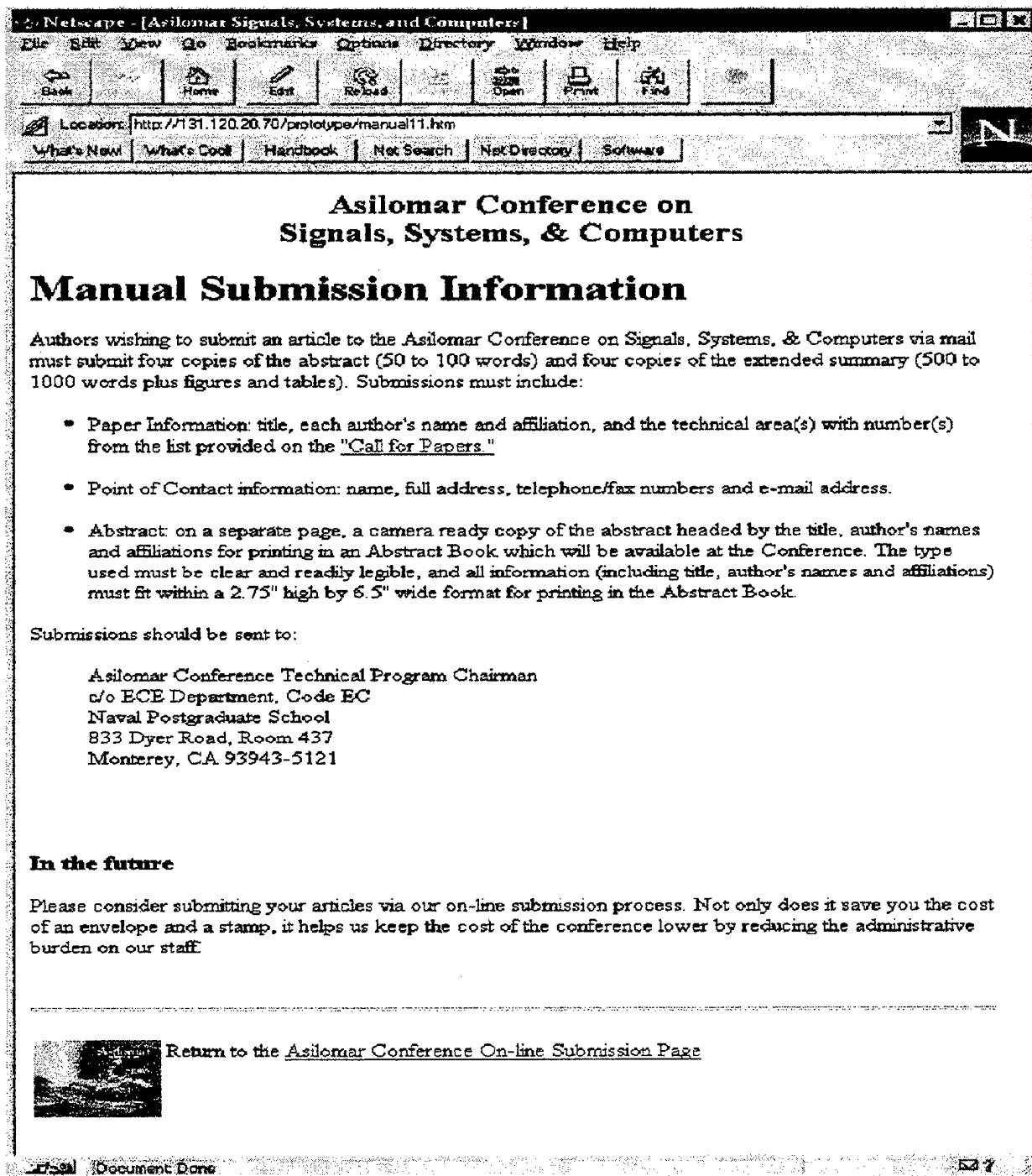


Figure 4-7. Manual11.htm.

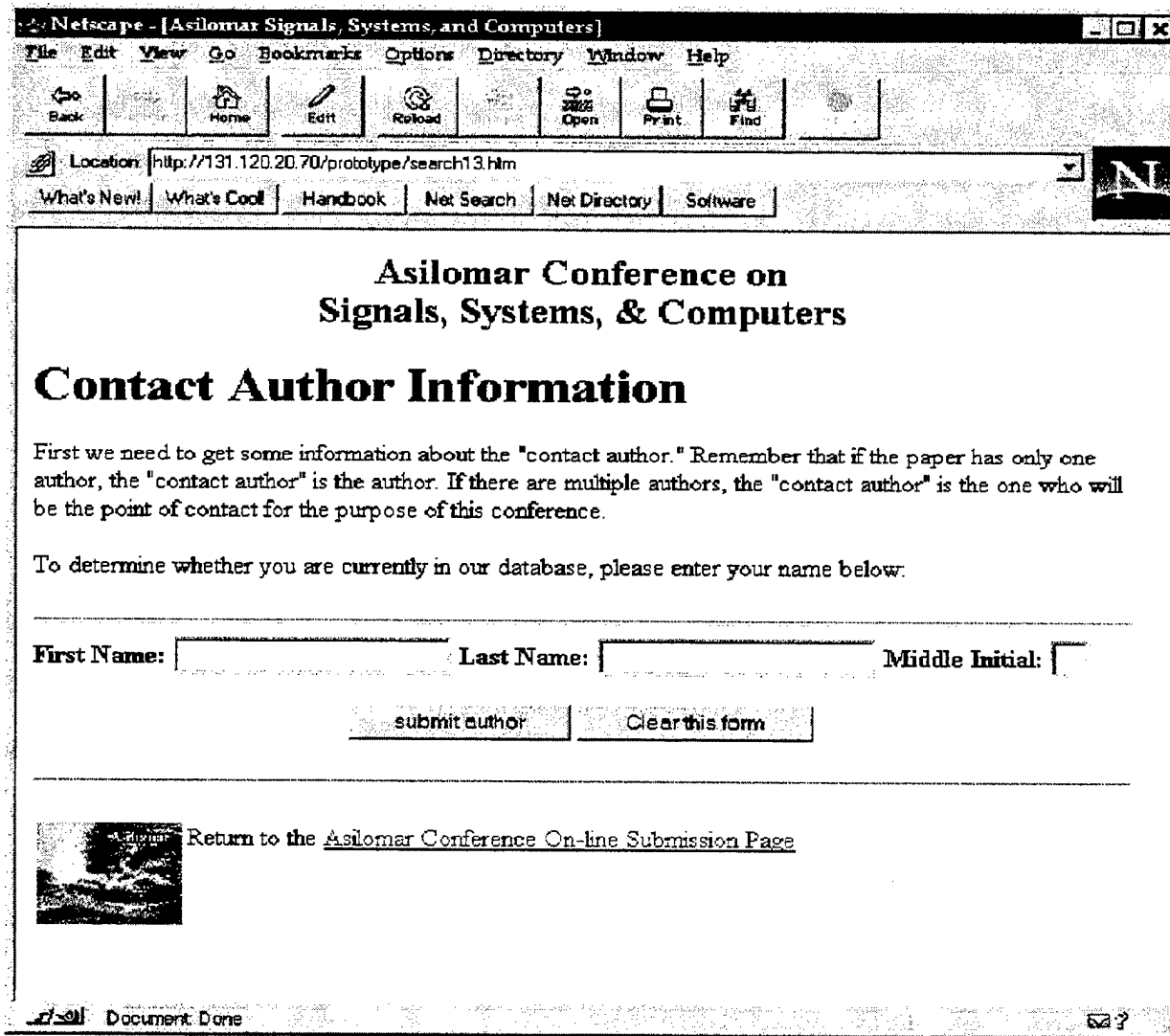


Figure 4-8. Search13.htm.

Netscape - [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Open Print Find

Location: http://131.120.20.70/cgi-win/D13.exe

What's New! What's Cool Handbook Net Search Net Directory Software

Add New Author Information

Your name was not found in our current database. Please provide the following information so we will be able to contact you.

First Name: Last Name: Middle Initial:

Honorific Title:

Institution:

Department:

Mail Stop:

Street address:

City: State:

Zip Code: Country:

Telephone number (with Area/Country codes):

Fax number (with Area/Country codes):

E-mail:

 [Return to the Asilomar Conference On-line Submission Page](#)

Document Done

Figure 4-9. D13.exe: No Author Found in Database.

Netscape - [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Forward Home Edit Reload Stop Open Print Find

Location: http://131.120.20.70/cgi-win/D13.exe

What's New! What's Cool! Handbook Net Search Net Directory Software

Edit Author Information

We have located your record in the database. Please ensure that your record is up-to-date by reviewing the information below and making any changes that are necessary. For your protection, changes are not made a part of the permanent record until verified by our staff.

First Name: Last Name: Middle Initial:

Honorific Title:

Institution:

Department:

Mail Stop:

Street address:

City: State:

Zip Code: Country:

Telephone number (with Area/Country codes):

Fax number (with Area/Country codes):

E-mail:

The database has found a unique entry based on the first and last name you submitted. If you believe this entry refers to another person, then you have two choices:

- If you might have used another name, then [Go back and re-submit your query.](#) or
- If this is your name but this is not your record, then [Add your name to our records.](#)

If this entry does refer to you, please review the information and update it as necessary, then click on "Submit Information" below

NO changes necessary-ALL data is correct

Netscape

Figure 4-10. D13.exe: One Author Found.

NetScape - [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Forward Home Edit Reload Open Print Find

Location: http://131.120.20.70/cgi-win/auth132.exe

What's New! What's Cool! Handbook Net Search Net Directory Software

Asilomar Conference on Signals, Systems, & Computers

Add New Author Information

Your name was not found in our current database. Please provide the following information so we will be able to contact you.

First Name: Last Name: Middle Initial:

Honorific Title:

Institution:

Department:

Mail Stop:

Street address: City:

State:

Zip Code: Country:

Telephone number (with Area/Country codes):

Fax number (with Area/Country codes):

E-mail:

 [Return to the Asilomar Conference On-line Submission Page](#)

Document Done

Figure 4-11. D13.exe: Submitting Author's Name Matches Another in the Database.

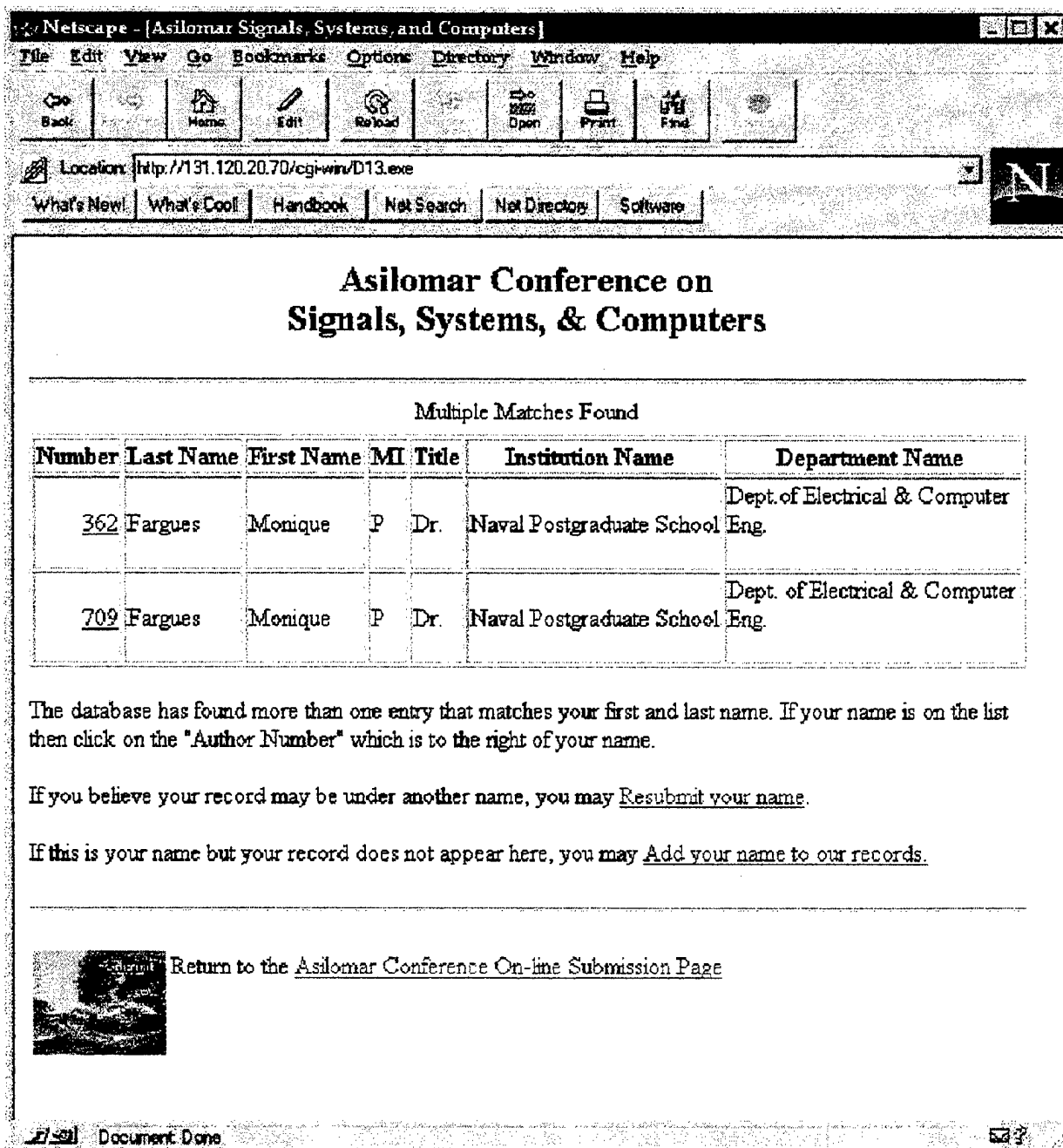


Figure 4-12. D13.exe: Multiple Matches Found in Database.

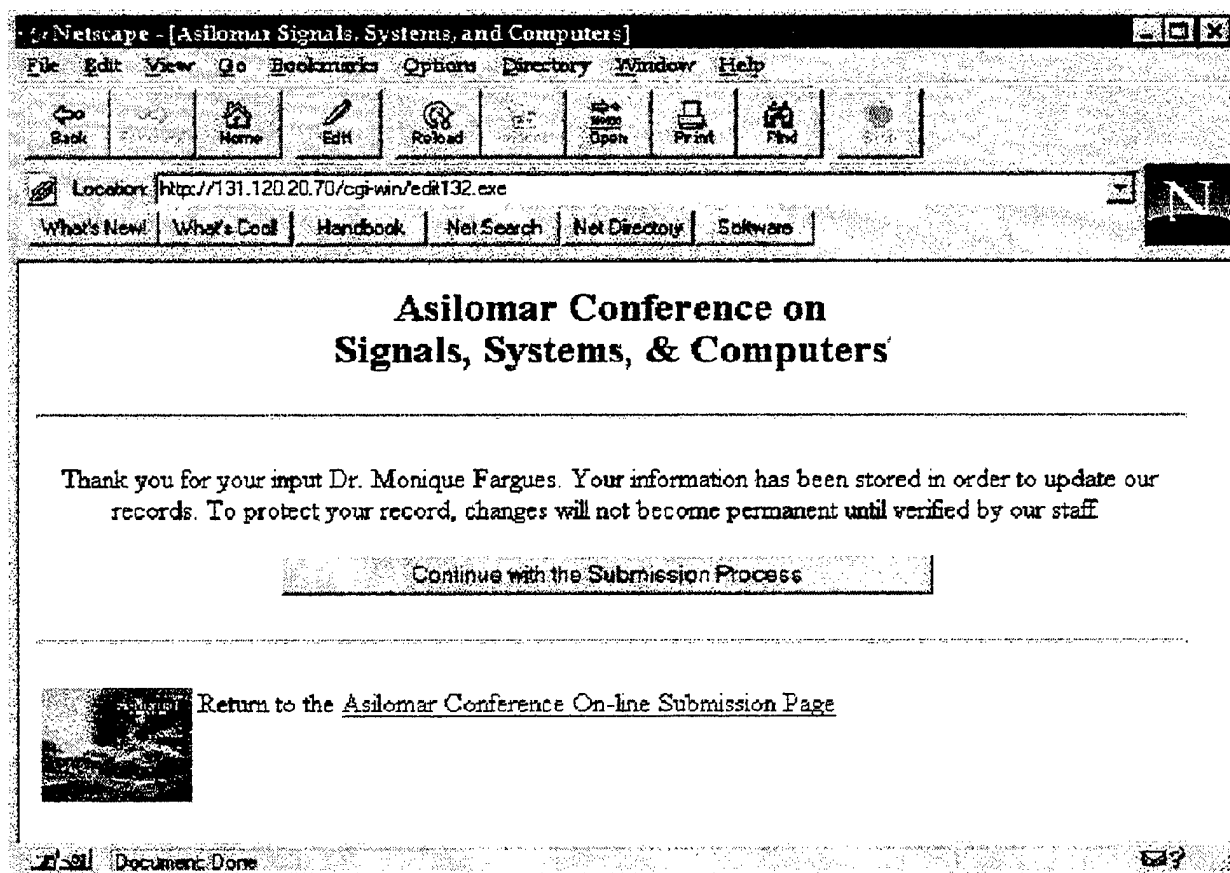


Figure 4-13. Information Page Generated by Add132.exe or Edit132.exe.

[- Netscape - [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Open Print Find

Location: http://131.120.20.70/cgi-win/d_submit.exe

What's New What's Cool Handbook Net Search Net Directory Software

Asilomar Conference on Signals, Systems, & Computers

New Article Submission

Please provide the following information about the article.

Title:

Keyword #1:

Keyword #2:

Keyword #3:

Abstract:

Please enter your abstract here by either typing or copying over this text.

Remember to limit the length of your abstract to 100 words. HTML will allow you to enter more than that in this box, BUT only the first 100 words will be retained.

If typing your next, you may want to use carriage returns to keep the text within the viewing area of this box. If cutting and pasting, the display in this box does not matter.

Document Done

Figure 4-14. D_Submit.exe Part 1.

Netscape - [Asilomar: Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Open Print Find

Location: http://131.120.20.70/cgi-win/d_submit.exe

What's New What's Cool Handbook Net Search Net Directory Software

the text within the viewing area of this box. If cutting and pasting, the display in this box does not matter.

Other Author Information

If there are multiple authors for this article, please enter the following information. In the boxes marked "Name Position" please enter the position in which you wish each author's name to appear when the abstract is printed.

Position of the Contact Author's Name:

Other Authors

Name Position: 2	First Name:	Middle Initial:
Last Name:		
Institution:		
Name Position: 3	First Name:	Middle Initial:
Last Name:		
Institution:		
Name Position: 4	First Name:	Middle Initial:
Last Name:		
Institution:		
Name Position: 5	First Name:	Middle Initial:
Last Name:		
Institution:		
Name Position: 6	First Name:	Middle Initial:
Last Name:		
Institution:		

 [Return to the Asilomar Conference Home Page](#)

Document Done

Figure 4-15. D_Submit.exe Part 2.

1 Netscape - [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Forward Home Edit Reload Open Print Find

Location: http://131.120.20.70/cgi-win/d_paper.exe

What's New! What's Cool Handbook Net Search Net Directory Software

Asilomar Conference on Signals, Systems, & Computers

Thank you for your input Dr. Monique Fargues. Please confirm that the following information is correct before it is added to our records. **This information can not be changed after it is submitted this time!**

Title:

Thesis example

Keyword #1: Image & Multidimensional signal processing

Keyword #2: Speech & Communications

Keyword #3: Higher-order Statistics

Abstract:

Please enter your abstract here by either typing or copying over this text.

Remember to limit the length of your abstract to 100 words. HTML will allow you to enter more than that in this box, BUT only the first 100 words will be retained.

If typing your next, you may want to use carriage returns to keep the text within the viewing area of this box. If cutting and pasting, the display in this box does not matter.

Document Done

Figure 4-16. Dpaper.exe Part 1.

Netscape - [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Go Back Home Edit Reload Stop Open Print Find

Location: http://131.120.20.70/cgi-win/d_paper.exe

What's New What's Cool Handbook Net Search Net Directory Software

Other Author Information

If there are multiple authors for this article, please enter the following information. It is assumed that you are the "Contact Author" for this paper. In the boxes marked "Name Position" please enter the position in which you wish each authors name to appear when the abstracts are printed.

Position of YOUR Name:

Name Position: First Name: Middle Initial:

Last Name:

Institution:

Name Position: First Name: Middle Initial:

Last Name:

Institution:

Name Position: First Name: Middle Initial:

Last Name:

Institution:

Name Position: First Name: Middle Initial:

Last Name:

Institution:

Name Position: First Name: Middle Initial:

Last Name:

Institution:

 [Return to the Asilomar Conference On-line Submission Page](#)

Document Done

Figure 4-17. Dpaper.exe Part 2.

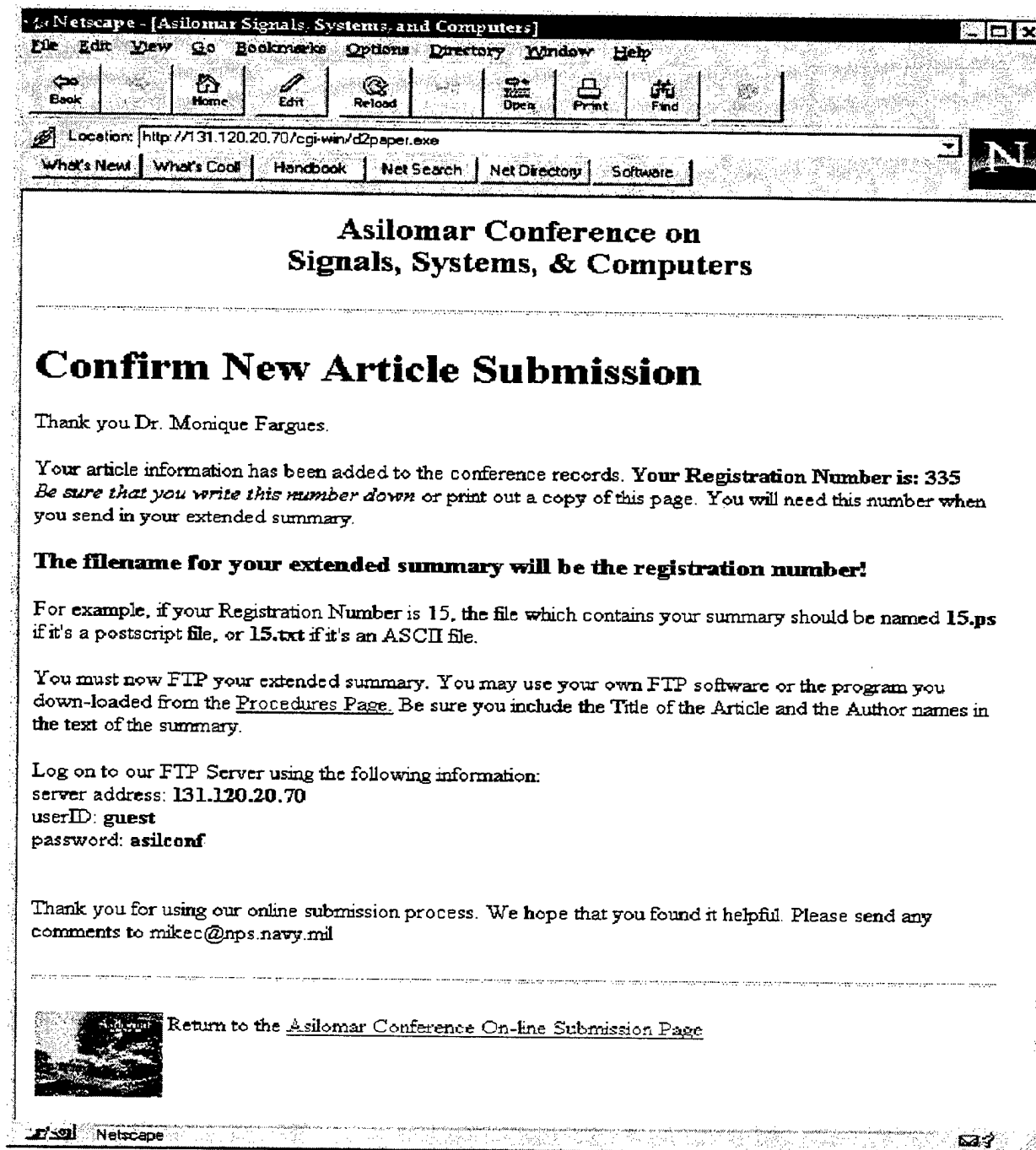


Figure 4-18. D2paper.exe.

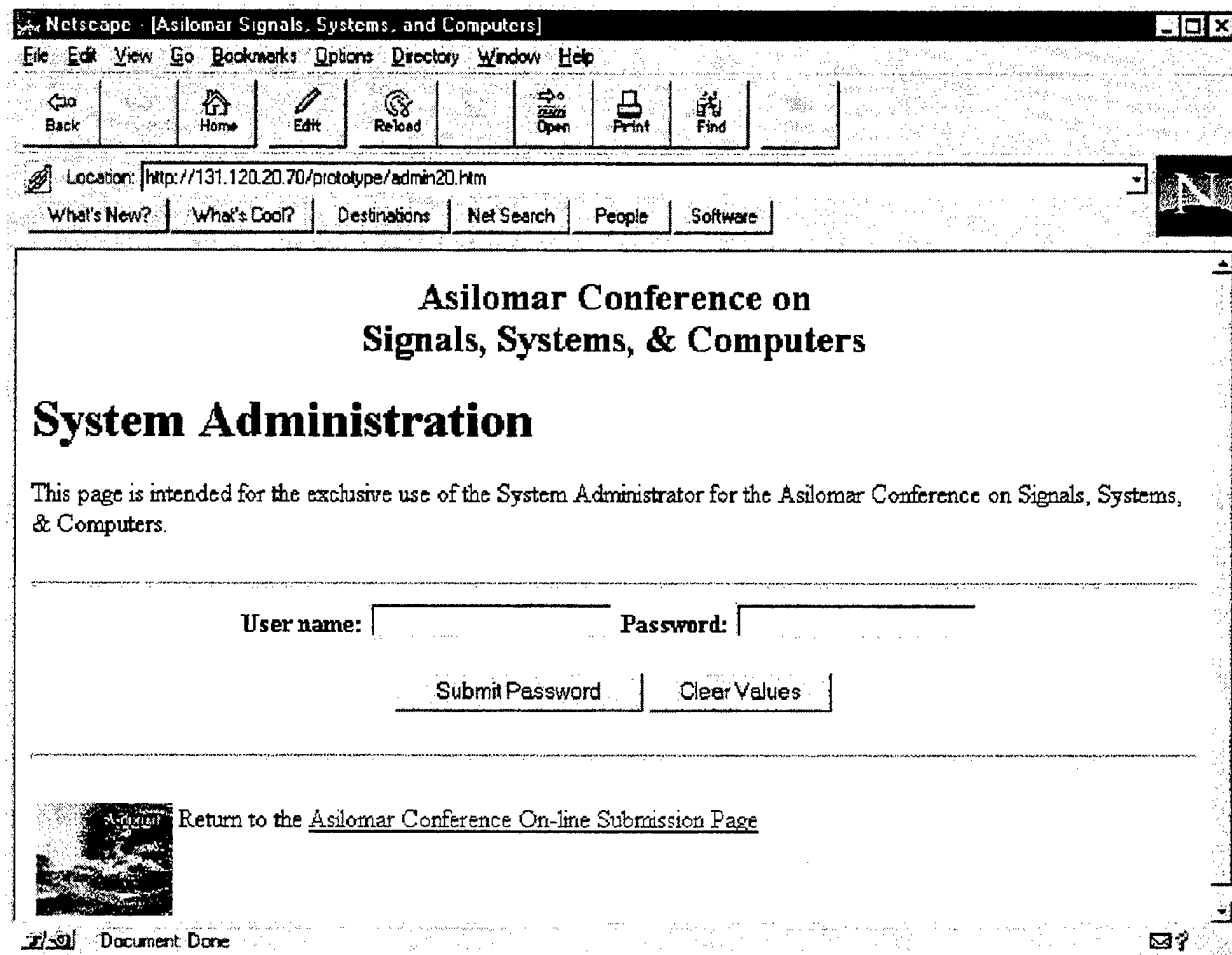


Figure 4-19. Admin20.htm.

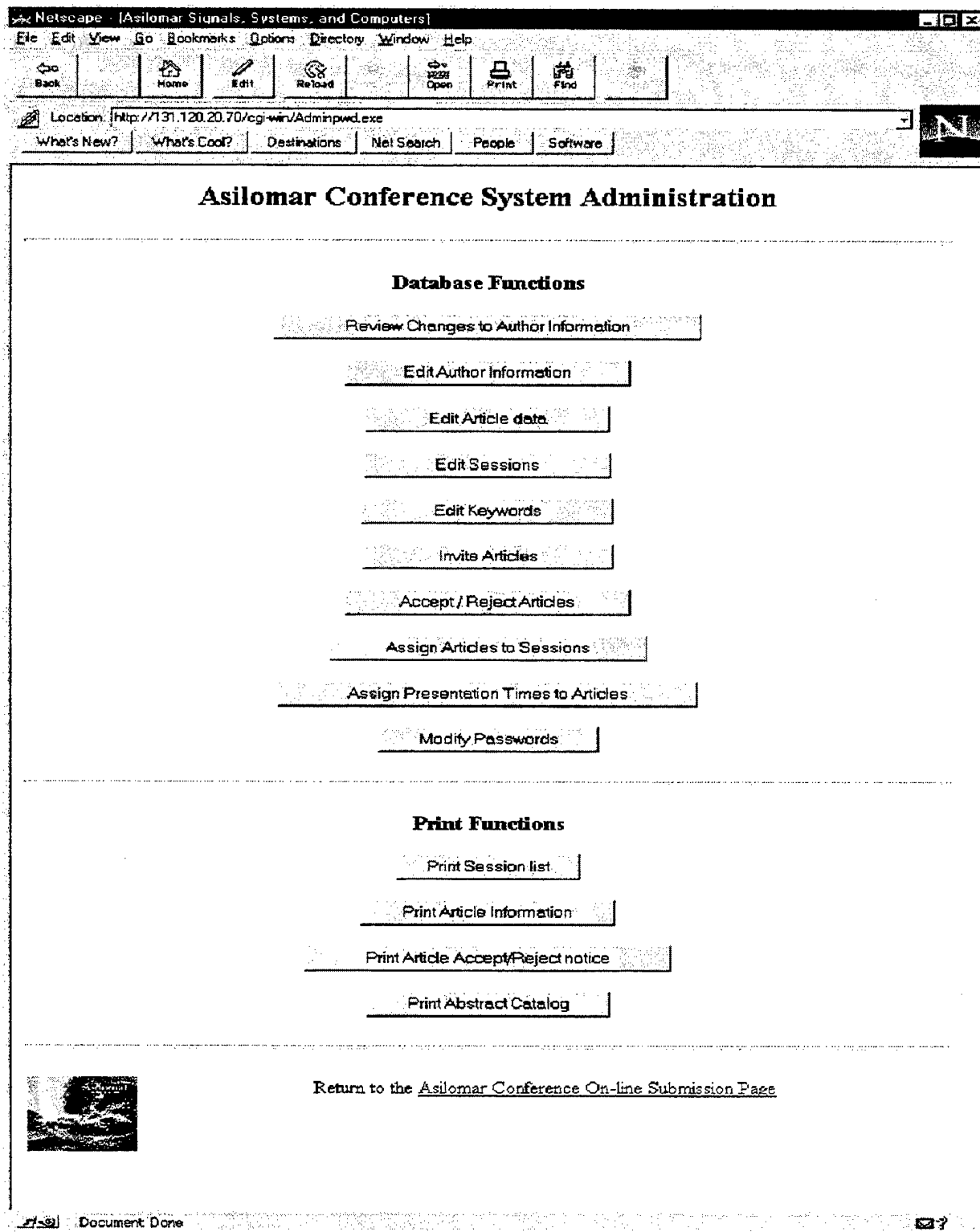


Figure 4-20. Aminpwd.exe.

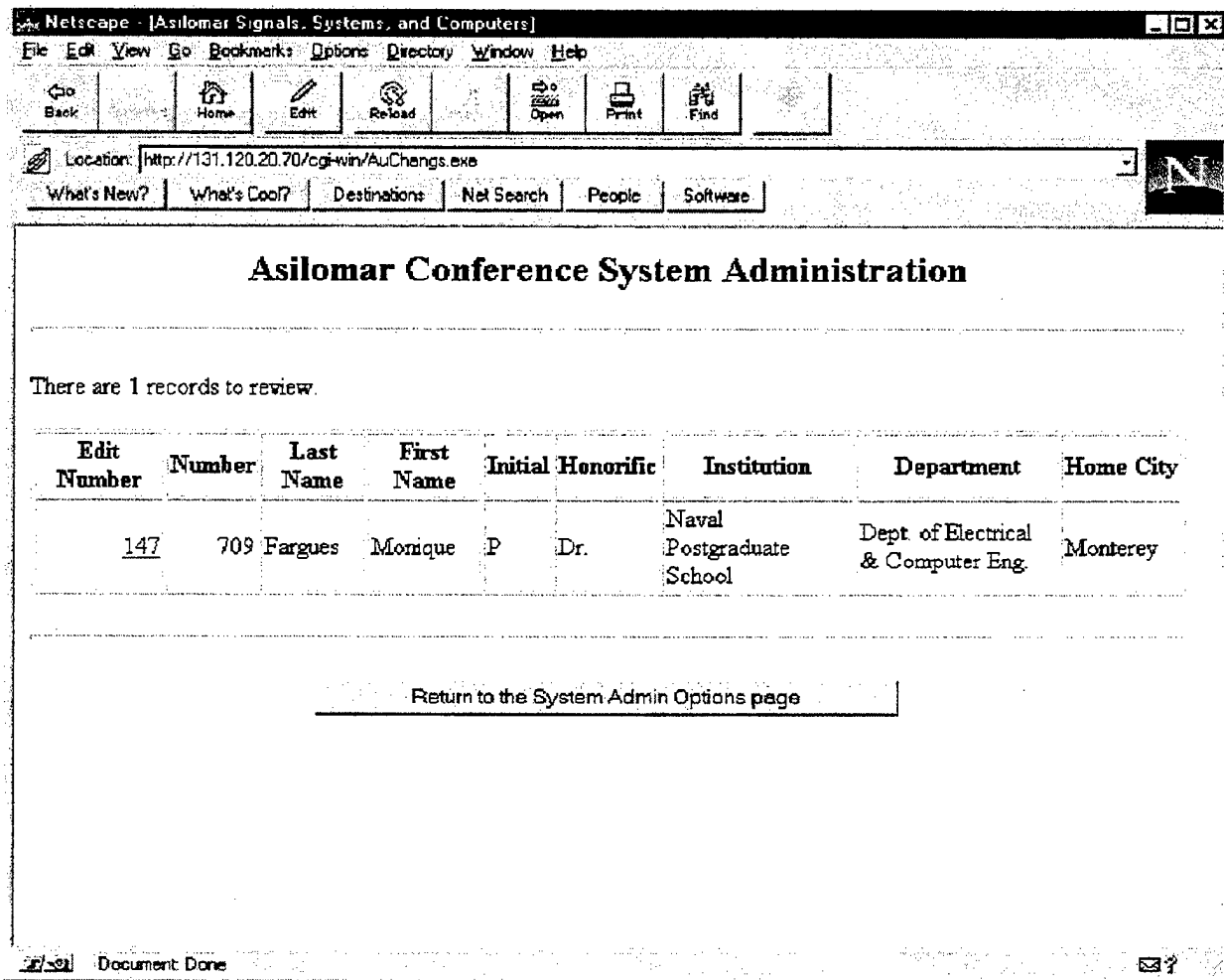


Figure 4-21. Auchangs.exe.

Netcape [Asimov's Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Open Print Find

Location: <http://131.120.20.70/cgi-win/ViewChng.exe?147>

What's New? What's Cool? Destinations Net Search People Software

Compare Author Information

Here is the old record along with the submitted changes.

OLD First Name:

NEW First Name:

OLD Last Name:

NEW Last Name:

OLD Middle Initial:

NEW Middle Initial:

OLD Honorable Title:

NEW Honorable Title:

OLD Institution:

NEW Institution:

OLD Department:

NEW Department:

OLD Mail Stop:

NEW Mail Stop:

OLD Street address:

NEW Street address:

OLD City:

NEW City:

Document Done

Figure 4-22. Viewchng.exe Part 1.

NetScape - [Asdromat Signals Systems and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Stop Open Print Find

Location: <http://131.120.20.70/cgi-bin/ViewChng.exe?1147>

What's New? What's Cool? Destinations Net Search People Software

OLD Street address:
NEW Street address:

OLD City:
NEW City:

OLD State:
NEW State:

OLD Zip Code:
NEW Zip Code:

OLD Country:
NEW Country:

OLD Telephone number (with Area/Country codes):
NEW Telephone number (with Area/Country codes):

OLD Fax number (with Area/Country codes):
NEW Fax number (with Area/Country codes):

OLD E-mail:
NEW E-mail:

Document Done

Figure 4-23. Viewchng.exe Part 2.

Netscape - [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Open Print Find

Location: http://131.120.20.70/cgi-win/inputau.exe

What's New? What's Cool? Destinations Net Search People Software

Asilomar Conference System Administration

Edit Author Information

To locate an author name in the database, please enter the name below:

First Name: Last Name: Middle Initial:

Document Done

Figure 4-24. Inputau.exe.

Netscape [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Open Print Find

Location: 

What's New? What's Cool? Destinations Net Search People Software

Asilomar Conference System Administration

No Author Found

The name Mike D Chalfant was not found. You may add information about this author or do another query.

First Name: Last Name: Middle Initial:

Honoric Title:

Institution:

Department:

Mail Stop:

Street address: City: State:

Zip Code: Country:

Telephone number (with Area/Country codes):

Fax number (with Area/Country codes):

E-mail:

 Document Done 

Figure 4-25. Authsrc.exe: No Match Found in Database.

Netcape [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Go Back Home Edit Reload Go Back Open Print Find

Location: http://131.120.20.70/cgi-win/AuthSrch.exe

What's New? What's Cool? Destinations Net Search People Software

Asilomar Conference System Administration

Author Information

We have located this record in the database. Changes made here modify the permanent record.

First Name: Last Name: Middle Initial:

Honorific Title:

Institution:

Department:

Mail Stop:

Street address:

City: State:

Zip Code: Country:

Telephone number (with Area/Country codes):

Fax number (with Area/Country codes):

E-mail:

The database has found a unique entry based on the first and last name you submitted. If you believe this entry refers to another person, then you have two choices:

- If the author might have used another name, then [Go back and re-submit your query](#), or
- If this is the correct name but not the desired record, then [Add a new author name to our records](#)

Document Done

Figure 4-26. Authsrch.exe: Unique Match Found in Database.

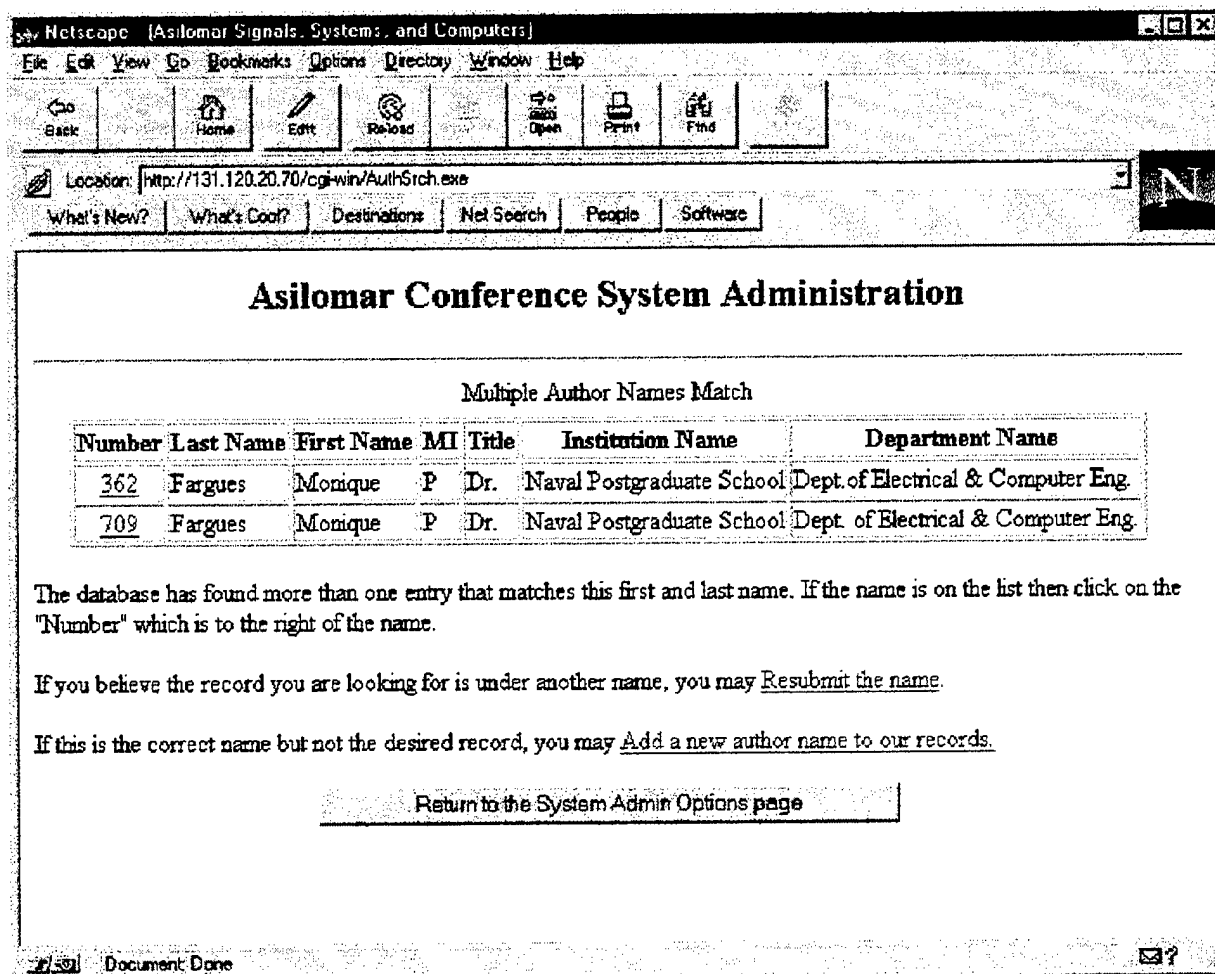


Figure 4-27. Authsrch.exe: Multiple Matches Found in Database.

Netcape [Asilomar Signals, Systems, and Computers] 98 1 23

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Open Print Find

Location: http://131.120.20.70/cgi-win/ArtEdit.exe

What's New? What's Cool? Destinations Net Search People Software

Asilomar Conference System Administration

In order to select the Article you would like to view, you will have to select it from a list of articles. You have three options for building this list:

Choose from a list of ALL the Articles?

[Show me a list of ALL of the Paper Titles](#)

Choose from a list of Articles with these characteristics?

Invited Papers:

Accepted Papers:

Sessions:

Keyword:

[Show me a list of Articles like this](#)
[Reset Article Fields](#)

Choose from a list of Articles by this Author?

Author First Name:

Author Last Name:

[Find an Article by this Author](#)
[Clear Names](#)

[Return to the System Admin Options page](#)

Document Done

Figure 4-28. Artledit.exe.

Netscape - [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Open Print Find

Location: <http://131.120.20.70/cgi-win/ArtSrch.exe>

What's New? What's Cool? Destinations Net Search People Software

Asilomar Conference System Administration

There are 321 records in the Submission Table.

Number	Title
262	2-1 Redundant Binary Addition with Threshold Logic
92	3-D Segmentation-based Video Processing
259	7/2 Counters and Multiplication with Threshold Logic
149	A Bit-level Systolic 2D-IIR Digital Filter Without Feedback
311	A Communication System Testbed
343	A Compound Near-Far End Least Square-Fourth Error Minimization for Adaptive Echo Cancellation
216	A Consistently Convergent OBE Algorithm that Does Not Require a priori Knowledge of Noise Bounds
199	A Fast Algorithm for the Blind Separation of Digital Co-channel Signals
307	A Fast Normalizer
121	A Generalization of the PDA Target Tracking Algorithm Using Hypothesis Clustering
341	A Genetic Algorithm for Register Allocation
42	A High-Performance Reduced-Complexity GMSK Demodulator
58	A Maximum Likelihood Approach to Blind Separation of Narrowband Digital Communication Signals
218	A Modified Root-Finding Formant Estimation Algorithm Based on LP Analysis
172	A Neural Network Training Algorithm Utilizing Multiple Sets of Linear Equations
239	A New Approach for Design of Paraunitary Filter Banks with Linear Phase
43	A New Approach to the Design of Linear Phase Biorthogonal Filter Banks and Wavelets
186	A New Factorization and Structure for Cosine Modulated Filter Banks with Variable System Delay
219	A New Method for Conversion of a 2's Complement to Canonic Sign Digit Number System and its Representation
147	A New Type of Perfect-Reconstruction QMF Banks
274	A Nonuniform Filterbank for Auditory Modeling
175	A Novel Cross-Correlation Based Approach to Blind Equalization and Comparison with Existing Methods
116	A Novel Error-Correcting Codec for M-ary Orthogonal Modulations
222	A Novel Systolic Design for Fast Computation of the Discrete Hartley Transform
79	A Parallel Implementation of an Adaptive Correlation Enhancer
288	A Parallel Two-Dimensional Block Adaptive Filter with Reduced Computational Complexity
315	A Parallel Two-Dimensional Block Adaptive Filter with Reduced Computational Complexity
352	A Realization of the LMS Algorithm with Sigma Delta Input Signals

Document Done

Figure 4-29. Artsrch.exe.

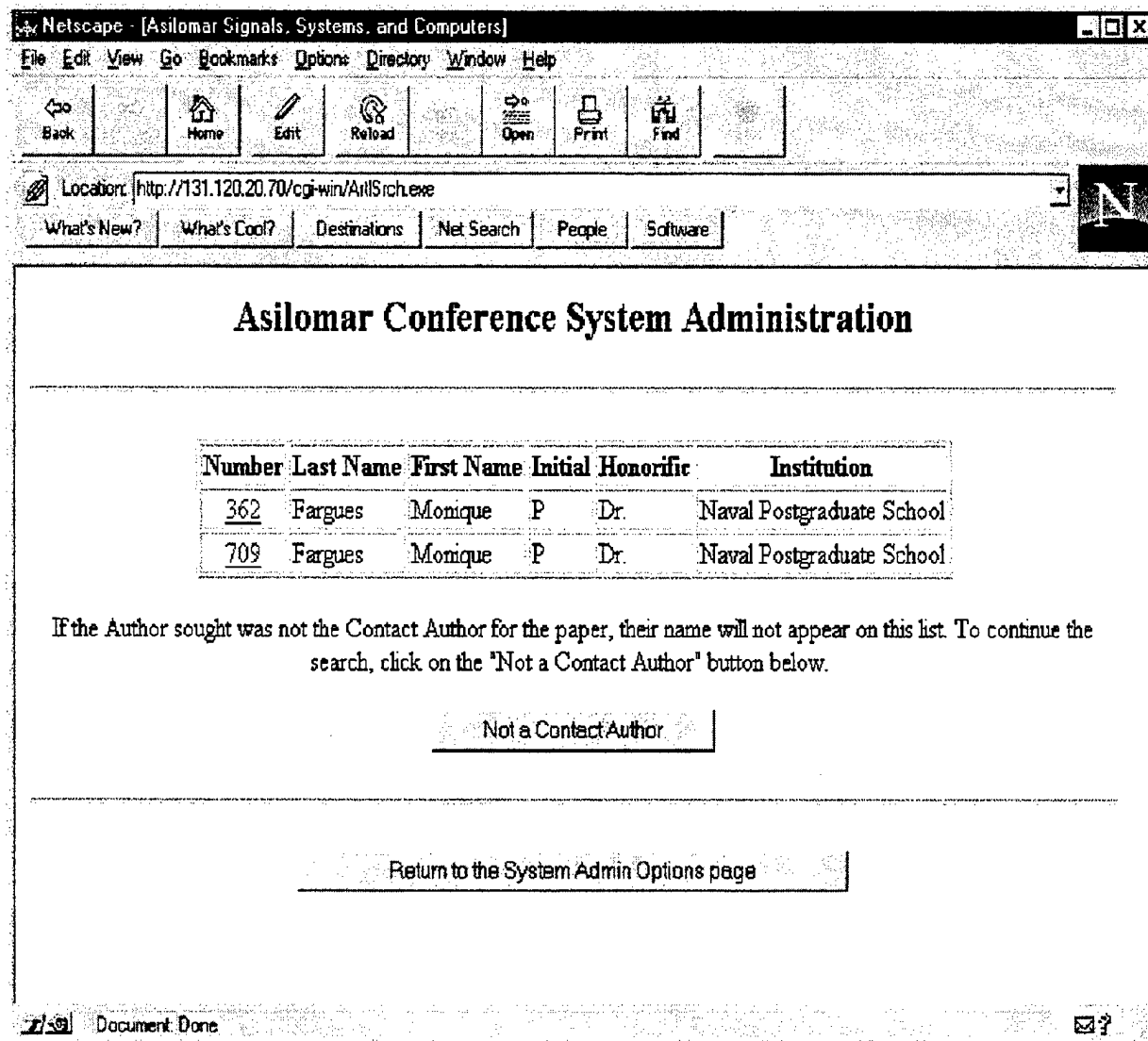


Figure 4-30. Artlauth.exe.

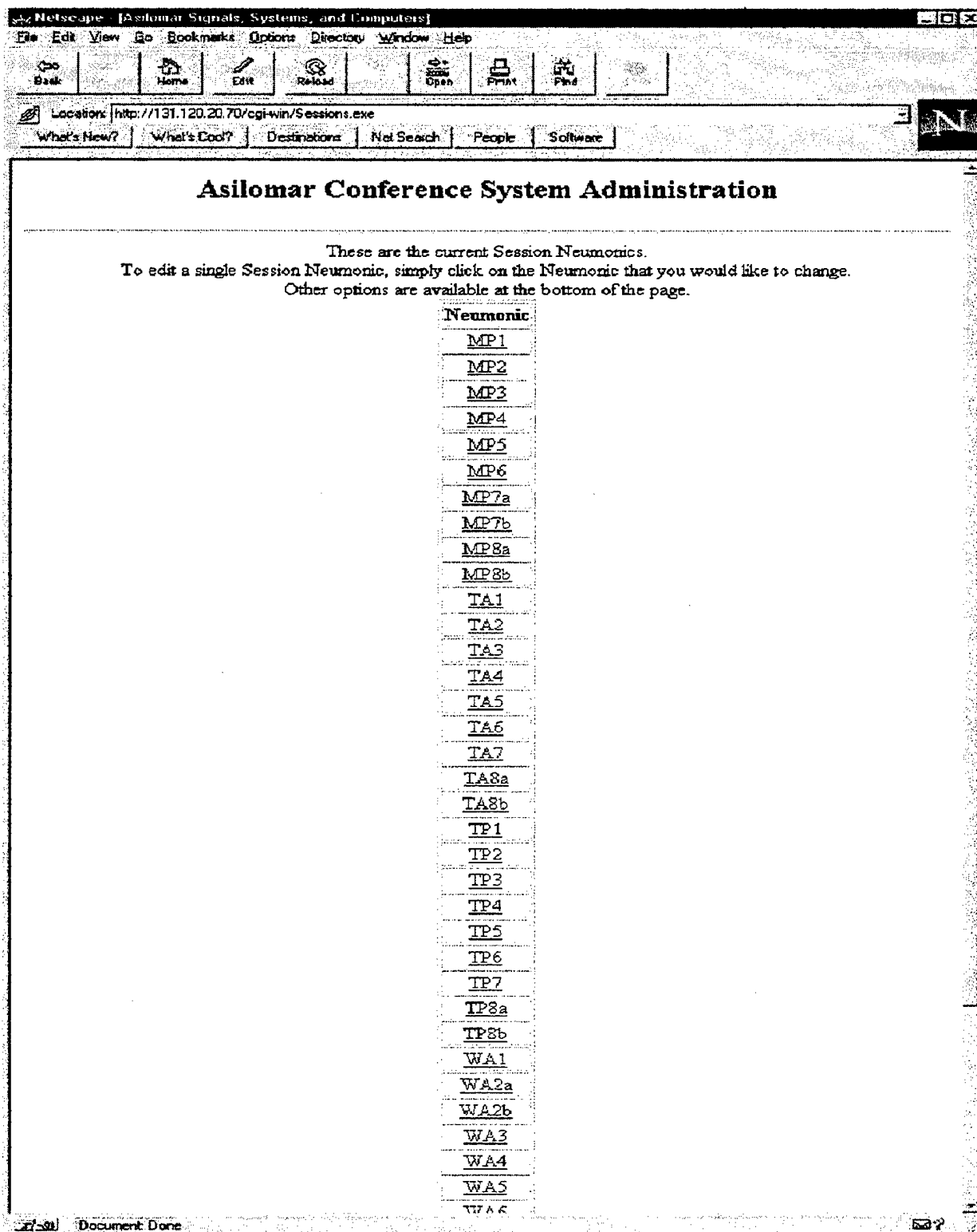


Figure 4-31. Sessions.exe.

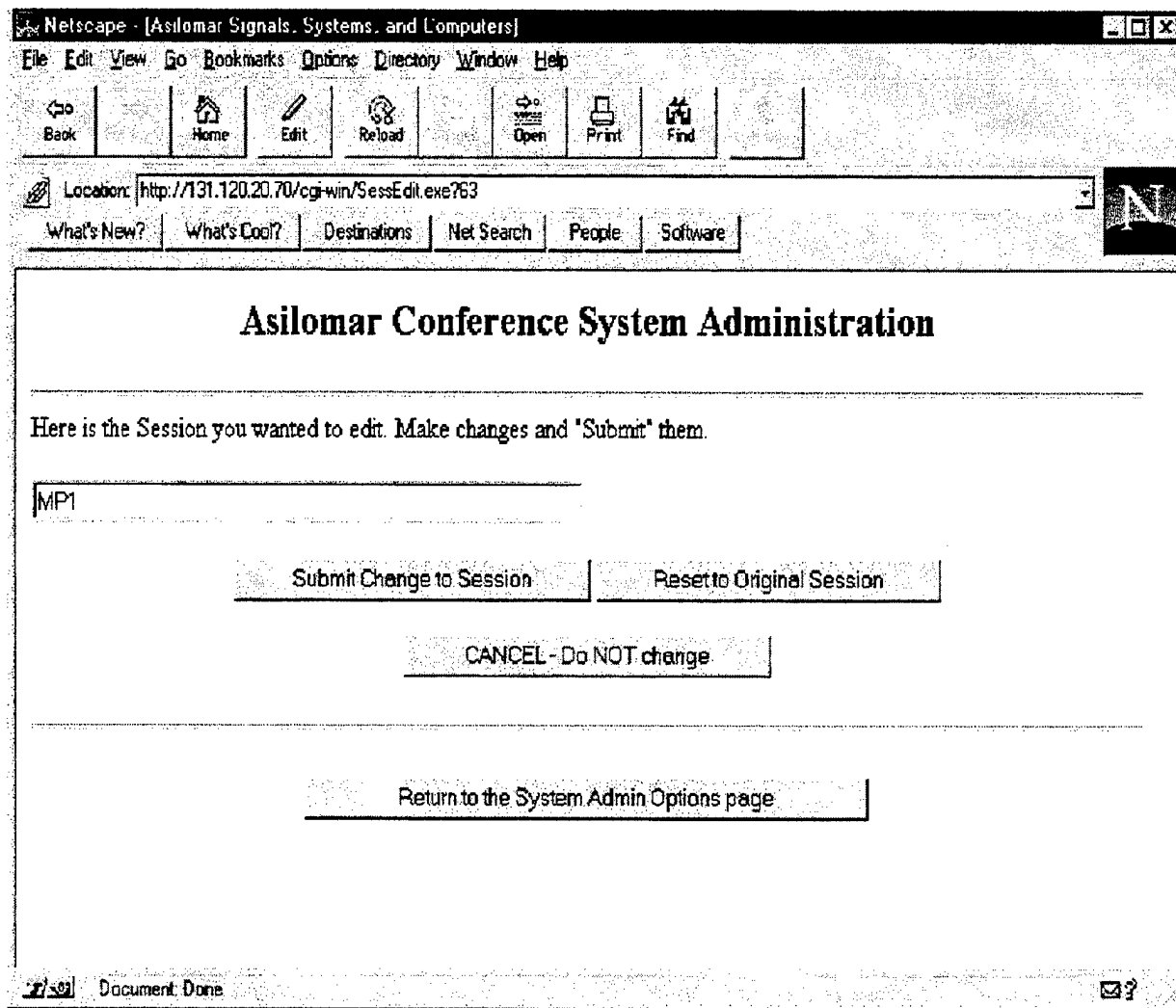


Figure 4-32. Sessedit.exe.

Netscape - [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Open Print Find

Location: http://131.120.20.70/cgi-win/SessAdd.exe

What's New? What's Cool? Destinations Net Search People Software

Asilomar Conference System Administration

Fill in the Session Neumonics you would like to add.
You may fill in all or none of these fields.

New Session:

New Session:

New Session:

New Session:

Document: Done

Figure 4-33. Sessadd.exe.

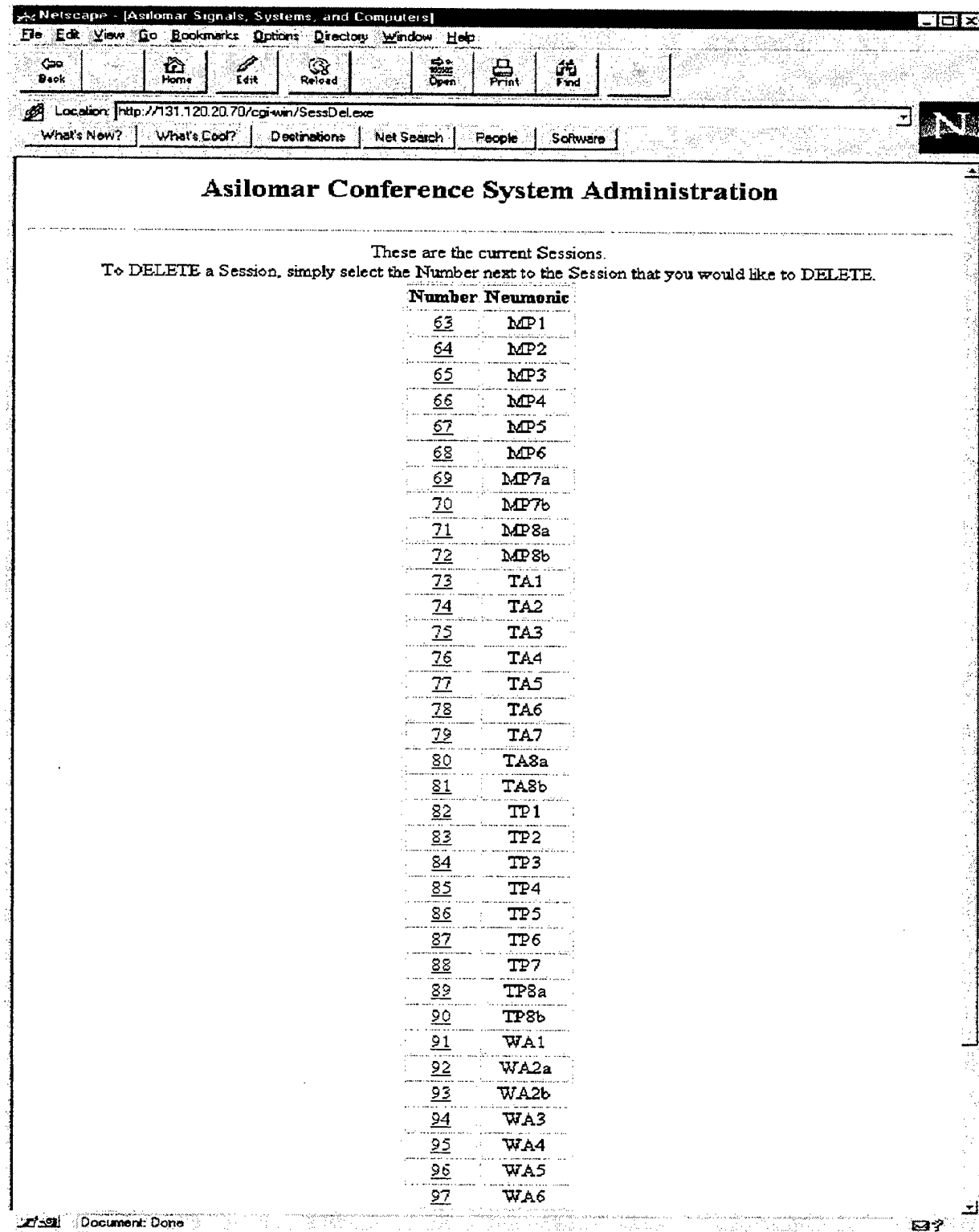


Figure 4-34. Sessdel.exe.

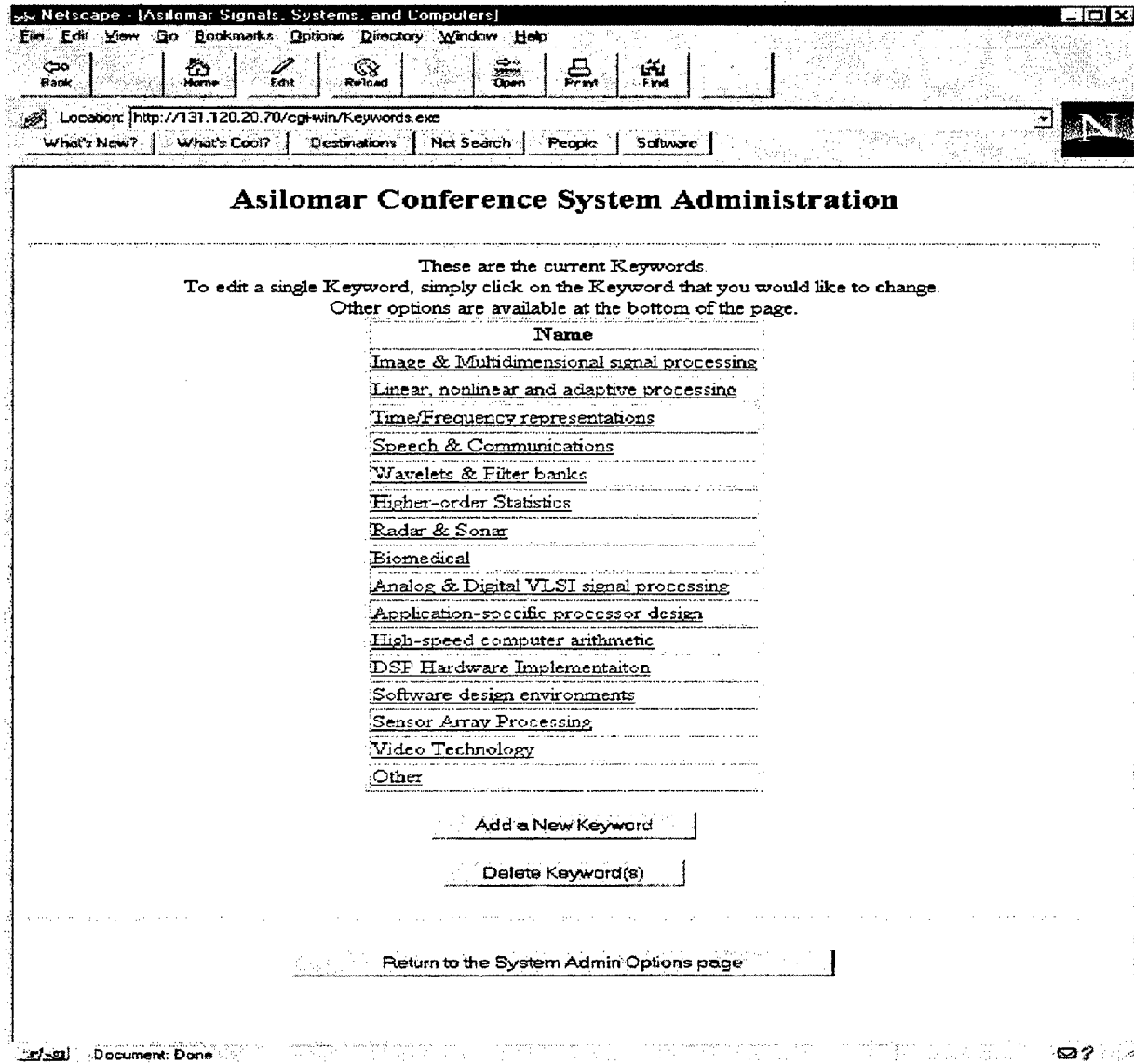


Figure 4-35. Keywords.exe.

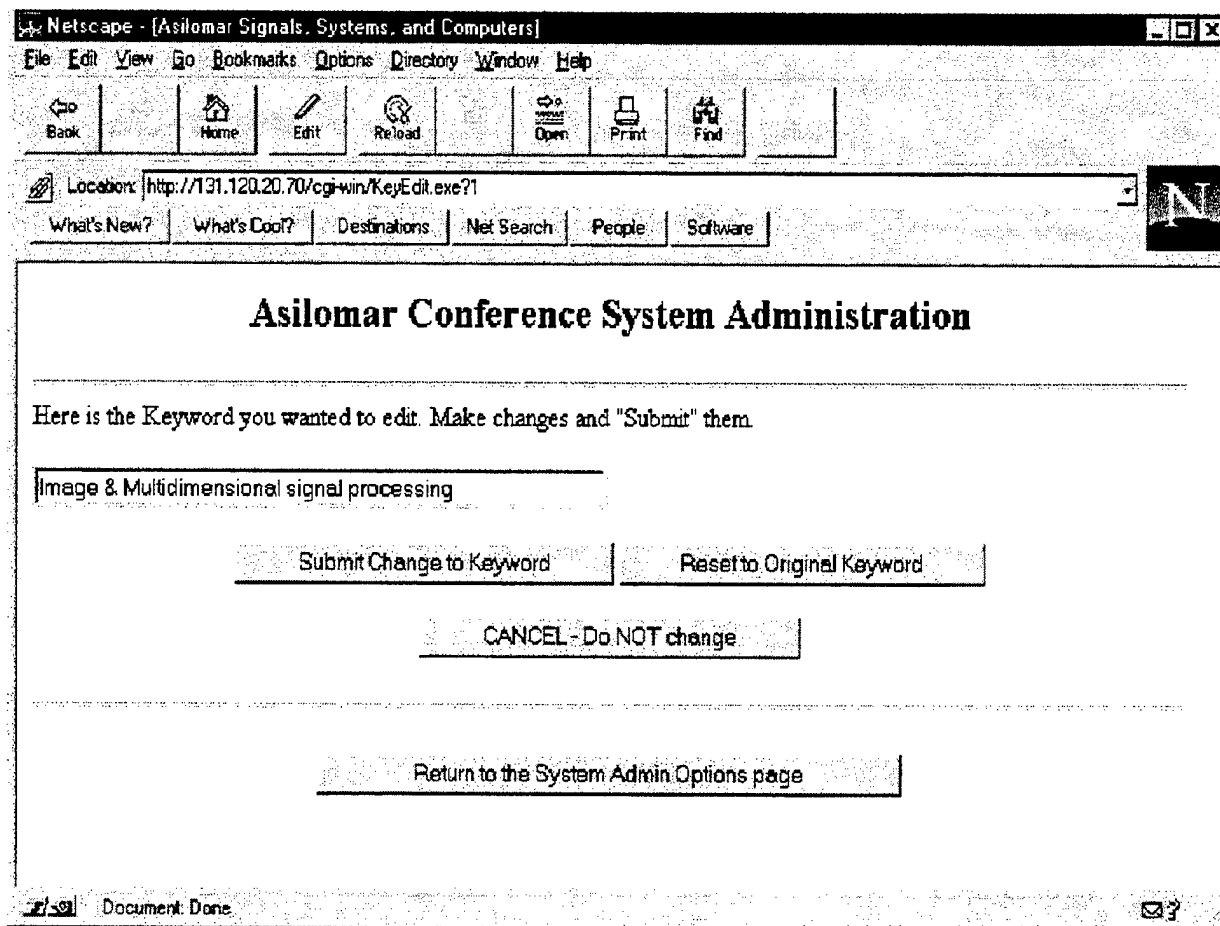


Figure 4-36. Keyedit.exe.

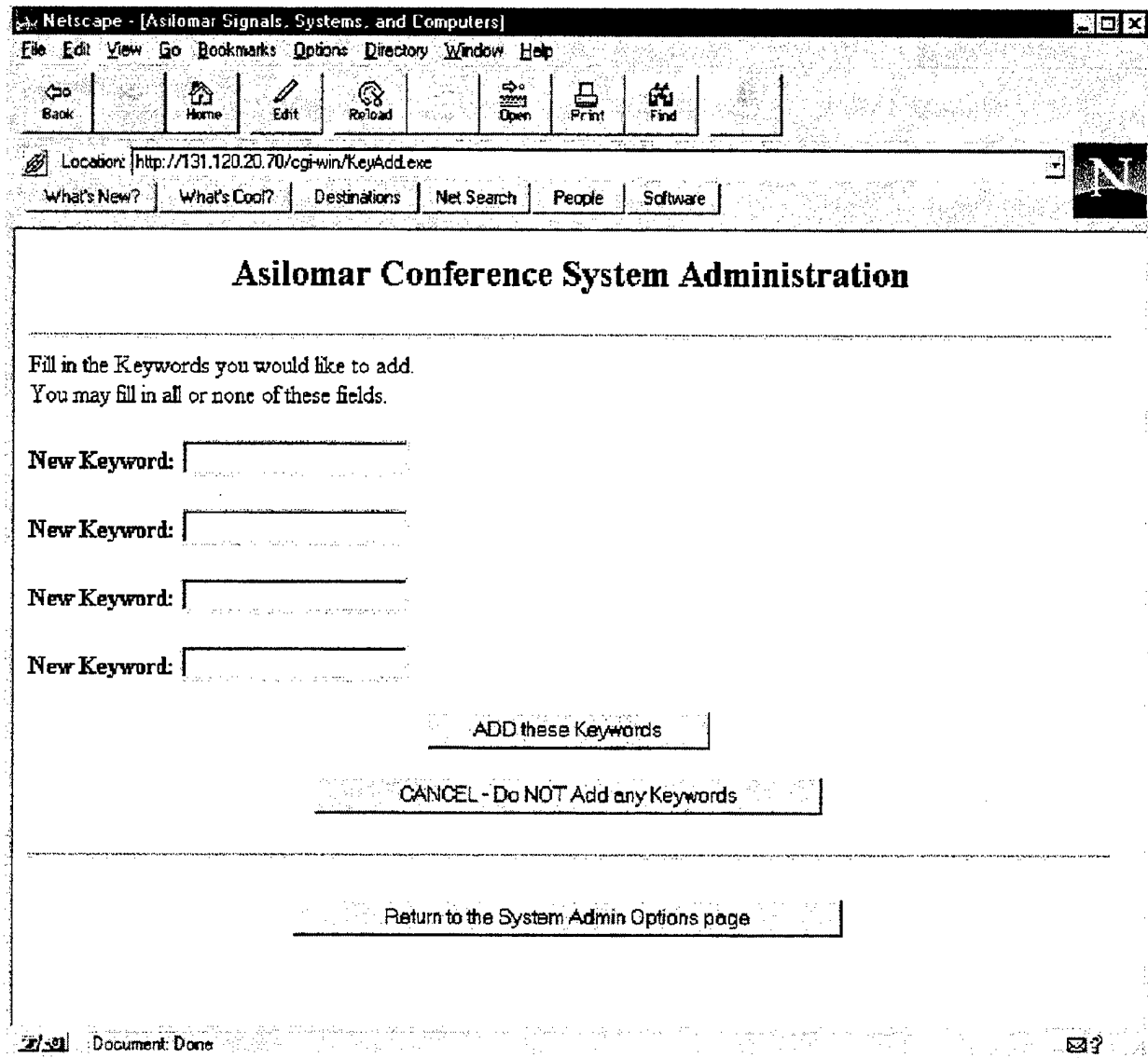


Figure 4-37. Keyadd.exe.

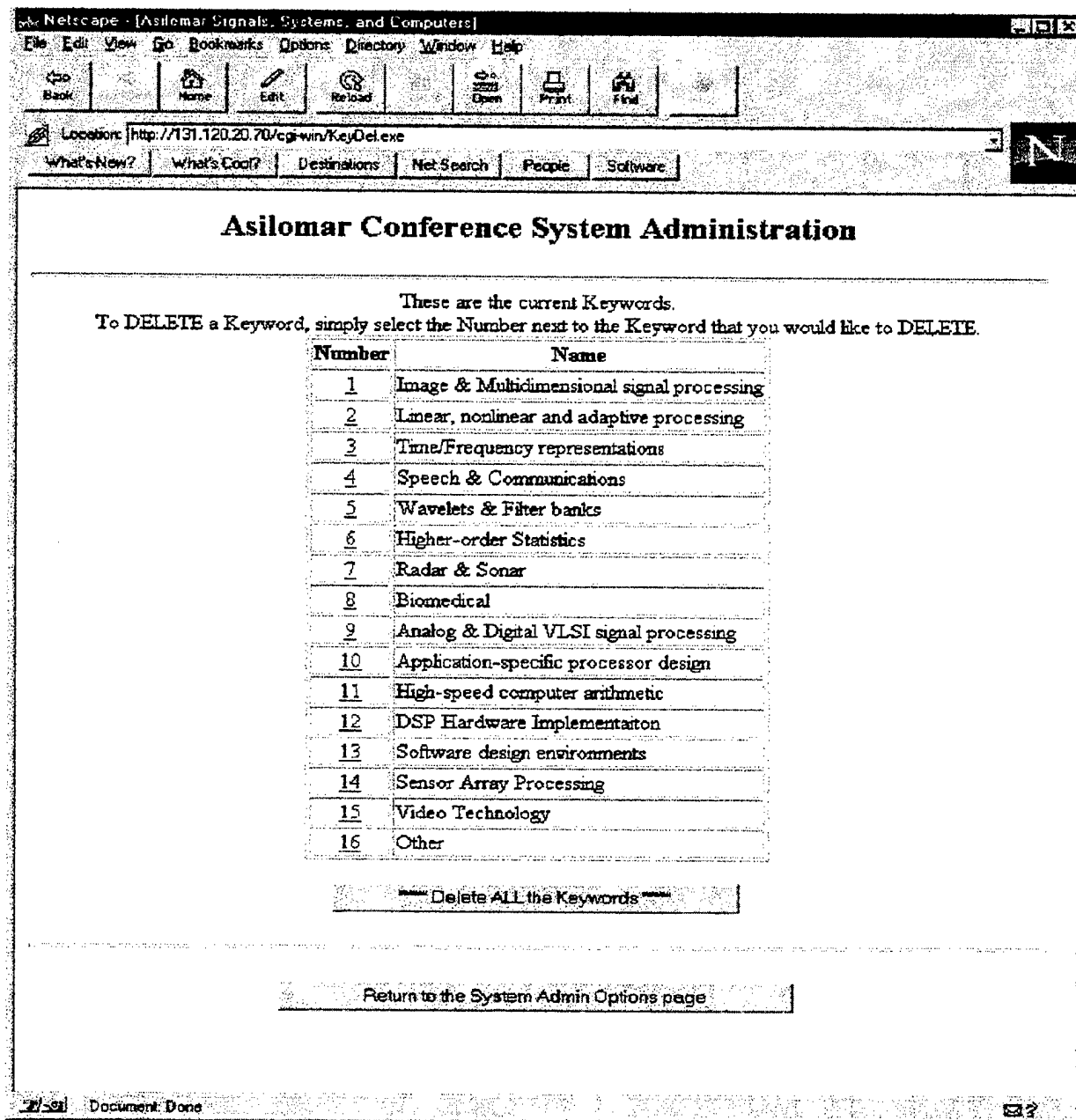


Figure 4-38. Keydel.exe.

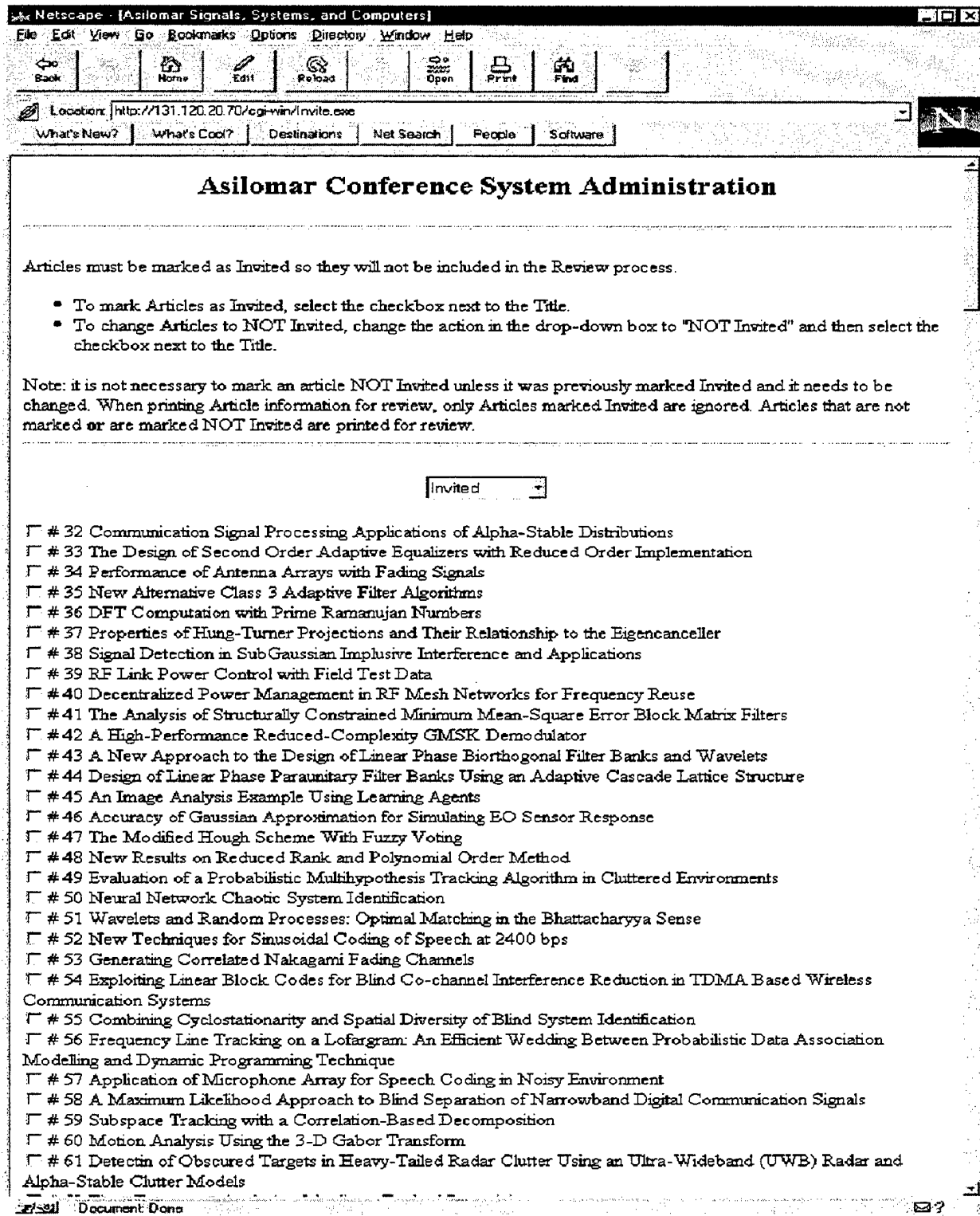


Figure 4-39. Invite.exe.

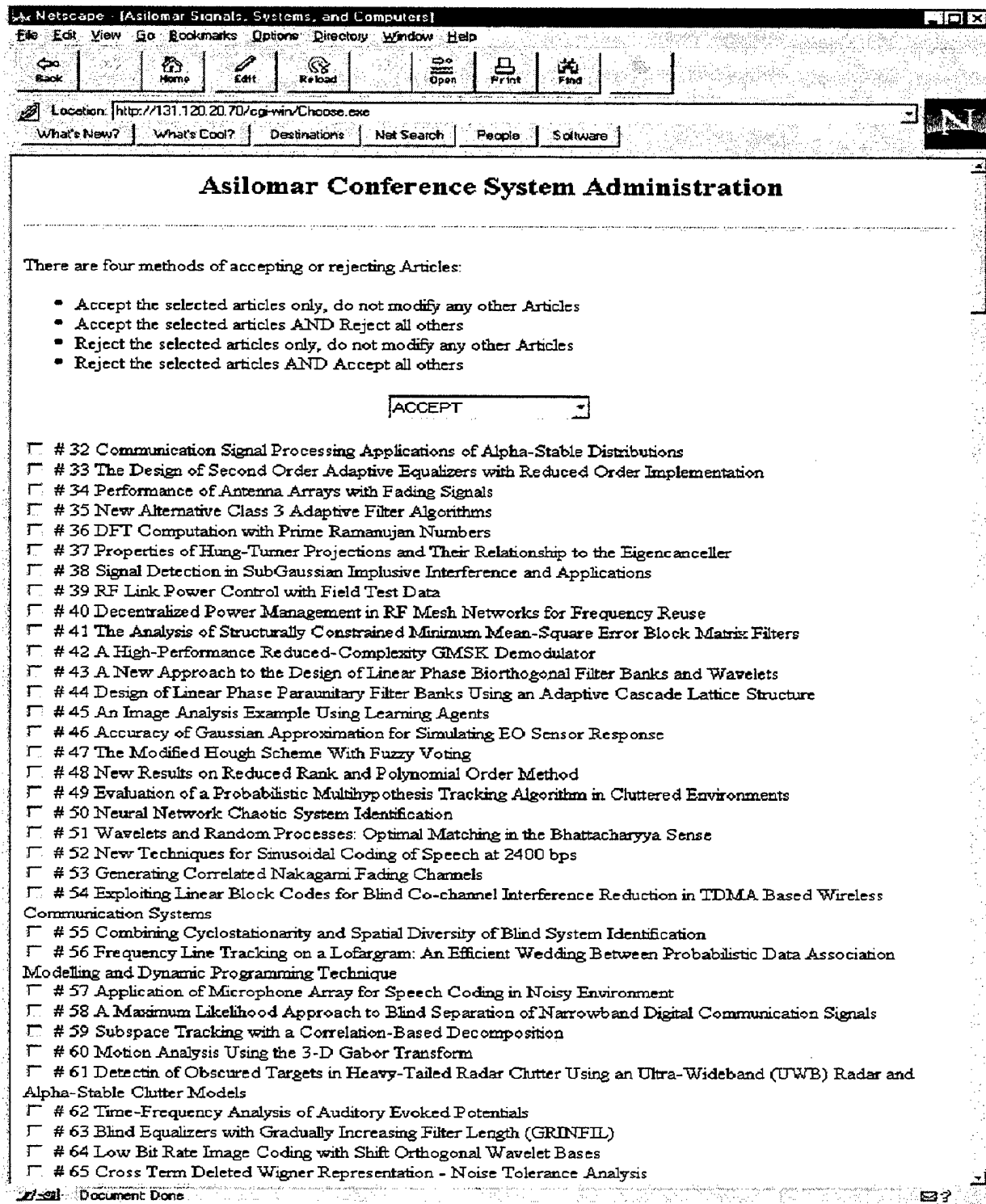


Figure 4-40. Choose.exe.

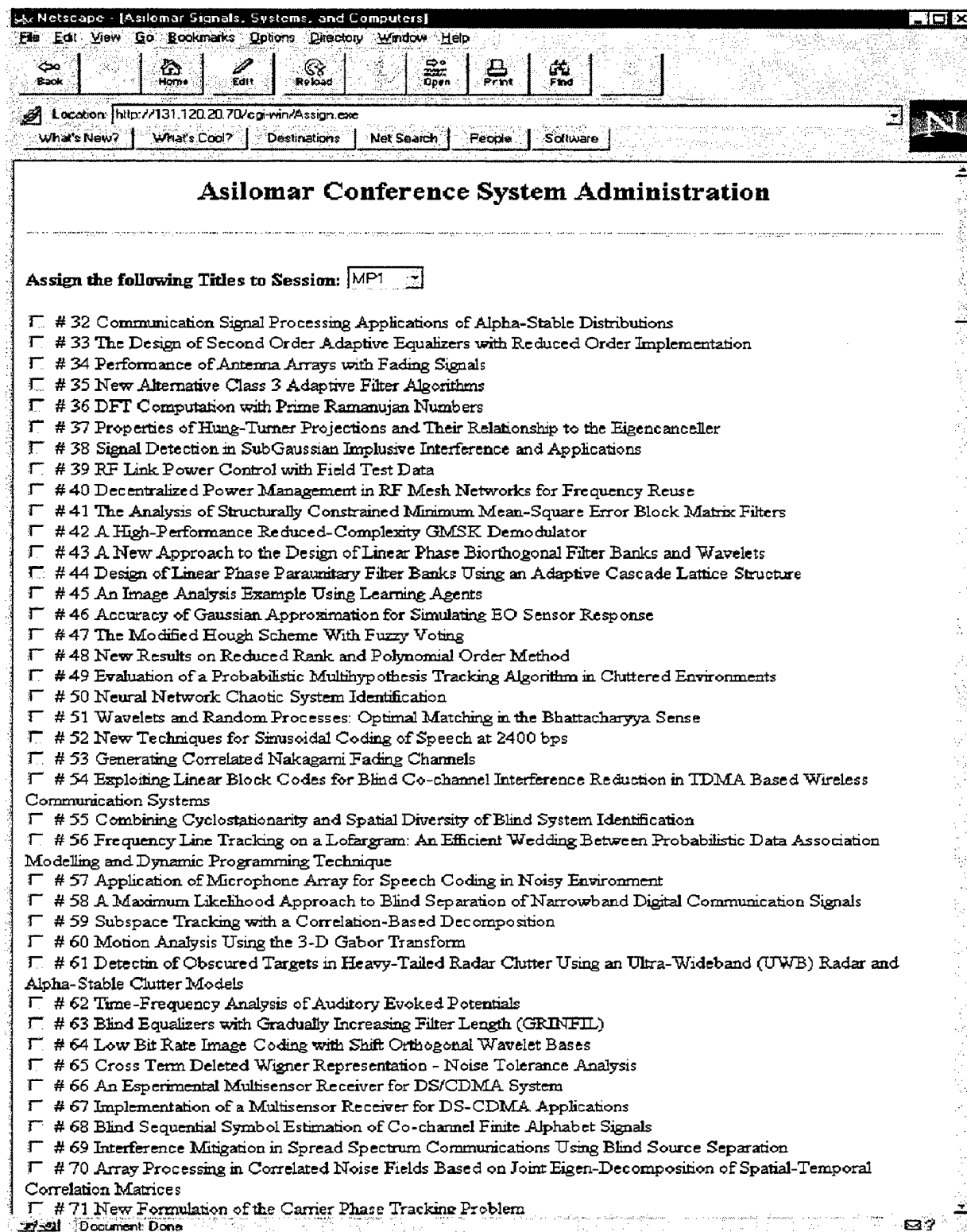


Figure 4-41. Assign.exe.

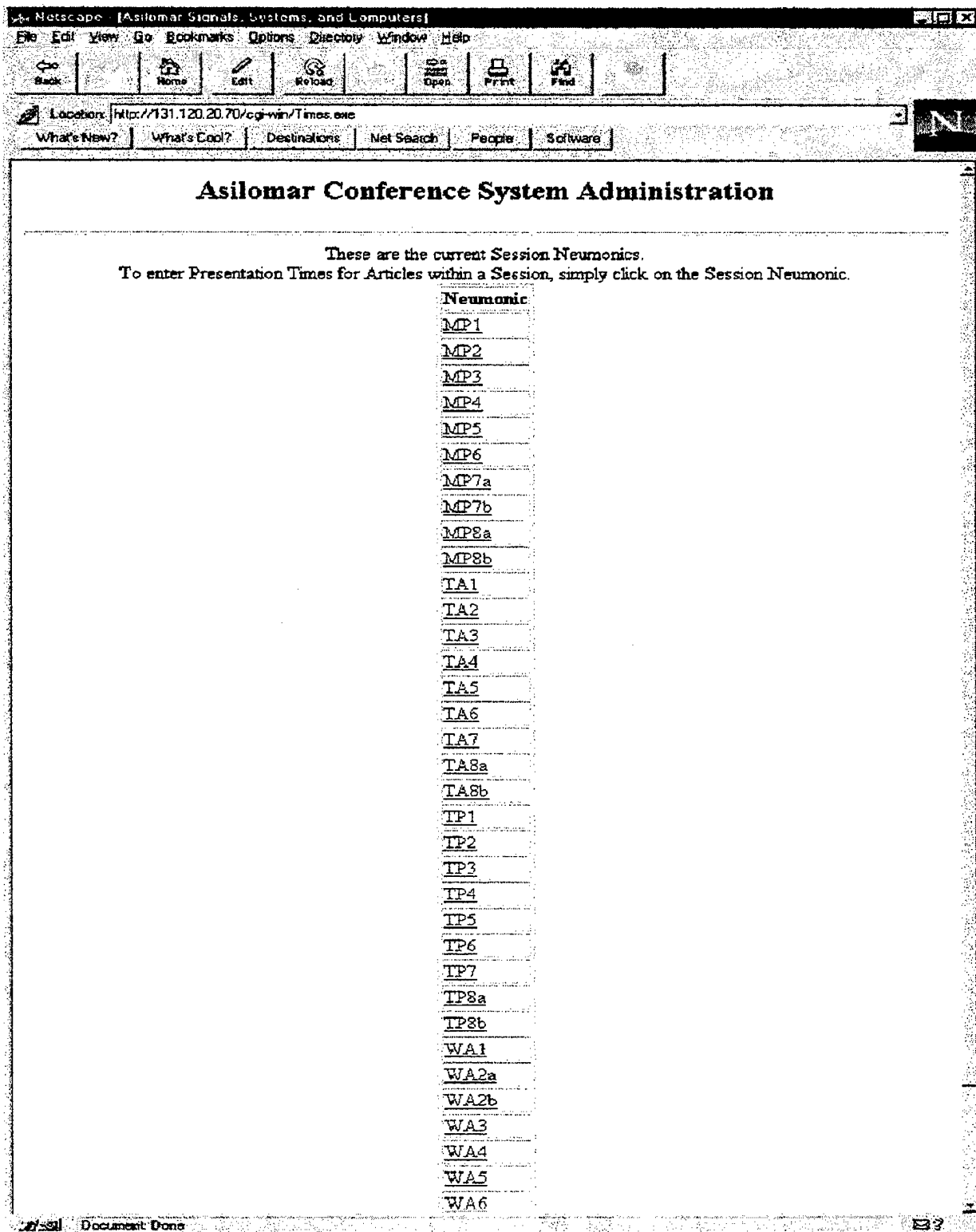


Figure 4-42. Times1.exe.

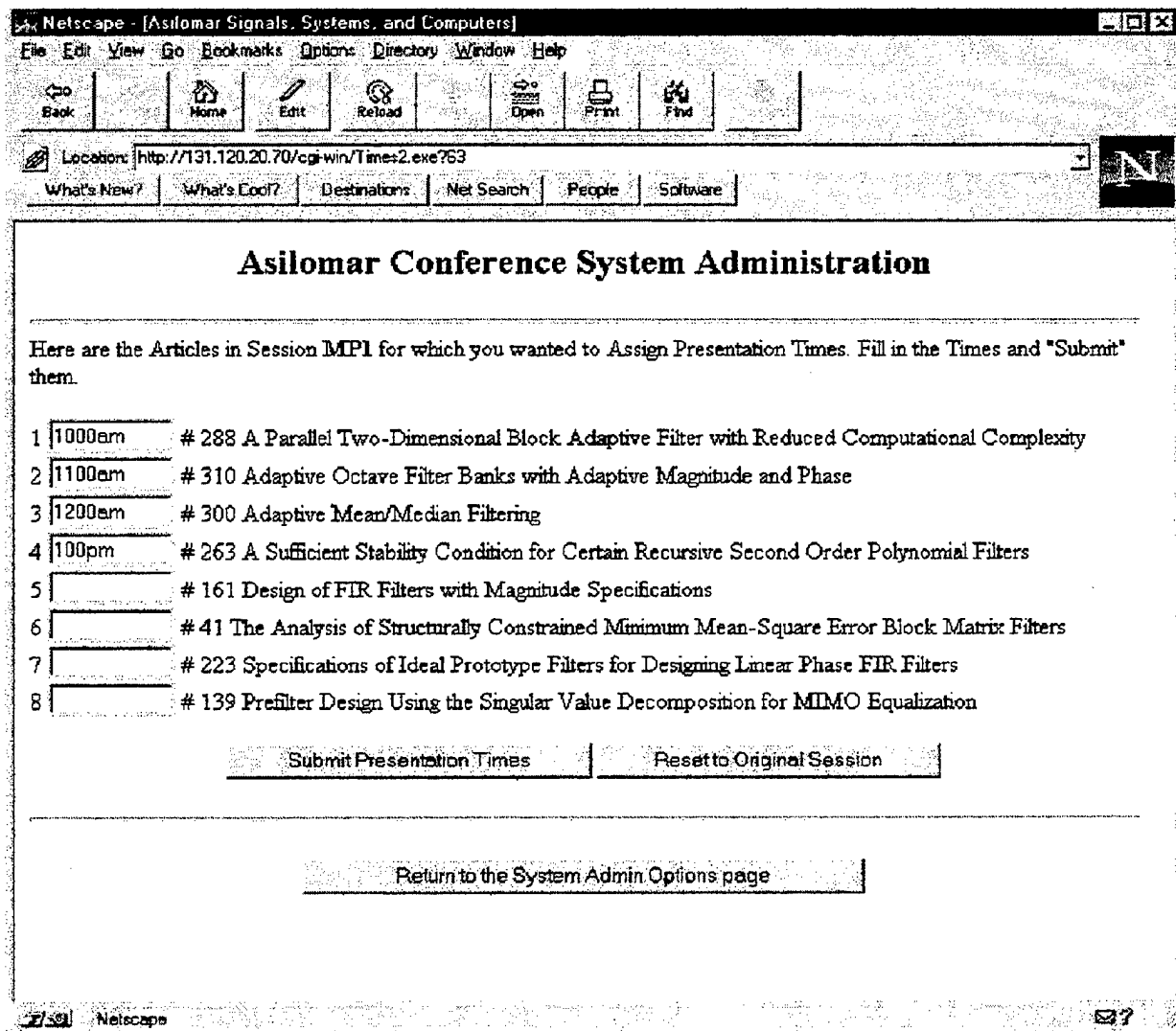


Figure 4-43. Times2.exe.

Netscape [Astomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Pass Open Print Find

Location: http://131.120.20.70/cgi-win/Passwds.exe

What's New? What's Cool? Destinations Net Search People Software

System Admin user passwords.

To change a System Admin password, enter the information below and "Modify a System Admin User Password." The new password must be entered twice to ensure validity. To ADD or DELETE a User, select the appropriate Buttons below.

User name:

Old Password:

New Password:

New Password Again:

Reviewer passwords.

To change a reviewer password, enter the information below and "Modify a Reviewer Password." The new password must be entered twice to ensure validity. To ADD or DELETE a Reviewer, select the appropriate Buttons below.

Reviewer name:

Old Password:

New Password:

New Password Again:

Document: Done

Figure 4-44. Passwds.exe.

Netscape - [Asilomar Signals, Systems, and Computers]

File Edit View Go Bookmarks Options Directory Window Help

Back Home Edit Reload Open Print Find

Location: http://131.120.20.70/cgi-win/PwdAdd.exe

What's New? What's Cool? Destinations Net Search People Software

Asilomar Conference System Administration

To ADD a new System Admin User, please enter the following. The password is entered twice to ensure validity

New User name:

Password:

Password Again:

Netscape

Figure 4-45. Pwdadd.exe.

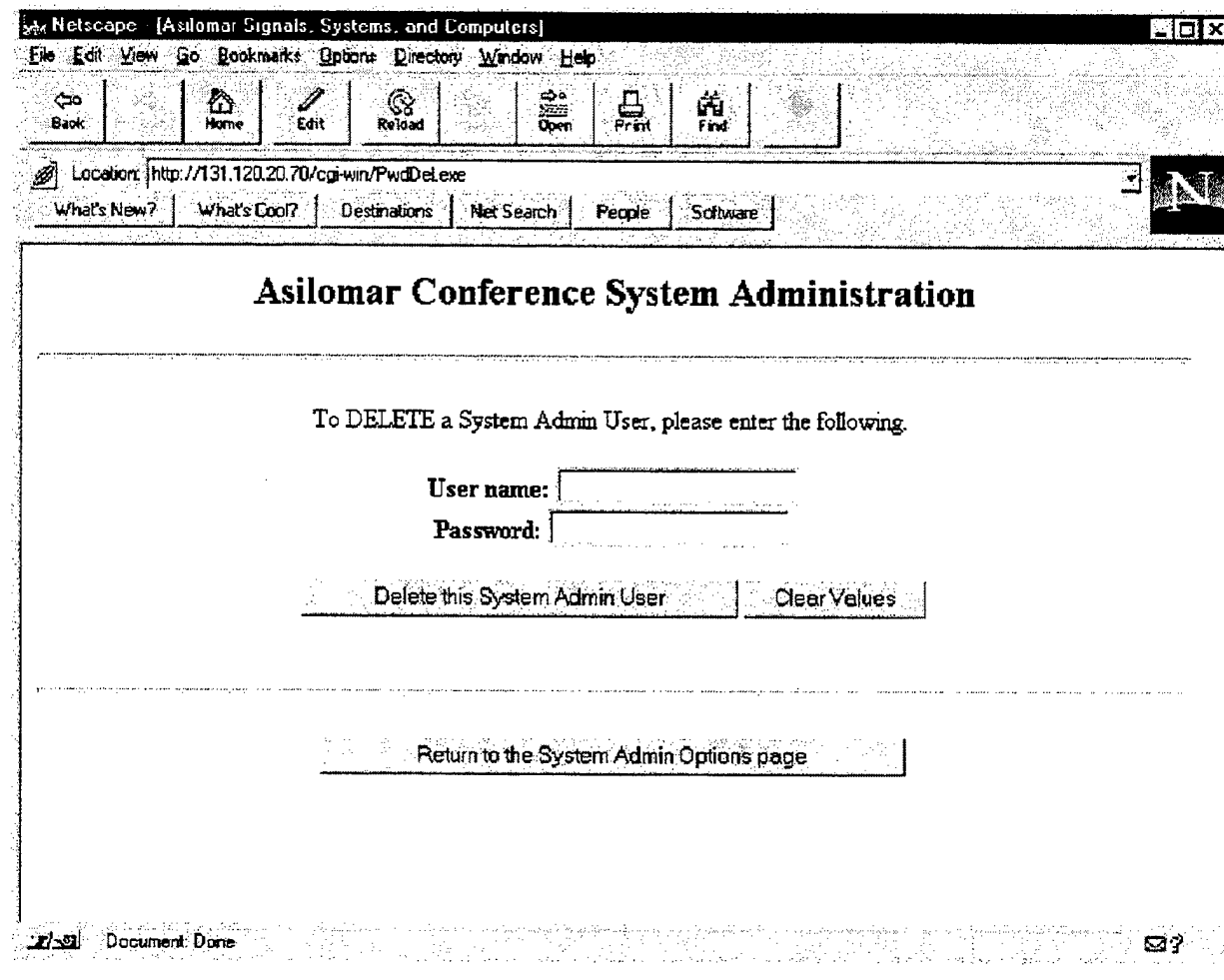


Figure 4-46. Pwddel.exe.

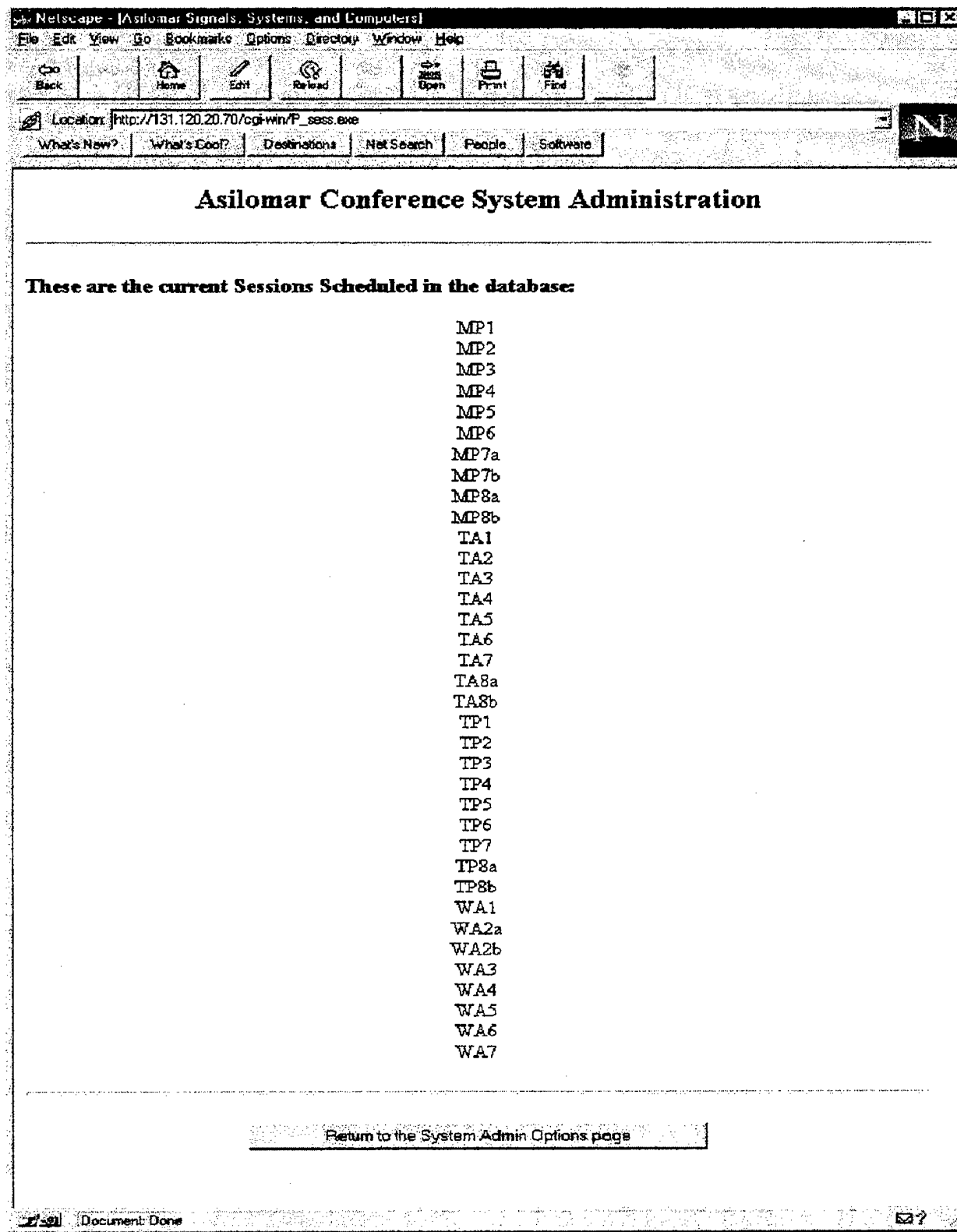


Figure 4-47. P_Sess.exe.

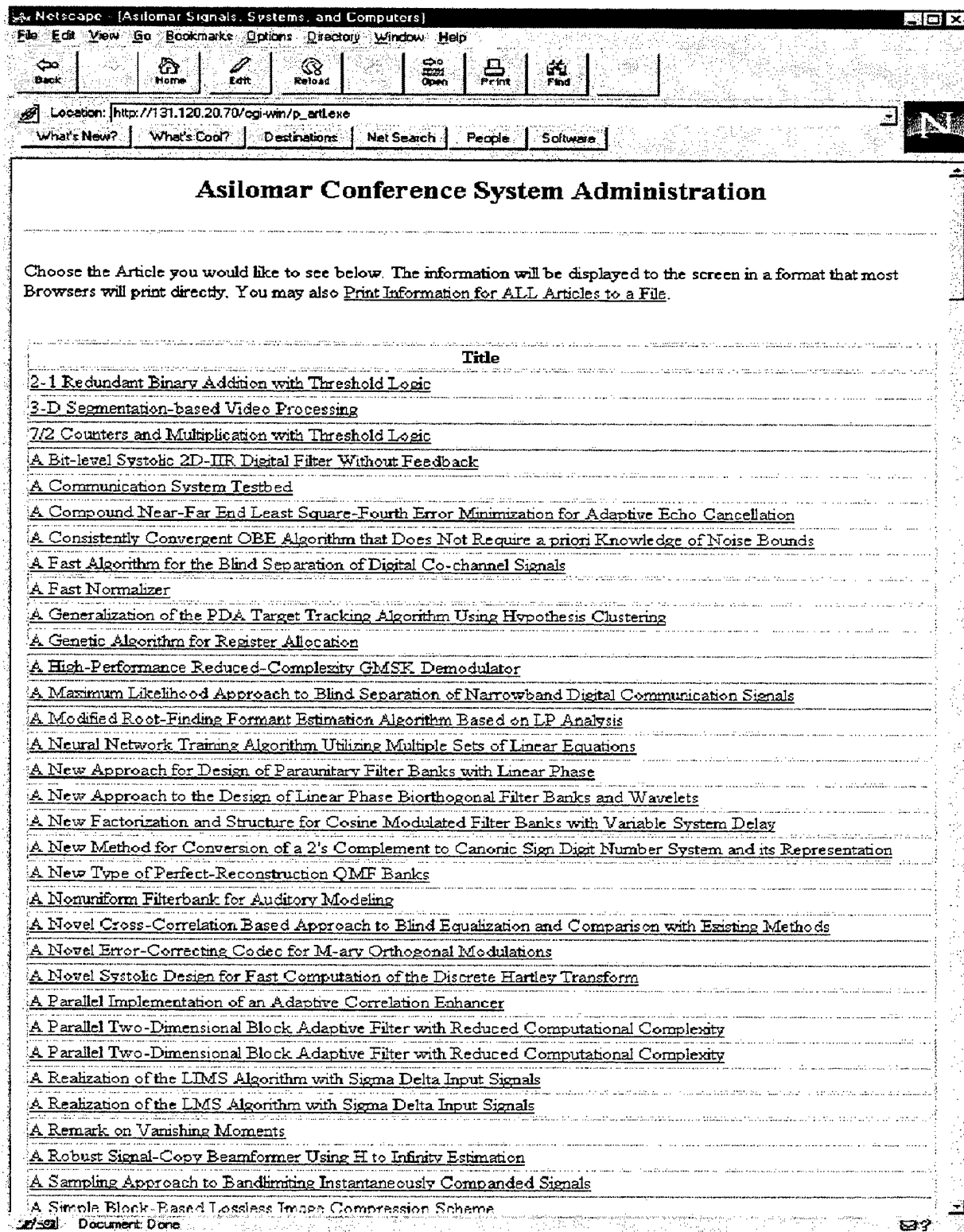


Figure 4-48. P_Artl.exe.

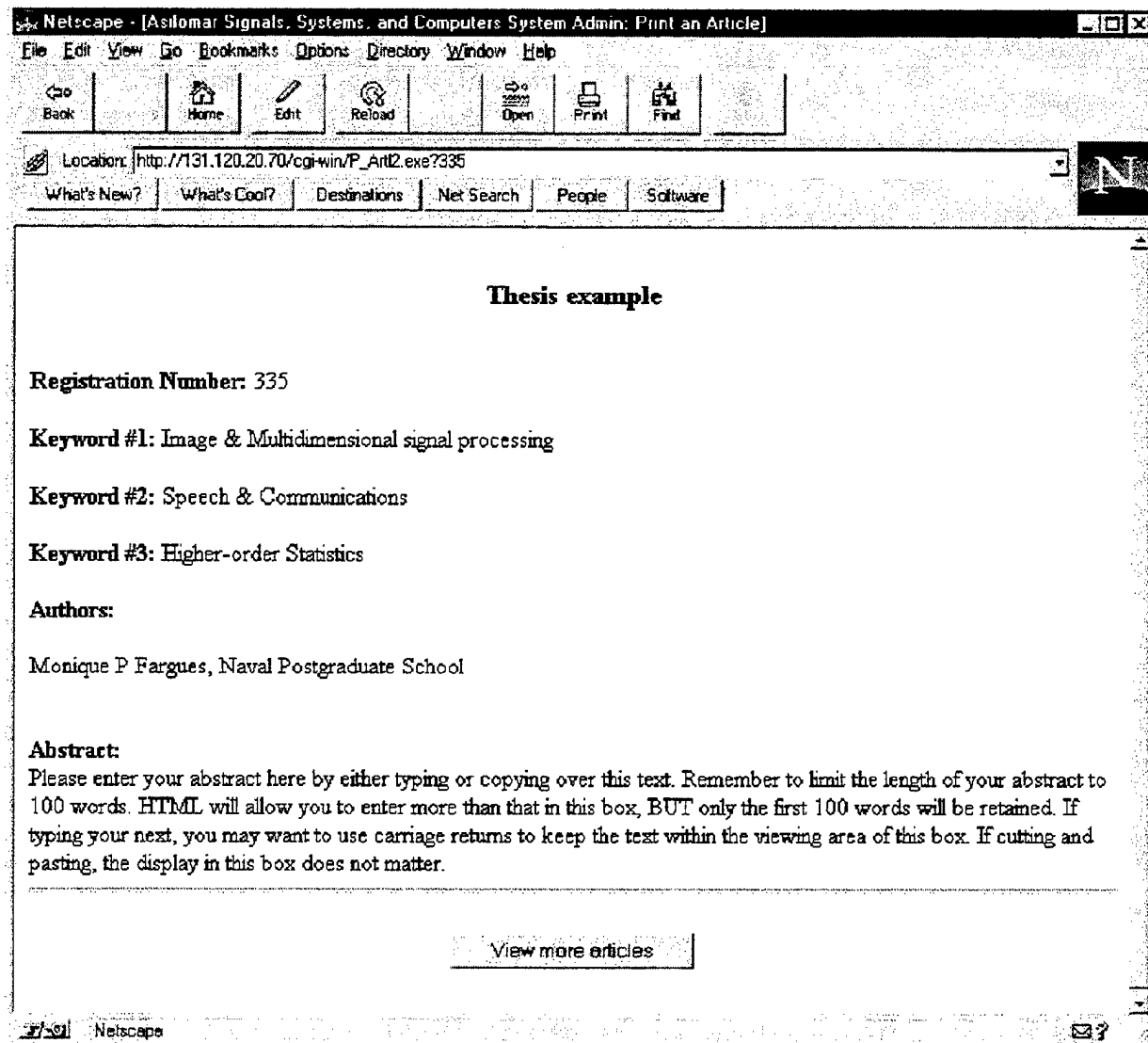


Figure 4-49. P_Art12.exe.

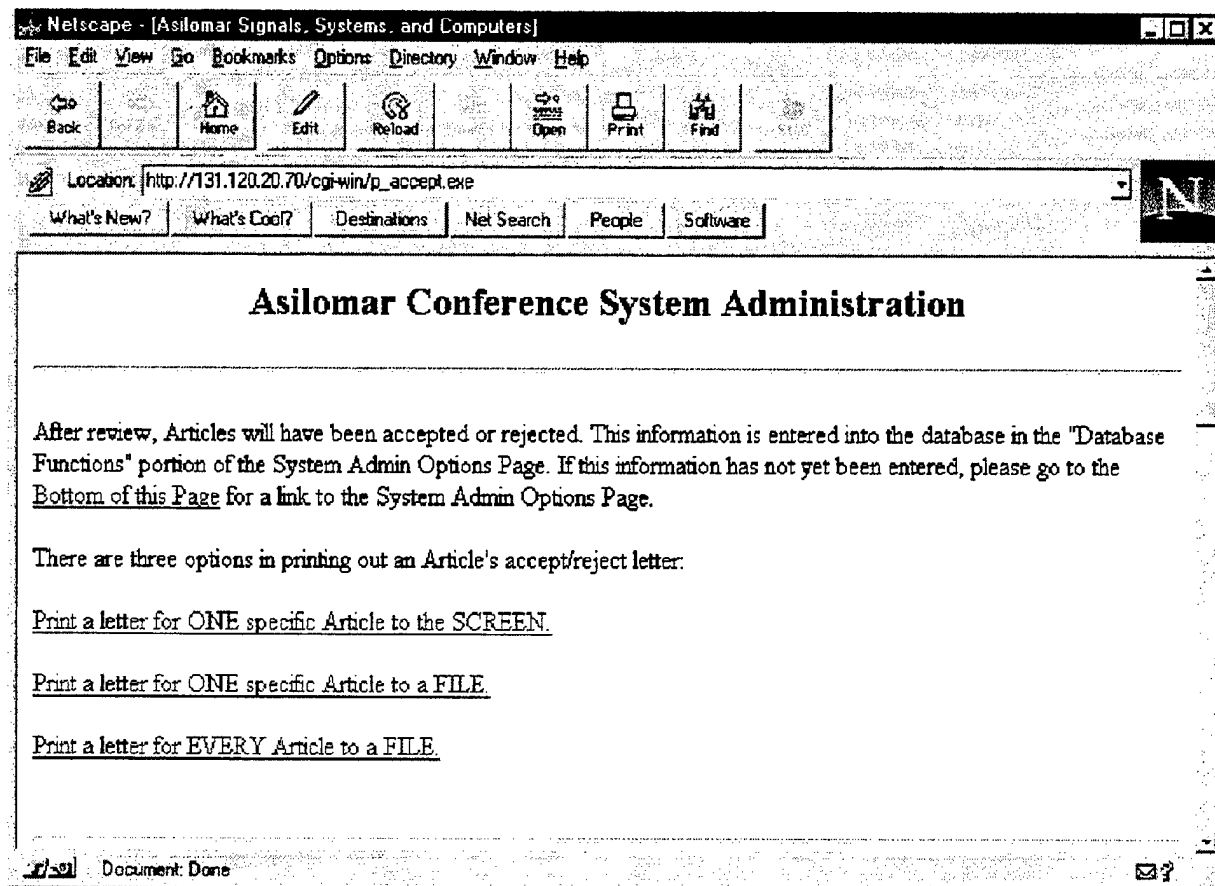


Figure 4-50. P_Accept.exe.

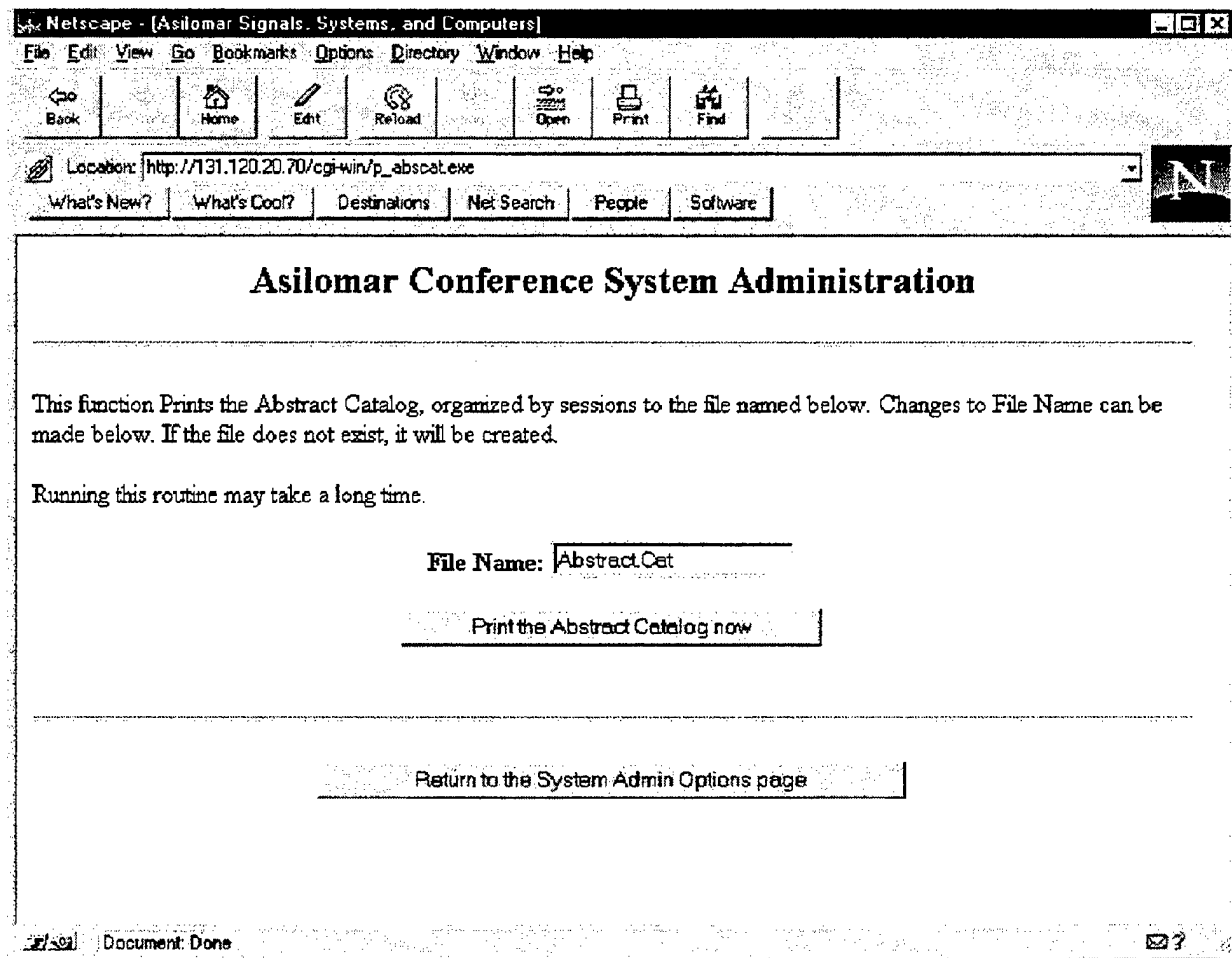


Figure 4-51. P_abscat.exe.

V. CONCLUSION & RECOMMENDATIONS

A. CONCLUSION

The Asilomar Conference on Signals, Systems, & Computers has begun the transition to the information age. The goals of this thesis were to: 1) analyze the Asilomar Conference on Signals, Systems, & Computers, 2) improve the article submission and review process, 3) develop a target information system and, 4) implement a portion of the target system. By attaining these goals, we have not only streamlined and automated many of the processes of the Asilomar Conference, but also have developed a framework for an Internet conferencing system. In the future, it could no longer be necessary for people to gather in a single geographic location for such a conference. By exploiting the power of the Internet, registration, reviewing, database maintenance, and even presenting can be done from any Internet connection.

B. UPGRADES TO CURRENT SYSTEM

1. Change the current Delphi executable programs to 32-bit executables.

This improvement could be accomplished by upgrading to Delphi 2.0 and importing the current programs and re-compiling them. Some changes will need to be made to insure compatibility. The 32-bit programs should run faster than the current 16-

bit programs. Switching to Delphi 2.0 would also allow the use of HREF Tools Corp's WEBHUB [Ref. 4] which more efficiently handles multiple users and increases the performance of the server software.

Executables created by either version 1.0 or 2.0 of Delphi have a small drawback. Variable information cannot be passed between executable programs. That is, once a program has completed its function and calls another program, all variable information (known as program *state*) is lost. WEBHUB, which is only compatible with Delphi 2.0, maintains information stored in variables after an executable is complete. This is enhanced by the fact that WEBHUB can maintain program state for multiple users accessing the same portion of the system.

2. Purchasing a more powerful FTP server.

A commercially available FTP server would improve upon the capability and security of the shareware program currently in place. Purchased software would also include technical support not available with the unregistered version of FTPD.

C. NEW FEATURES

1. Reviewing articles on-line.

Automating the review process would enable reviewers to review papers

remotely. The present methodology could be modified to include making the review "blind" or even "double-blind" to ensure fairness. Papers could be reviewed by multiple reviewers who each give an accept or reject vote along with whatever comments they feel are appropriate. The final determination could be made manually, human tallying of the votes, or automatically, with the system counting the votes and only presenting "tie-votes" for human determination. The automated tally system could even allow for weighting the votes by rating reviewers by experience or expertise.

2. Scheduling.

An automatic scheduling routine could be built to schedule the selected articles into sessions. Currently assigned by the reviewers, automating this process would relieve the reviewers of this tedious task. Computer generated schedules could be modified as necessary by the System Administrator.

3. Registration.

Allowing attendees to register on-line would further reduce the mailing and administrative costs to participants and administrators for the conference. On-line payment capability would be a pre-requisite for this implementation. An ideal system would also coordinate requests for Hotel room, rental car, and dinner reservations.

4. Send Mail Utility.

An attempt was made to implement a shareware utility for Delphi called "SendMail" [Ref. 9]. This utility would automatically generate an e-mail acknowledgement once the user had completed the submission process. The functionality was initially implemented into the author submission sub-system, but was removed because the shareware version halted the server each time an e-mail message was sent. The code for SendMail is still in place, but has been commented out. The registered version is supposed to eliminate this problem. This feature, which would acknowledge receipt of a user's inputs, increases the reliability of the data and the users' sense of security about the system.

APPENDIX A. DELPHI TUTORIAL

A. OVERVIEW

In setting up a Windows based web server there are four basic software components. First, web server software such as Website [Ref. 6], WinHTTPD [Ref. 5], or ZBServer [Ref. 11] is needed to link a computer to a specific Internet address. This software will activate your homepage when your IP address is accessed by a user. Web server software packages vary greatly in price and performance but there are similarities in the way all of them operate. In general, once set up, web servers will:

- 1) assume you have already established an Internet connection with a valid IP address
- 2) create the following sub-directories below its own:
 - A) cgi-bin - for Windows Perl cgi scripts
 - B) cgi-win - for other Windows cgi scripts.
 - C) cgi-dos - for DOS Perl cgi scripts
 - D) htdocs - for HTML (Hyper Text Markup Language) documents
- 3) assume you have already created an HTML document that will serve as your homepage. This document usually must be named "Index.htm" and placed in the "htdocs" directory.

(Note: "cgi" stands for Common Gateway Interface and is merely a protocol that scripts must follow. A script is a program that executes on the web server and interacts with

HTML documents.)

The second software item you will need is HTML documents. These documents are written in plain text (any text editor will suffice) and determine how and what a web browser will display. There are many programs available as shareware that greatly simplify the process of creating HTML documents. These include: HTMLAssist [Ref. 12], and HotDog [Ref. 13]. HTML allows the presentation of text, graphics, sounds, and animation, and will allow users to navigate through your pages via hypertext links.

It is possible to run a web site with just the two software packages mentioned above, however users will be limited to merely looking at pages you present. If you want users to be able to interact with your pages, you will need at least one more piece of software. This piece is a scripting language. Languages such as Perl, Delphi, and Cold Fusion create executable files that are run whenever a user accesses your page. These executables can perform such tasks as read data from a form, then create an entirely new page using that data. We will see an example of this later in this tutorial.

Although scripting languages allow users to interact with web pages, information will not be recorded unless the program interacts with a database of some sort. This is the final software requirement. Newer scripting languages have built in database interactions and serve as the go between for web pages and database operations. Data storage, retrieval, queries, and reports can all be performed via a web page using a web

server, HTML documents, a scripting language, and a database.

This was a brief overview of the software components required to run a PC based web server. In general web server software is relatively simple to set up and maintain. The challenge lies in learning and using a scripting language to develop the types of applications that will be useful to you and your users. The rest of this tutorial will focus on using Borland Delphi Application Development software as a scripting language.

B. SAMPLE APPLICATION

The first step in designing any web application is developing a general plan of how each page will look, what functions it will perform, and how the user will navigate through it. The example I will use for this tutorial is an application that runs a game called "MadLib." In this game, the user is required to enter in words of a specific type, then the application uses those words in a pre-formatted story and creates a new web page with the story on it. My plan for this application is as follows:

- 1) I will need an input form to read in the user's words
- 2) I can create that form using basic HTML commands
- 3) I will need a CGI script to:
 - get those words off the form
 - create a new web page with the story using those words

Now that I have decided on what I need, I can create the necessary components. The HTML form which will input user's words then load the CGI application appears below. Explanatory comments are incorporated with the format `<-- comments -->`.

<HTML>

<HEAD>

<TITLE> Delphi test: A MadLib Input form</TITLE>

<CENTER><H1> MadLib Input Form</H1></CENTER>

<HR>

</HEAD>

<BODY>

A "MadLib" is a game where I take your inputs for some words and put these words into a story. Depending on how creative you are, the results can be hilarious!

<P>

Please provide appropriate responses for the following fields. Have fun!<P>

<HR>

<CENTER>

< -- ***** The following statement is the **key statement** of this form. It loads the CGI application "madlib.exe" when the "Submit" button is pressed on the form. Madlib.exe is the executable file that you will create in Delphi *****-- >

<FORM METHOD="POST" ACTION="http://131.120.20.70/cgi-win/madlib.exe">

< -- The "Table" section creates an input table with visible "Borders" between cells. -- >

<TABLE BORDER>

<TD>A girl's name: </TD>

<TD><INPUT NAME="girl"></TD>

```

<TR>

<TD>A boy's name: </TD>

<TD><INPUT NAME="boy"></TD>

<TR>

<TD>A day of the week: </TD>

```

< -- The "Select" command creates a drop down menu with each "Option" available as a selection -- >

```

<TD><SELECT NAME="day" TYPE="TEXT">

    <OPTION>Sunday

    <OPTION>Monday

    <OPTION>Tuesday

    <OPTION>Wednesday

    <OPTION>Thursday

    <OPTION>Friday

    <OPTION>Saturday

</SELECT>

</TD>

<TR>

<TD>A female relative</TD>

<TD><SELECT NAME="relative" TYPE="TEXT">

```

```

        <OPTION>mother
        <OPTION>sister
        <OPTION>aunt
        <OPTION>grandmother
    </SELECT>

</TD>

<TR>

<TD>An adjective: </TD>

<TD><INPUT NAME="adjective"></TD>

<TR>

<TD>A type of pet: </TD>

<TD><INPUT NAME="pet"></TD>

<TR>

<TD>A piece of furniture: </TD>

<TD><INPUT NAME="furniture"></TD>

<TR>

<TD>A verb ending in "ing": </TD>

<TD><INPUT NAME="ingverb"></TD>

<TR>

<TD>A noun: </TD>

<TD><INPUT NAME="noun"></TD>

<TR>

```

```

<TD>A time of day:</TD>

<TD><SELECT NAME="time" TYPE="TEXT">

    <OPTION>morning

    <OPTION>afternoon

    <OPTION>evening

    <OPTION>night

</SELECT>

</TD>

<TR>

<TD>A present tense action verb: </TD>

<TD><INPUT NAME="actverb"></TD>

<TR>

</TABLE>

```

```

<P>

```

< -- The following "Input" statements are standard types. "Submit" sends the form to the URL listed in the "Form" statement at the beginning of the document. The "Value" statement applies a label to the button. -- >

```

<INPUT TYPE="submit" VALUE="Run the MadLib">

<INPUT TYPE="reset" VALUE="Start Over">

</CENTER>

```

</FORM>

<P><HR><P>

Back to Asilomar Home Page<P>

Questions or comments? Send mail to:

mikec@nps.navy.mil<P>

<i>This page last modified 12 February 1996.</i>

</BODY>

</HTML>

The next step is to compose the CGI program. I will outline the steps I followed using Delphi's Integrated Development Environment (IDE).

- 1) To load the IDE click on the icon that resembles a house with columns
- 2) You should be presented with a new project, if not select "New" from the "File" menu
- 3) On the Component Palette, click on the CGI speed tab, click on the first CGI icon, then click in the form window. The first CGI icon (CGIEnvData) loads general scripting information to your form, the second (CGIDB) adds database specific components which will not be used in this example.
- 4) The CGI element icon should have black boxes in the corners to indicate that it is selected. Look at the Object Inspector window. The 'properties' tab should be

selected. All of the default values are acceptable. If you click on the 'events' tab you should not see any available events.

- 5) Click on a blank spot in the form window to deselect the CGI element. Now let's look at the Object Inspector window. Make sure that the 'properties' tab is selected. For this example, all of the default values are acceptable. These values and their meanings are covered in section three of this tutorial.
- 6) Now, click on the 'events' tab in the object inspector. This is a list of things that will happen when the user initiates certain actions. I want this application to create a form when it is called, so click on the box next to "OnCreate". Now type a name for a procedure which will create a form (I used the name FormCreate in the example. You may use any number of characters, but no spaces). After you type the name, hit 'enter', this will bring up the code editor window. The following code will already have been added for you:

```
-----  
unit Unit1;  
interface  
uses  
    SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  
    Forms, Dialogs, Cgi, DB;  
type  
    TForm1 = class(TForm)  
        CGIEnvData1: TCGIEnvData;  
    procedure FormCreate(Sender: TObject);  
    private  
        { Private declarations }  
    public  
        { Public declarations }  
    end;  
var
```



```
form1: TForm1;
```

```
implementation
```

```
{ $R *.DFM }
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
end;
```

```
end.
```

- 7) The cursor will be after the begin statement. You will now write the code that will a) get data from the HTML form, and b) create the new page using that data.
 - 8) The first step is to declare variables that the Delphi application will use. The variable declaration section comes before the 'begin' statement and after the 'procedure' statement. You will need to declare one variable for each 'INPUT NAME' or 'SELECT NAME' in the HTML document. Note: to avoid confusion, use different variable names than what 'NAMES' were in the HTML document. Since we are reading in text from the form, the variables will be set to type 'string'. My variable declarations section was as follows:
-

```
var
```

```
  dgirl : string;
```

```
  dboy : string;
```

```
  dday : string;
```

```
  drelative : string;
```

```
  dadjective : string;
```

```
  dpet : string;
```

```
  dfurniture : string;
```

```
  dingverb : string;
```

```
  dnoun : string;
```

```
  dtime : string;
```

```
  dactverb : string;
```

- 9) Now move the cursor past the 'begin' statement. Since this procedure uses CGIEnvData, we must add the line: with CGIEnvData1 do The '1' on the end tells Delphi that this will be the first instance of CGIEnvData.
- 10) Now on a new line type: begin This starts the procedure to gather and use the data.
- 11) After this second 'begin' statement three statements must be added which need to be present in ALL Website Delphi applications. They are:

```
webSiteINIFilename := paramstr(1);
application.onException := cgiErrorHandler;
application.processMessages;
```

- 12) The next line to be added: `createStdout;` tells Website that an HTML form will be created.
- 13) The next line is: `sendPrologue;` This statement is accomplished before the creation of the new HTML document and is the beginning of the section where you will gather the user's data. In the following section, you will be setting the Delphi variables you declared earlier, equal to the 'INPUT NAMES' from the form. The format for that is:
`variable := getSmallField( 'INPUT NAME' or 'SELECT NAME' );`

In the example this section was as follows:

```
dgirl := getSmallField( 'girl' );
dboy := getSmallField( 'boy' );
dday := getSmallField( 'day' );
drelative := getSmallField( 'relative' );
dadjective := getSmallField( 'adjective' );
dpet := getSmallField( 'pet' );
dfurniture := getSmallField( 'furniture' );
dingverb := getSmallField( 'ingverb' );
dnoun := getSmallField( 'noun' );
dtime := getSmallField( 'time' );
dactverb := getSmallField( 'actverb' );
```

- 14) Now that you have collected the data, you will create the new HTML document via the 'send' command. The general format is:
`send( '<HTML tag>' + variables + 'regular text' );`

Note that items within the () can be in any order, so long as proper HTML is used. The key here is that variables may be included and are indicated by not being enclosed in quotes. There are a few special 'send' commands such as 'sendHR' and 'sendTitle' that should be self-explanatory.

- 15) The following is a completion of the new HTML document:

```
-----
send( '<HTML><HEAD>' );
sendTitle( 'Here is your custom-made MadLib...' );
send( '</HEAD><BODY>' );
send('<center><h1>Here is your MadLib</h1>');
```

```

send('Thank you for you input.</center>');
send('<p>');
sendHR;

```

```

send('One ' + dday + ' ' + dtime + ' ' + dboy + ' woke up and ');
send('found his ' + drelative + ' ' + dgirl + ' ' + dingverb + ' ');
send('her ' + dpet + ' on the ' + dfurniture + ' . Amazed by ');
send('this affectionate display, ' + dboy + ' asked ');
send(dgirl + ' , "Would you like to ' + dactverb + ' my ');
send(dadjective + ' ' + dnoun + ' when you"re done?");
send( '<P>' );
send('<p>');
sendHR;
send( '<a href="http://131.120.20.70">' );
send( 'Return to the Asilomar Home Page</A>');
send( '<P>' );
send( 'This application was created using Delphi CGI components. ');
send( '<P>' );
send( 'Generated on ' + webdate(now) );
send( '</BODY></HTML>' );

```

- 16) Now that the new form is complete you must tell Website. The next line is:
closeStdout;
- 17) Then: end;
- 18) Now the applications opened just after the 'begin' statement must be closed via
the command: closeApp(application);
- 19) Now another: end; to close the procedure
- 20) And one more: end; to close the entire application
- 21) Now let's save the project. Select File | Save project as
- 22) Delphi will prompt you to enter a name for the code you have just written. This
name must be different than the name of the executable file which Delphi asks
for next. For instance, if you planned to have the executable named "Madlib.exe"
as in this example, you should name the .PAS file something else. I used
"Madlibpr.pas". Now Delphi will ask for your project name. This is the time to
enter the executable program's name.
- 23) You are now ready to compile! From the top menus, select Options | Project.
The 'Project Options' window will open. On the quick tabs, select
'Directories/Conditionals'. The field 'Output Directory' is blank. This field
indicates the directory where you want the compiled executable to be placed.
Click on the down arrow to the right of the field and you will be given a choice of
directories that you have previously entered. 'C:\website\cgi-win' is the directory
for Windows CGI applications, so select that from the list or type it in manually.
Click 'OK'.

- 24) Now from the menu, select Compile | Compile. If you have any errors in your code, they will be highlighted in red. A vaguely helpful message will be displayed at the bottom of the window. Note that semi-colons or the absence of semi-colons is very important. One misplaced semi-colon can cause many errors. Generally spaces are not important to Delphi. Also notice that some words become bold when you type them (i.e. **var**, **procedure**, **begin**, etc.) these words are key words and may not be used elsewhere in your script.
- 25) Once your code has compiled, you are done. Exit Delphi and fire up your web browser. A complete copy of the example code follows on the next two pages.

```

unit Madlibpr;
interface
uses
    SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
    Forms, Dialogs, Cgi, DB;

type
    Tform1 = class(TForm)
        CGIEnvData1: TCGIEnvData;

    procedure FormCreate(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
    var
        form1: Tform1;
implementation
{$R *.DFM}
procedure Tform1.FormCreate(Sender: TObject);
var
    dgirl : string;
    dboy : string;
    dday : string;
    drelative : string;
    dadjective : string;
    dpet : string;
    dfurniture : string;
    dingverb : string;
    dnoun : string;
    dtime : string;
    dactverb : string;

begin
    with CGIEnvData1 do
    begin
        webSiteINIFilename := paramstr(1);
        application.onException := cgiErrorHandler;
        application.processMessages; { need to have this ! }
        CreateStdout;
        sendPrologue;
        dgirl := getSmallField('girl');
    end;
end;

```

```

    dboy := getSmallField( 'boy' );
    dday := getSmallField( 'day' );
    drelative := getSmallField( 'relative' );
    dadjective := getSmallField( 'adjective' );
    dpet := getSmallField( 'pet' );
    dfurniture := getSmallField( 'furniture' );
    dingverb := getSmallField( 'ingverb' );
    dnoun := getSmallField( 'noun' );
    dtime := getSmallField( 'time' );
    dactverb := getSmallField( 'actverb' );
    send( '<HTML><HEAD>' );
    sendTitle( 'Here is your custom-made MadLib...' );
    send( '</HEAD><BODY>' );
    send( '<center><h1>Here is your MadLib</h1>' );
    send( 'Thank you for you input.</center>' );
    send( '<p>' );
    sendHR;

    send( 'One ' + dday + ' ' + dtime + ' ' + dboy + ' woke up and ' );
    send( 'found his ' + drelative + ' ' + dgirl + ' ' + dingverb + ' ' );
    send( 'her ' + dpet + ' on the ' + dfurniture + ' . Amazed by ' );
    send( 'this affectionate display, ' + dboy + ' asked ' );
    send( dgirl + ' , "Would you like to ' + dactverb + ' my ' );
    send( dadjective + ' ' + dnoun + ' when you"re done?"' );
    send( '<p>' );
    sendHR;
    send( '<a href="http://131.120.20.70">' );
    send( 'Return to the Asilomar Home Page</A>' );
    send( '<P>' );
    send( 'This application was created using Delphi CGI components. ' );
    send( '<P>' );
    send( 'Generated on ' + webdate(now) );
    send( '</BODY></HTML>' );

    closeStdout;
    end;
closeApp( application );
end;
end.
-----

```

Delphi may, at first, seem intimidating, even with previous programming experience. Our initial expectations were of a totally "point-and-click" interface, which, as we have seen is not the case. But having worked with several Delphi applications, we have grown accustomed to the format. Many of the statements used in the example you just completed, are common to all CGI scripts. The layout is always the same. First come the variable declarations, then the standard Website statements, then prologue operations, then web page creation. Delphi also offers the power of a built in database which allows seamless compatibility.

The scripting language is a close relative of Pascal. We did not include in this tutorial a discussion of programming techniques, explanations of variable types, and some other minor details. While it would take several hundred pages, all of this information is available in the online help which is quite extensive. Help for the CGI environment elements can be downloaded and incorporated into the Delphi help file. Delphi also has built-in interactive tutors which include a demonstration of how to build an application (non-CGI only) which can be used to increase familiarity with the program environment.

APPENDIX B. APPLICATION TEMPLATES

An Application Template is a tool used in system design to determine which processes should be kept, discontinued, or re-engineered. For the Asilomar Conference, seven major processes were identified:

- 1) Submissions via Postal Mail (Snail Mail),
- 2) Submission via E-mail,
- 3) The existing Asilomar Web page and FTP site,
- 4) The article review process,
- 5) Conference topic area determination,
- 6) Presentations of articles,
- 7) The hard copy proceedings of the Conference.

Each of these processes is rated three different ways: 1) User satisfaction versus strategic value (Fig. B-1), 2) Technical quality versus strategic value (Fig. B-2) and, 3) Technical quality versus technical evolution (Fig. B-3). These three ratings are then totaled for each process and displayed in the Summarized Assessment graph (Fig. B-4). This graph indicates by quadrant which process should be: 1) replaced or discarded, 2) kept and tuned, 3) renovated or re-engineered and, 4) built upon.

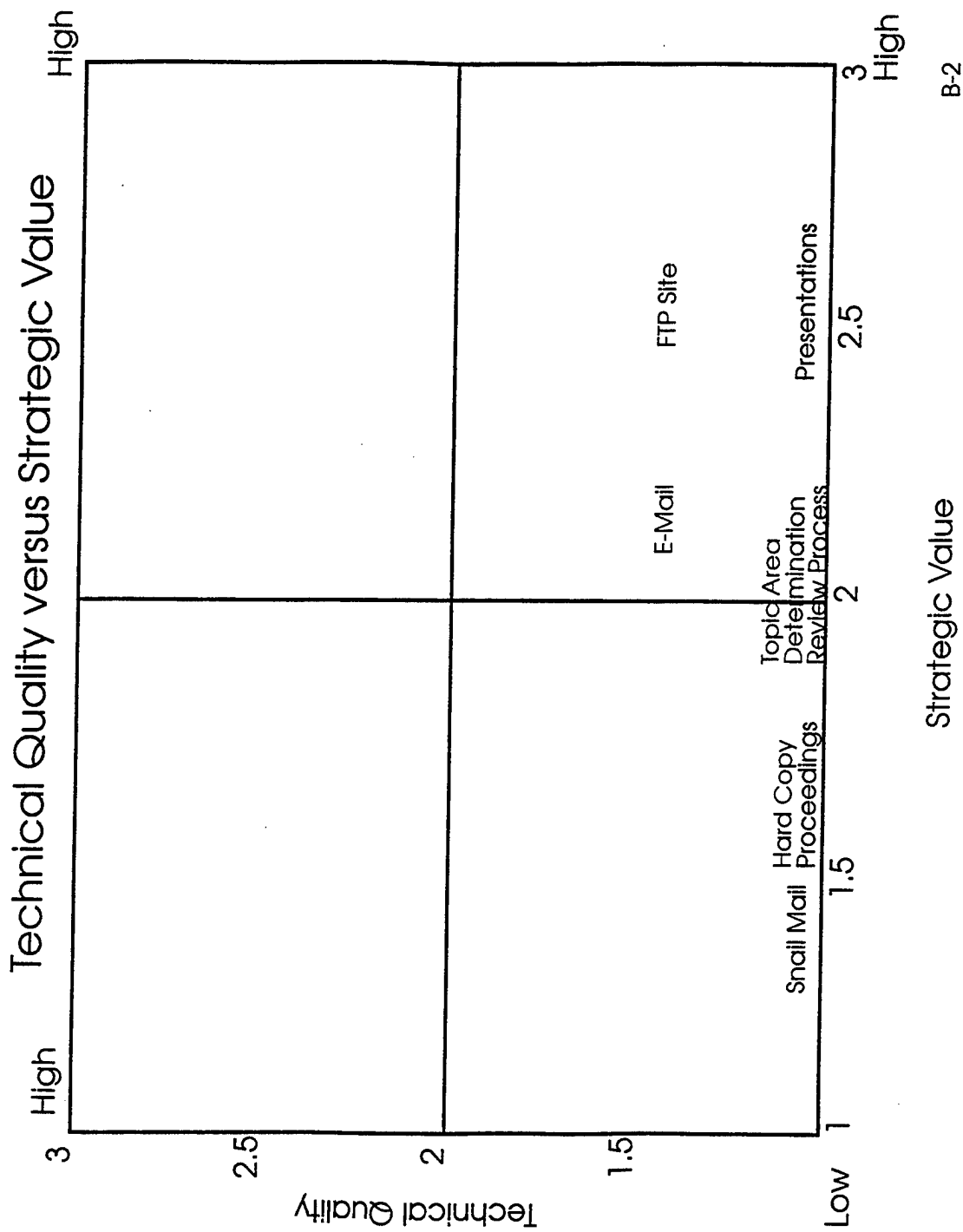


Figure B-1. Technical Quality Versus Strategic Value.

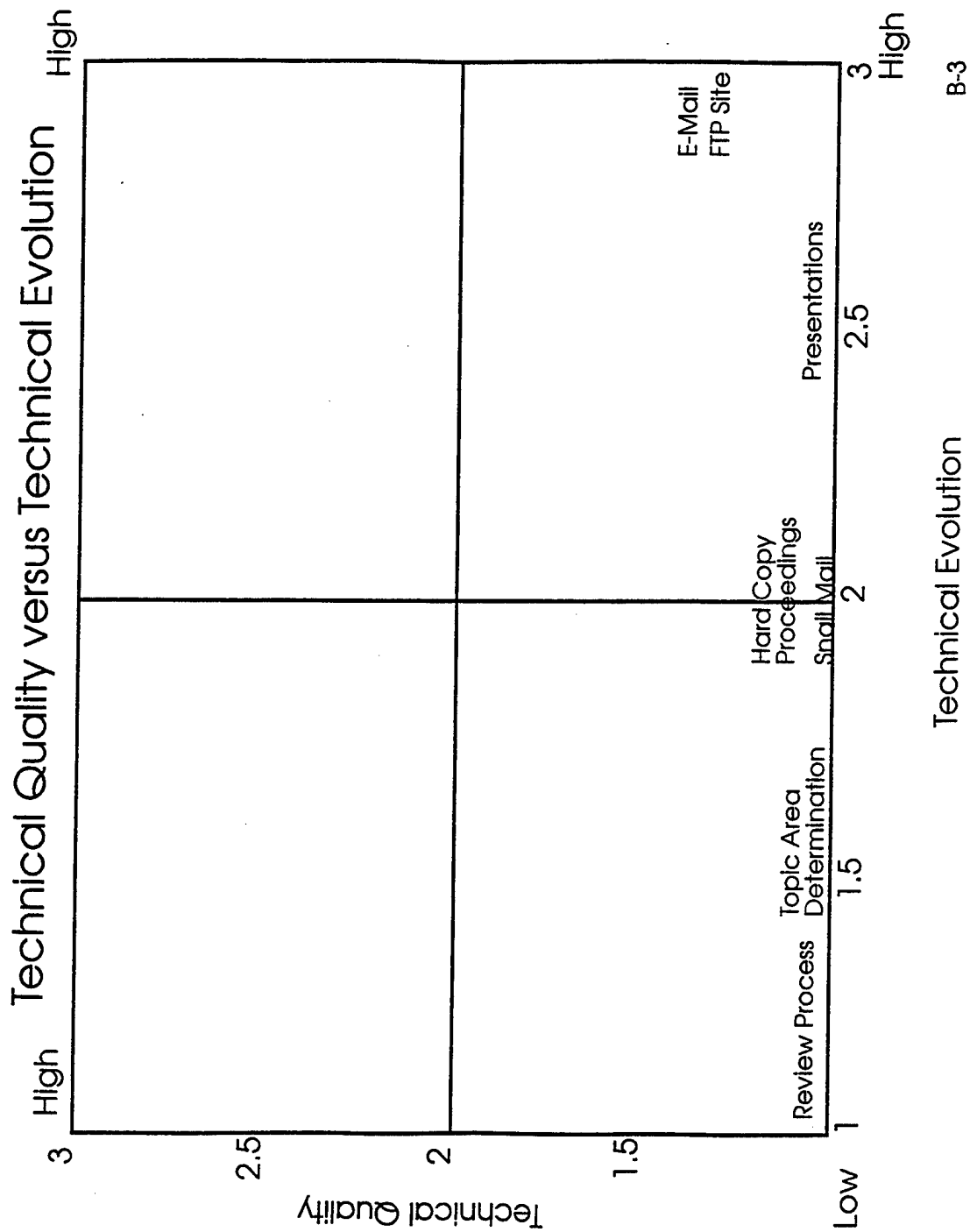


Figure B-2. Technical Quality Versus Technical Evolution.

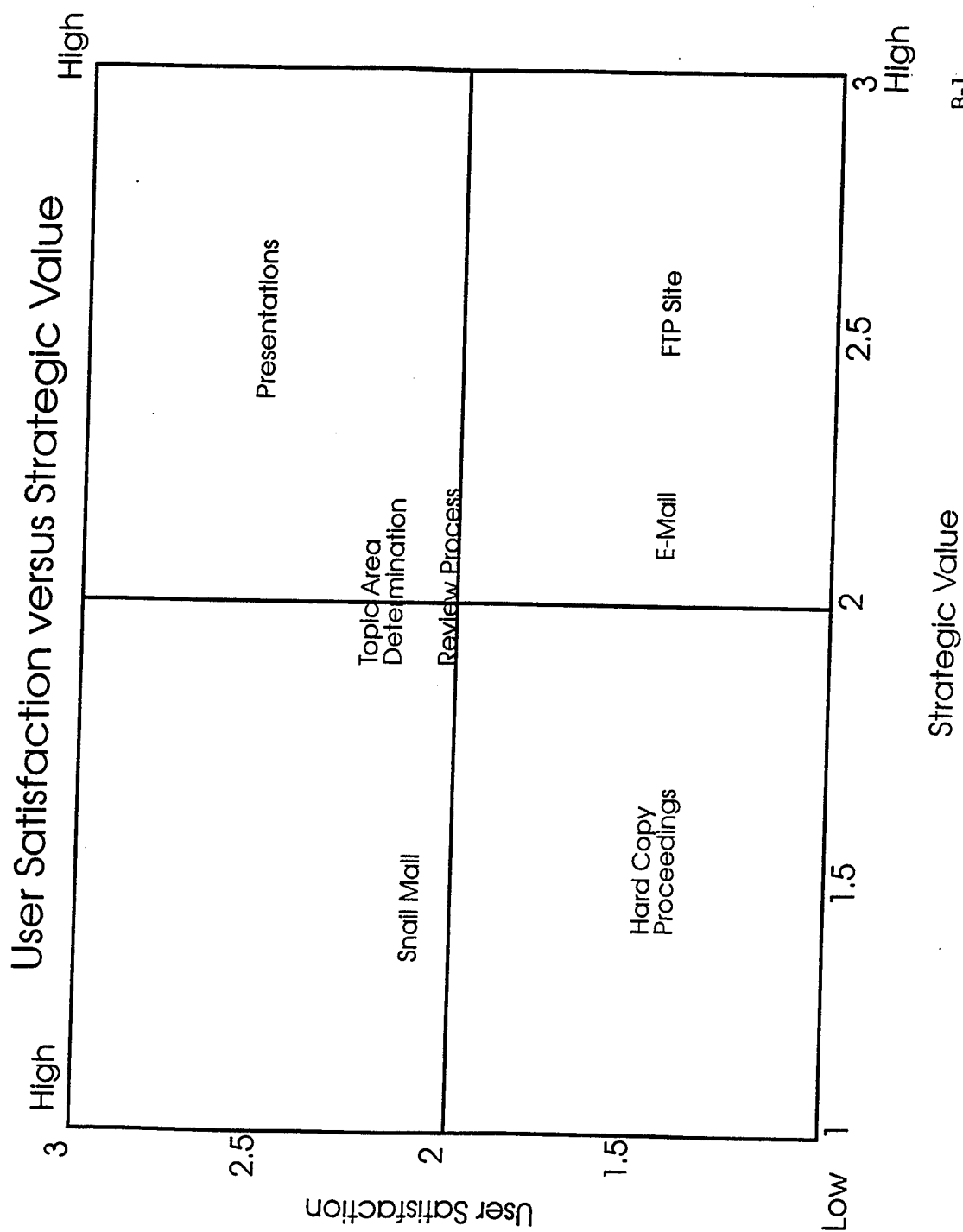


Figure B-3. User Satisfaction Versus Strategic Value.

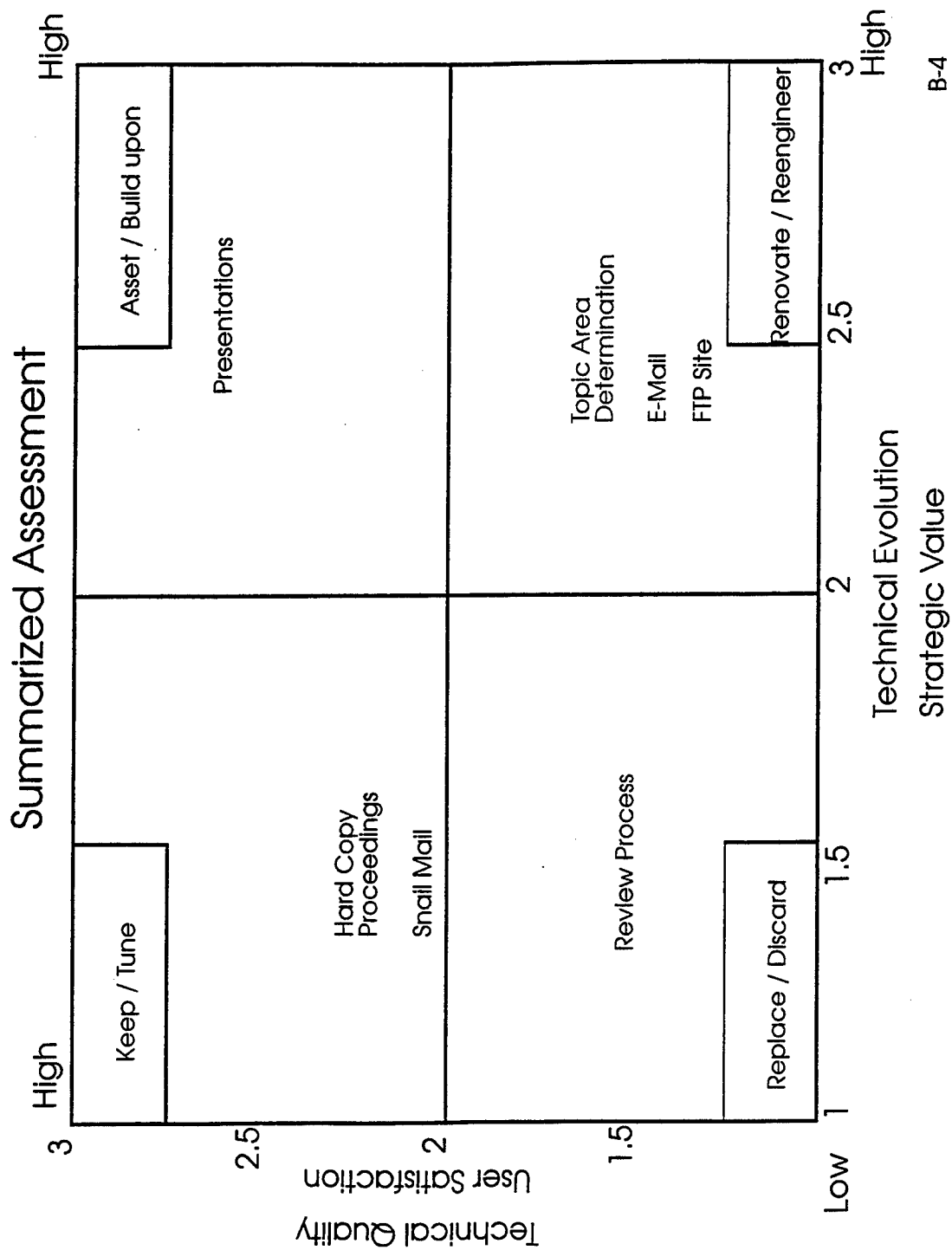


Figure B-4. Summarized Assessment.

APPENDIX C. SALSA DATABASE DESIGN TABLES

Salsa is a shareware database design tool. Users enter information about the data they wish to store, and Salsa will create the necessary relational database tables. For example, in this thesis, it was determined that:

- 1) A Contact Author could submit more than one Paper,
- 2) A Paper could have more than one Author,
- 3) A Paper could have more than one Keyword,
- 4) A Presentation Session could have more than one Paper.

These relationships, along with the format of each data element (last name, address, etc.) were entered into Salsa which generated the database tables. Salsa also automatically generates reports that give database managers a quick reference of the database structure. Figure C-1 shows the titles of each table created, and the labels for each data element within a table. Figures C-2 through C-8 summarize the data format for each data element.

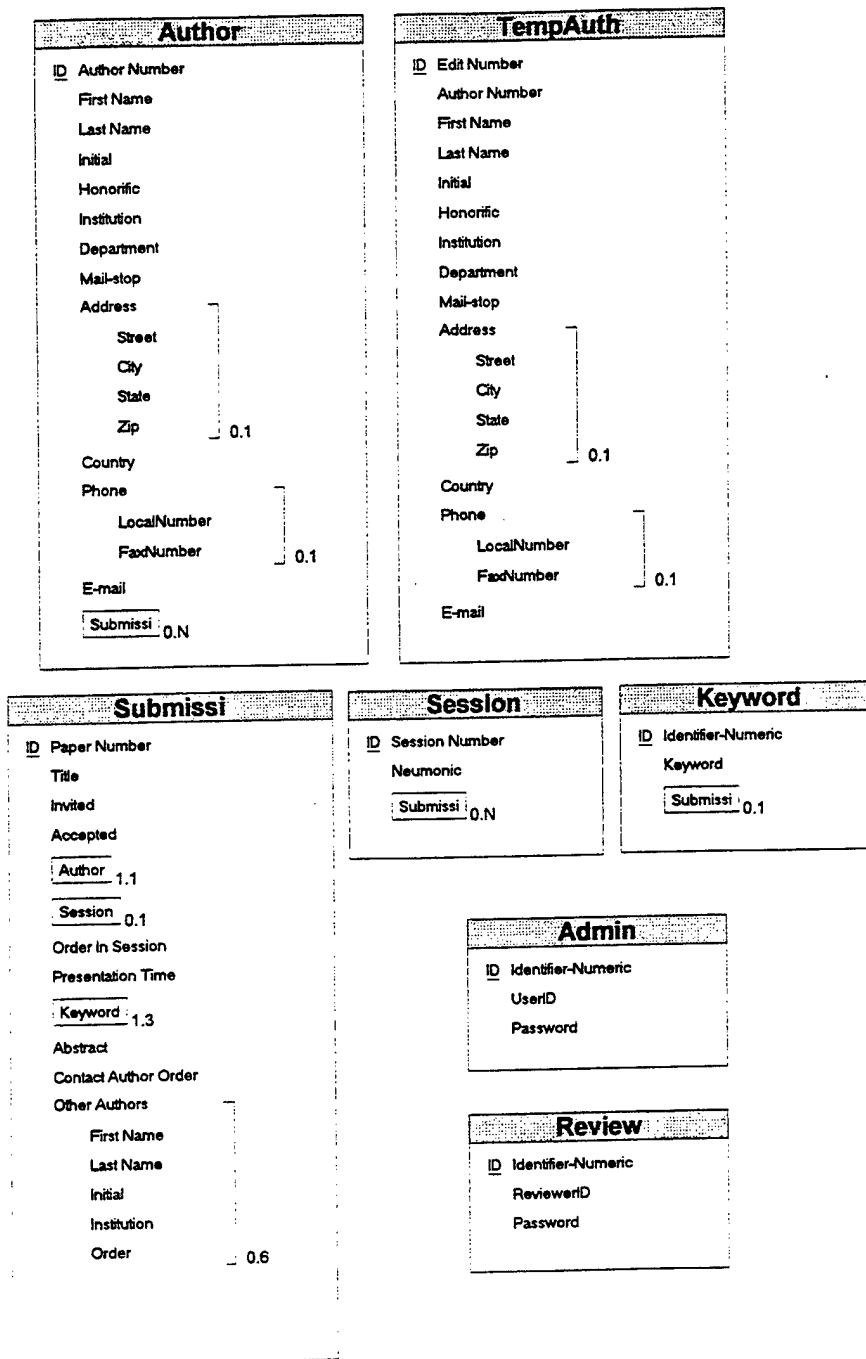


Figure C-1. Database tables.

Semantic Object Report

Album: ASILOMAR.ALB

AUTHOR Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Author Number	Unique	1	1	Long Integer		
First Name	None	0	1	Text	25	
Last Name	None	0	1	Text	25	
Initial	None	0	1	Text	1	
Honorific	None	0	1	Text	5	
Institution	None	0	1	Text	60	
Department	None	0	1	Text	60	
Mail-stop	None	0	1	Text	30	
Address	None	0	1	Group		
Street	None	0	1	Text	40	
City	None	0	1	Text	20	
State	None	0	1	Text	18	
Zip	None	0	1	Text	10	
Country	None	0	1	Text	25	
Phone	None	0	1	Group		
LocalNumber	None	0	1	Text	21	
FaxNumber	None	0	1	Text	21	
E-mail	None	0	1	Text	50	
Submissi	None	0	N (No Limit)	Semantic Object		

Figure C-2. Author.db data elements.

Semantic Object Report

Album: ASILOMAR.ALB

TEMPAUTH Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Edit Number	Unique	1	1	Short Integer		
Author Number	None	0	1	Text	10	
First Name	None	0	1	Text	25	
Last Name	None	0	1	Text	25	
Initial	None	0	1	Text	10	
Honorific	None	0	1	Text	10	
Institution	None	0	1	Text	60	
Department	None	0	1	Text	60	
Mail-stop	None	0	1	Text	30	
Address	None	0	1	Group		
Street	None	0	1	Text	40	
City	None	0	1	Text	20	
State	None	0	1	Text	18	
Zip	None	0	1	Text	10	
Country	None	0	1	Text	25	
Phone	None	0	1	Group		
LocalNumber	None	0	1	Text	21	
FaxNumber	None	0	1	Text	21	
E-mail	None	0	1	Text	50	

Figure C-3. Tempauth.db data elements.

Semantic Object Report

Album: ASILOMAR.ALB

SUBMISSI Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Paper Number	Unique	1	1	Long Integer		
Title	None	0	1	Text	160	
Invited	None	0	1	Text	1	
Accepted	None	0	1	Text	1	
Author	None	1	1	Semantic Object		
Session	None	0	1	Semantic Object		
Order In Session	None	0	1	Text	2	
Presentation Time	None	0	1	Text	7	
Keyword	None	1	3	Semantic Object		
Abstract	None	0	1	Memo		
Contact Author Order	None	0	1	Text	1	
Other Authors	None	0	6	Group		
First Name	None	0	1	Text	25	
Last Name	None	0	1	Text	25	
Initial	None	0	1	Text	10	
Institution	None	0	1	Text	60	
Order	None	0	1	Text	1	

Figure C-4. Submissi.db data elements.

Semantic Object Report

Album: ASILOMAR.ALB

SESSION Semantic Object

Caption:

Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Session Number	Unique	1	1	Long Integer		
Neumonic	None	0	1	Text	5	
Submissi	None	0	N (No Limit)	Semantic Object		

Figure C-5. Session.db data elements.

Semantic Object Report

Album: ASILOMAR.ALB

KEYWORD Semantic Object

Caption:

Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Identifier-Numeric	Unique	1	1	Long Integer		
Keyword	None	0	1	Text	50	
Submissi	None	0	1	Semantic Object		

Figure C-6. Keyword.db data elements.

Semantic Object Report

Album: ASILOMAR.ALB

ADMIN Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Identifier-Numeric	Unique	1	1	Long Integer		
UserID	None	0	1	Text	20	
Password	None	0	1	Text	20	

Figure C-7. Admin.db data elements.

Semantic Object Report

Album: ASILOMAR.ALB

REVIEW Semantic Object

Caption:

Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Identifier-Numeric	Unique	1	1	Long Integer		
ReviewerID	None	0	1	Text	20	
Password	None	0	1	Text	20	

Figure C-8. Review.db data elements.

APPENDIX D. SURVEY RESULTS

1995 Conference Survey results.

A total of 118 surveys were returned. Actual results are included but highlights are as follows:

- Most attendees have WWW access already
- Netscape was by far the browser of choice
- Respondents were very positive about web access to:
 - 1) Retrieving registration materials,
 - 2) Submitting articles,
 - 3) And Previewing selected abstract.
- Respondents were unsure about whether they would send comments to article authors about their articles.

The following pages summarize the data collected from the 1995 survey.

1. What is your age?

3%	4	O under 25 years old	10%	12	O 45 to 54 years old
50%	59	O 25 to 34 years old	9%	11	O 55 to 64 years old
26%	30	O 35 to 44 years old	1%	1	O over 65 years old

2. What is your gender?

88%	101	O Male
12%	14	O Female

3. What is your occupation?

40%	47	O Academic Faculty	15%	18	O Design Engineer
30%	35	O Full-time Student			O Production Engineer
1%	1	O Engineering Management			O Other:
			9%	10	Research Engineer
			2%	2	Post Doc
			1%	1	Scientist
			1%	1	Industrial Engineer
			1%	1	Mathematician
			1%	1	Engineering Services

4. If you are involved with signal processing, what best describes your area of sub-specialty?

10%	15	O Speech Processing	2%	3	O Biomedical Processing
14%	21	O Image Processing	21%	31	O Radar / Sonar / Array Processing
11%	16	O Computer Processing	27%	39	O Digital Processing
12%	18	O Comm. Processing	2%	3	O Networking

5. What is your association with the IEEE?

22%	25	O Not a Member	11%	13	O Senior Member
23%	27	O Student Member	6%	7	O Fellow
37%	43	O Member			

6. How are you participating in this conference?

7%	9	O Attending

83% 105 O Presenting an article
 10% 13 O Session Chair

7. In the preceding 12 months, how often did you attend a professional conference?

16%	19	O None	3%	4	O 6 - 8 times
44%	52	O 1 - 2 times			O 9 - 11 times
36%	43	O 3 - 5 times			O 12 or more times

8. Which word processor do you use most often when writing professional work (articles, technical memos, etc.)?

07%	8	O WordPerfect	0	O Amipro
20%	25	O Microsoft Word	08%	10 O Framemaker
57%	70	O TeX		O Other:
			02%	3 Publisher
			02%	2 Interleaf
			01%	1 Wordstar
			01%	1 TROFF
			01%	1 Scientific word
			01%	1 Windows XP

9. What type of electronic on-line access do you have at this time?

(Due to multiple answers by respondents, percentages are not included)

104	O World-Wide-Web access	O I have local e-mail on a LAN
45	O I have a WWW home page	O I'm not sure
94	O I have Internet FTP access	
1	O I do not have any on-line access at this time	
107	O I have Internet e-mail access	

10. What type of electronic on-line access do you expect to have 12 months from now?

(Due to multiple answers by respondents, percentages are not included.)

96	O World-Wide-Web access	O local e-mail on a LAN
63	O a WWW home page	O I'm not sure
82	O Internet FTP access	
12	O I will not have any on-line access	
94	O Internet e-mail access	

11. If you have World-Wide-Web access, what browser(s) do you use (select all that apply)?

(Due to multiple answers by respondents, percentages are not included.)

99	O Netscape Navigator		O PSINet Pipeline
27	O Spyglass Mosaic	2	O Netcruiser
3	O America Online Browser		O Quarterdeck Internet Suite
	O COMPUERVE Browser	1	O Tradewind WINWeb /
	MACWeb		
	O Prodigy Browser	2	O Microsoft Win95 Browser
	O Cyberjack		O Other:
		4	NCSA Mosaic

12. If you have on--line access, what is the highest data rate you normally use?

3%	3	O < 9600 Bps Modem	4%	5	O 28.8 KBps Modem
3%	3	O 9600 Bps Modem	30%	35	O greater than 28.8 (ISDN, T1,etc)
21%	25	O 14.4 KBps Modem	39%	46	O I access the Internet through a LAN and do not know what data rate I use.

With respect to the professional conferences you attend, please rate how likely you would be to use the following capabilities, if they were available on-line. Assume that you have the required technical capabilities and tools.

	1 Very Unlikely	2 Unlikely	3 Not Sure	4 Likely	5 Very Likely
11. Retrieve Registration forms and information. Avg: 4.30	1 0 10	2 0 3	3 0 3	4 0 25	5 0 74
12. Submit an article abstract to be considered for presentation at the conference. Avg: 4.32	1 0 8	2 0 4	3 0 6	4 0 22	5 0 75
13. Preview article abstracts that have been selected for presentation at the conference. Avg: 4.17	1 0 10	2 0 3	3 0 8	4 0 31	5 0 63
14. Send comments about an article that has been selected for presentation at the conference to the author. Avg: 3.51	1 0 11	2 0 8	3 0 37	4 0 28	5 0 30

APPENDIX E. ARTICLE SUBMISSION SUB-SYSTEM CODE

This appendix contains the HTML and Delphi code for the HTML documents and Delphi executables that comprise the Article Submission Sub-system. The relationships of the programs are depicted in Figure E-1.

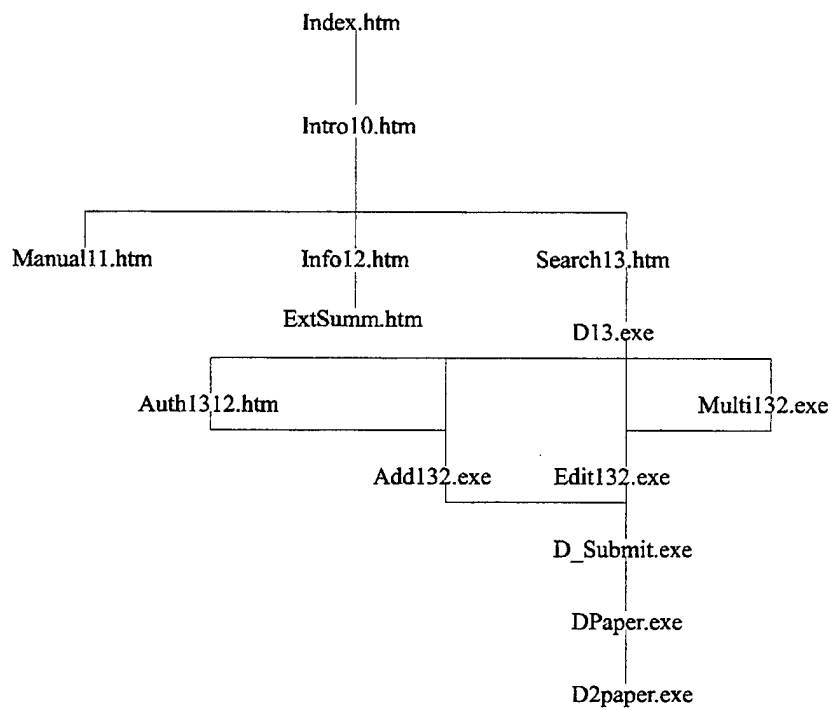


Figure E-1. Article Submission Sub-system.

<HTML>

<HEAD>

<TITLE>Asilomar Signals, Systems, and Computers</TITLE>

</HEAD>

<BODY bgcolor=FFFFFF>

<center><H2>Asilomar Conference on

Signals, Systems, & Computers</center></H2>

<H1>Electronic Submission Information</H1>

<P>Authors wishing to submit an article to the Asilomar
Conference on Signals, Systems, & Computers via on-line
submission not only save the cost of an envelope and a stamp,
they help us keep down the cost of the conference by reducing the
administrative burden on our staff.

<P>What capabilities do I need to submit my Article on-
line?

<P>What are the procedures to submit my Article on-
line?

<P>I am ready to submit my article on-line!

<hr>What capabilities do I need to submit my Article on-line?

<P>Authors wishing to submit an article on-line need the following
capabilities:

Internet access;

Forms capable Web Browser (HTML 3 or better);

FTP upload software

Download Windows 3.x (16-bit) FTP software
NOW!

Download Windows 95 or NT (32-bit) FTP software
 NOW!

<hr>

What are the procedures to submit my Article on-line?

<P>Submitting an article on-line to the Conference
is done in the following steps:

Ensure you have FTP software, download it above if necessary.

Select "Submit an article" at the bottom of this page. The forms presented will allow you to:

Ensure that the information about you is accurate in our database; and

Submit the information about your paper (Title, Keywords, Abstract, and other
Authors);

<i>Note: if you have already received a registration number, user ID and password, the above
steps should be complete.</i>

Finally, submit an Extended Summary for your paper via
FTP.

Extended summaries must be in Post Script (.PS) or ASCII (.txt) format

The filename of your summary should be your registration number; e.g. "101.PS"
would be a valid filename.

<center>

<P><hr>

<h3>I'm ready to submit my Article on-line!</h3>

</center><P><hr>

Return to the Asilomar Conference On-line Submission Page

</BODY>

</HTML>

<HTML>

<HEAD>

<TITLE>Asilomar Signals, Systems, and Computers</TITLE>

</HEAD>

<BODY bgcolor=FFFFFF>

<center><H2>Asilomar Conference on

Signals, Systems, & Computers</center></H2>

<H1>What is the Extended Summary?</H1>

<P>An extended summary is a concise summary of the significance of the submitted paper. It's length should be 500 to 1000 words and can include figures. The extended summary will be used to evaluate the paper for significance and originality.

<P>For electronic submission, the extended summary should be saved in Postscript Format and should include the Paper Title and the Author's name.

<P><hr><P>

Return to the Asilomar
Conference On-line Submission Page

</BODY>

</HTML>

<HTML>

<HEAD>

<TITLE>Asilomar Signals, Systems, and Computers</TITLE>

</HEAD>

<BODY bgcolor=FFFFFF>

<center><H2>Asilomar Conference on

Signals, Systems, & Computers</center></H2>

<H1>Manual Submission Information</H1>

<P>Authors wishing to submit an article to the Asilomar Conference on Signals, Systems, & Computers via mail must submit four copies of the abstract (50 to 100 words) and four copies of the extended summary (500 to 1000 words plus figures and tables). Submissions must include:

Paper Information: title, each author's name and affiliation, and the technical area(s) with number(s) from the list provided on the

 "Call for Papers."

Point of Contact information: name, full address, telephone/fax numbers and e-mail address.

Abstract: on a separate page, a camera ready copy of the abstract headed by the title, author's names and affiliations for printing in an Abstract Book which will be available at the Conference. The type used must be clear and readily legible, and all information (including title, author's names and affiliations) must fit within a 2.75" high by 6.5" wide format for printing in the Abstract Book.

<P> Submissions should be sent to:

Asilomar Conference Technical Program Chairman

c/o ECE Department, Code EC

Naval Postgraduate School

833 Dyer Road, Room 437

Monterey, CA 93943-5121

<h3>In the future</h3>

Please consider submitting your articles via our on-line submission process. Not only does it save you the

cost of an envelope and a stamp, it helps us keep the cost of the conference lower by reducing the administrative burden on our staff.

<P><hr><P>

Return to the Asilomar

Conference On-line Submission Page

</BODY>

</HTML>

<HTML>

<HEAD>

<TITLE>Asilomar Signals, Systems, and Computers</TITLE>

</HEAD>

<BODY bgcolor=FFFFFF>

<center><H2>Asilomar Conference on

Signals, Systems, & Computers</H2></center>

<H1>Contact Author Information</H1>

<P> First we need to get some information about the "contact author." Remember that if the paper has only <emp>one<emp> author, the "contact author" is the author. But if there are <emp>multiple<emp> authors, the "contact author" is the one whom will be the point of contact for the purpose of this conference.

<P> To determine whether you are currently in our database, please enter your name below:

<FORM METHOD="POST" ACTION=" ../cgi-win/D13.exe ">

<HR>

First Name: <INPUT NAME="first" Size="20" TYPE="TEXT">

Last Name: <INPUT NAME="last" Size="20" TYPE="TEXT">

Middle Initial: <INPUT NAME="initial" Size="2" TYPE="TEXT">

<P><CENTER><INPUT TYPE="submit" Value="submit author">

<INPUT TYPE="reset" VALUE="Clear this form"></CENTER>

<P><hr><P>

Return to the Asilomar
Conference On-line Submission Page

</BODY>

</HTML>

The database query page: **D13.exe**

unit D_13;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DB, DBTables, Cglib, Cgi;

type

TForm1 = class(TForm)

Authors: TTable;

DataSource1: TDataSource;

Query1: TQuery;

CGIEnvData1: TCGIEnvData;

CGIDB1: TCGIDB;

Query1FName: TStringField;

Query1LName: TStringField;

Query1ANumber: TIntegerField;

Query1Initial: TStringField;

Query1Honorific: TStringField;

Query1Institution: TStringField;

Query1Department: TStringField;

Query1Mailstop: TStringField;

Query1Address_Street: TStringField;

Query1Address_City: TStringField;

Query1Address_Zip: TStringField;

Query1Country: TStringField;

Query1Phone_LocalNumber: TStringField;

Query1Phone_FaxNumber: TStringField;

Query1Email: TStringField;

Query1Address_State: TStringField;

procedure FormCreate(Sender: TObject);

procedure CGIDB1OnSendingHotField(currentRecord: TDataset;
var s: OpenString);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

```

function MixCaseStr(S: string): string;

implementation

{$R *.DFM}

function MixCaseStr(S: string): string;
{ This function converts a string to Mixed case.
That is Capital first letter and the rest lower case. }
var
    i: integer;

begin

    if (S[1] >= 'a') and (S[1] <= 'z') then
        Dec(S[1], 32);
    for i := 2 to Length(S) do
        if (S[i] >= 'A') and (S[i] <= 'Z') and (S[i-1] <> ' ') and (S[i-1] <> 'e') and (S[i-1] <> '-') and (S[i-1] <> 'c') then
            Inc(S[i], 32);
        MixCaseStr := S;

    end;

procedure TForm1.FormCreate(Sender: TObject);
var
    counter: integer;
    dFNamein: string;
    dLNamein: string;
    dInitialin: string;

    dFName: string;
    dLName: string;
    dInitial: string;

    dbANumber: string;
    dbFName: string;
    dbLName: string;
    dbInitial: string;
    dbHonorific: string;
    dbInstitution: string;
    dbDepartment: string;
    dbMailstop: string;

```



```

dbAddress_Street: string;
dbAddress_City: string;
dbAddress_State: string;
dbAddress_Zip: string;
dbCountry: string;
dbPhone_LocalNumber: string;
dbPhone_FaxNumber: string;
dbEmail: string;

```

```

TC: TDataSet;

```

```

begin

```

```

    with CGIEnvData1 do

```

```

        begin

```

```

            webSiteINIFilename:=paramstr(1);
            application.onException:=cgiErrorHandler;
            application.processMessages;

```

```

            createStdout;
            sendPrologue;

```

```

            dFNamein:= getsmallfield ('first');           {
            dLNamein:= getsmallfield ('last');           { Get fields from search page }
            dInitialin:= getsmallfield ('initial');       {

```

```

            dFName:= MixCaseStr (dFNamein);              { Convert First Name to mixed case }
            dLName:= MixCaseStr (dLNamein);              { Convert Last Name to mixed case }
            if (dInitialin[1] >= 'a') and (dInitialin[1] <= 'z') then begin
                Dec(dInitialin[1], 32);
                dInitial:= dInitialin;                   { Convert Middle Initial to upper case }
            end
            else dInitial:= dInitialin;

```

```

            send ('<HTML><HEAD>');                        {
            SendTitle('Asilomar Signals, Systems, and Computers'); { Every page gets this }
            send ('</HEAD><BODY bgcolor=FFFFFF>');        {

```

```

        end;

```

```

        counter := 0;

```

```

        with Query1 do {Check the database for the author First and Last name}
        begin
            close;

```

```

sql.clear;
if (dLName <> "") and (dFName <> "") then
  sql.add('Select*FROM Author WHERE LName = "' + dLName + '" + 'AND FName = "' +
dFName + '"')
  else if (dFName = "") then sql.add('Select*FROM Author WHERE LName = "' + dLName
+ '"');

```

```

open;
fieldByName('ANumber').displayLabel:='Number';
fieldByName('FName').displayLabel:='First Name';
fieldByName('LName').displayLabel:='Last Name';
fieldByName('Initial').displayLabel:='MI';
fieldByName('Honorific').displayLabel:='Title';
fieldByName('Institution').displayLabel:=' Institution Name ';
fieldByName('Department').displayLabel:=' Department Name ';

```

```

Authors.Open;
Authors.First;

```

```

dbANumber := fieldByName('ANumber').AsString;
dbFName := fieldByName('FName').AsString;
dbLName := fieldByName('LName').AsString;
dbInitial := fieldByName('Initial').AsString;
dbHonorific := fieldByName('Honorific').AsString;
dbInstitution := fieldByName('Institution').AsString;
dbDepartment := fieldByName('Department').AsString;
dbMailstop := fieldByName('Mailstop').AsString;
dbAddress_Street := fieldByName('Address_Street').AsString;
dbAddress_City := fieldByName('Address_City').AsString;
dbAddress_State := fieldByName('Address_State').AsString;
dbAddress_Zip := fieldByName('Address_Zip').AsString;
dbCountry := fieldByName('Country').AsString;
dbPhone_LocalNumber := fieldByName('Phone_LocalNumber').AsString;
dbPhone_FaxNumber := fieldByName('Phone_FaxNumber').AsString;
dbEmail := fieldByName('Email').AsString;

```

```

counter := Recordcount;    {How many records were returned}
Authors.Close;
end;

```

```

with cgiEnvData1 do      {Output the results of the query}
begin
  if counter = 0 then begin    {No records returned}

    send('<center><H1>Add New Author Information</H1></center>');

```

```

send ('<P> Your name was not found in our current database. Please provide');
send (' the following information so we will be able to contact you.<HR>');

```

```

send('<P><FORM ACTION="add132.exe" METHOD="POST">');
send('<B>First Name:  </B><INPUT NAME="FName" Size="25" TYPE="text"
VALUE="'+ dFName +' ">');
send('<B> Last Name:  </B><INPUT NAME="LName" Size="25" TYPE="text"
Value="'+ dLName +' ">');
send('<B> Middle Initial: </B><INPUT NAME="Initial" Size="2" TYPE="text" Value="'+
dInitial +' "><HR>');
send('<B>Honorific Title:</B><INPUT NAME="Honorific" Size="5" TYPE="text">');
send('<P><B>Institution:  </B><INPUT NAME="Institution" Size="60"
TYPE="text">');
send('<br><B>Department:  </B><INPUT NAME="Department" Size="60"
TYPE="text">');
send('<br><B>Mail Stop:  </B><INPUT NAME="Mailstop" Size="30"
TYPE="text">');
send('<P><B>Street address: </B><INPUT NAME="Address_Street" Size="40"
TYPE="text">');
send('<br><B>City:      </B><INPUT NAME="Address_City" Size="20"
TYPE="text">');
send('<B>State:      </B><INPUT NAME="Address_State" Size="18" TYPE="text">');
send('<br><B>Zip Code:  </B><INPUT NAME="Address_Zip" Size="10"
TYPE="text">');
send('<B>Country:    </B><INPUT NAME="Country" Size="25" TYPE="text"
VALUE="USA ">');
send('<P><B>Telephone number (with Area/Country codes): </B><INPUT
NAME="Phone_LocalNumber" Size="21" TYPE="text">');
send('<br><B>Fax number (with Area/Country codes):  </B><INPUT
NAME="Phone_Faxnumber" Size="21" TYPE="text">');
send('<br><B>E-mail:    </B><INPUT NAME="Email" Size="50" TYPE="text">');

send('<P><CENTER><INPUT TYPE="submit" Value="Submit New Information">
<INPUT TYPE="reset">');
send('VALUE="Clear this form"></CENTER>');
end

```

```

else if counter = 1 then begin   {Only one record returned}

```

```

send('<center><H1>Edit Author Information</H1></center>');
send ('<P>We have located your record in the database. Please ensure that');
send (' your record is up-to-date by reviewing the information below and');
send (' making any changes that are necessary. For your protection, changes are not');
send (' made a part of the permanent record until verified by our staff.<BR><HR>');

```

```

send('<P><FORM ACTION="edit132.exe" METHOD="POST">');
send('<INPUT NAME="ANumber" TYPE="hidden" Value="" + dbANumber + ">');
send('<B>First Name: </B><INPUT NAME="FName" Size="25" TYPE="text"
VALUE="" + dbFName + ">');
send('<B> Last Name: </B><INPUT NAME="LName" Size="25" TYPE="text"
Value="" + dbLName + ">');
send('<B> Middle Initial: </B><INPUT NAME="Initial" Size="2" TYPE="text" Value=""
+ dbInitial + "><HR>');
send('<B>Honorific Title:</B><INPUT NAME="Honorific" Size="5" TYPE="text"
Value="" + dbHonorific + ">');
send('<P><B>Institution: </B><INPUT NAME="Institution" Size="60" TYPE="text";
send(' Value="" + dbInstitution + ">');
send('<br><B>Department: </B><INPUT NAME="Department" Size="60"
TYPE="text" Value="" + dbDepartment + ">');
send('<br><B>Mail Stop: </B><INPUT NAME="Mailstop" Size="30" TYPE="text"
Value="" + dbMailstop + ">');
send('<P><B>Street address: </B><INPUT NAME="Address_Street" Size="40"
TYPE="text";
send('Value="" + dbAddress_Street + ">');
send('<br><B>City: </B><INPUT NAME="Address_City" Size="20" TYPE="text"
Value="" + dbAddress_City + ">');
send('<B>State: </B><INPUT NAME="Address_State" Size="2" TYPE="text"
Value="" + dbAddress_State + ">');
send('<br><B>Zip Code: </B><INPUT NAME="Address_Zip" Size="10"
TYPE="text" Value="" + dbAddress_Zip + ">');
send('<B>Country: </B><INPUT NAME="Country" Size="25" TYPE="text"
Value="" + dbCountry + "> ');
send('<P><B>Telephone number (with Area/Country codes):</B><INPUT
NAME="Phone_LocalNumber" Size="21";
send(' TYPE="text"Value="" + dbPhone_LocalNumber + ">');
send('<br><B>Fax number (with Area/Country codes): </B><INPUT
NAME="Phone_Faxnumber" Size="21";
send(' TYPE="text"Value="" + dbPhone_FaxNumber + ">');
send('<br><B>E-mail: </B><INPUT NAME="Email" Size="50" TYPE="text"
Value="" + dbEmail + ">');

```

```

send('<p>The database has found a unique entry based on the first and last');
send(' name you submitted. If you believe this entry refers to another ');
send('person, then you have two choices:');
send('<ul><li>If you might have used another name, then <a
href=" ../prototype/search13.htm">');
send('Go back and re-submit your query;</a> or');
send('<li>If this is your name but this is not your record, then <a href=" ../cgi-
win/auth132.exe">');
send('Add your name to our records.</a></ul>');

```

```

    send('<P>If this entry does refer to you, please review the information and update it as
necessary, ');
    send('then click on "Submit Information" below<BR>');
    send('<P><CENTER><INPUT TYPE="submit" Value="Submit Information"> <INPUT
TYPE="reset">');
    send('VALUE="Reset to Original Information"></CENTER></form>');

    send('<P><FORM ACTION="d_submit.exe" METHOD="POST">');
    send ('<INPUT TYPE="hidden" Name="ANumber" Value="" + dbANumber +"">');
    send('<P><CENTER><INPUT TYPE="submit" Value="NO changes necessary - ALL
data is correct!"></CENTER></form>');

```

end

else begin {more than one record returned}

```

    send ('<center><H2>Asilomar Conference on<BR>');
    send ('Signals, Systems, & Computers</center></H2>');
    send('<P><HR><Center>');
    CGIDB1.drawTable;
    Query1.close;
    send ('</center><p>');
    send('The database has found more than one entry that matches your first');
    send(' and last name. If your name is on the list then click on the "Author Number" which
is');
    send(' to the right of your name. ');
    send('<p>If you believe your record may be under another name,');
    send(' you may <a href=" ../prototype/search13.htm">Resubmit your name</a>');
    send('<p>If this is your name but your record does not appear here,');
    send(' you may <a href=" ../cgi-win/auth132.exe">Add your name to our records.</a>');
    end;

```

{ HTML page Footer }

```

    send('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar
facility">');
    send('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
    send('Conference On-line Submission Page</A>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

```

end; {with cgiEnvData1 do}

end; {Procedure TForm1.FormCreate}

procedure TForm1.CGIDB1OnSendingHotField(currentRecord: TDataset;

```
var s: OpenString);  
  
begin  
  s:='<A HREF="../cgi-win/mult132.exe?'+ currentrecord.fieldbyName('ANumber').AsString +  
  '>' + currentrecord.FieldbyName('ANumber').AsString + '</A>';  
  
end; {Procedure TForm1.CGIONSendingHotField}  
  
end.
```

Add New Author Information page: **Add132.exe**

unit D132add;

{ WHEN CALLED: This program is activated from D13.exe when the author name is not found in the AUTHOR Table ** OR ** from auth1312.htm when adding a new instance of the author name (a new person with the same name)}.

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, Cgldb, Cgi, DBTables, DB;

type

TForm1 = class(TForm)
 DataSource1: TDataSource;
 Authors: TTable;
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;

 procedure FormCreate(Sender: TObject);

private

 { Private declarations }

public

 { Public declarations }

end;

var

 Form1: TForm1;

 dANumber: string;

 dFName: string;

 dLName: string;

 dInitial: string;

 dFNamein: string;

 dLNamein: string;

 dInitialin: string;

 dHonorific: string;

 dInstitution: string;

```

dDepartment: string;
dMailstop: string;
dAddress_Street: string;
dAddress_City: string;
dAddress_State: string;
dAddress_Zip: string;
dCountry: string;
dPhone_AreaCode: string;
dPhone_LocalNumber: string;
dPhone_FaxNumber: string;
dEmail: string;

function MixCaseStr(S: string): string;

```

implementation

```

{$R *.DFM}

{*****}
function MixCaseStr(S: string): string;
{ This function converts a string to Mixed case.
That is Capital first letter and the rest lower case.  }
var
  i: integer;

begin

  if (S[1] >= 'a') and (S[1] <= 'z') then
    Dec(S[1], 32);
  for i := 2 to Length(S) do
    if (S[i] >= 'A') and (S[i] <= 'Z') and (S[i-1] <> '-') and (S[i-1] <> '-') and (S[i-1] <> 'c') then
      Inc(S[i], 32);
    MixCaseStr := S;

end;
{*****}

procedure TForm1.FormCreate(Sender: TObject);
begin

  with CGIEnvData1 do begin

    {Standard Header}
    webSiteINIFilename:=paramstr(1);

```



```

application.onException:=cgiErrorHandler;
application.processMessages;
createStdout;
sendPrologue;

```

{Get data from FORM}

```

dFNamein:= getsmallfield ('FName');
dLNamein:= getsmallfield ('LName');
dInitialin:= getsmallfield ('Initial');

dFName:= MixCaseStr (dFNamein);           { Convert First Name to mixed case }
dLName:= MixCaseStr (dLNamein);           { Convert Last Name to mixed case }
if (dInitialin[1] >= 'a') and (dInitialin[1] <= 'z') then begin
  Dec(dInitialin[1], 32);
  dInitial:= dInitialin;                   { Convert Middle Initial to upper case }
end
else dInitial:= dInitialin;

```

```

dHonorific:= getsmallfield ('Honorific');
dInstitution:= getsmallfield ('Institution');
dDepartment:= getsmallfield ('Department');
dMailstop:= getsmallfield ('Mailstop');
dAddress_Street:= getsmallfield ('Address_Street');
dAddress_City:= getsmallfield ('Address_City');
dAddress_State:= getsmallfield ('Address_State');
dAddress_Zip:= getsmallfield ('Address_Zip');
dCountry:= getsmallfield ('Country');
dPhone_LocalNumber:= getsmallfield ('Phone_LocalNumber');
dPhone_FaxNumber:= getsmallfield ('Phone_Faxnumber');
dEmail:= getsmallfield ('Email');

```

{Append data to AUTHOR Table}

```

Authors.open;
Authors.AppendRecord([nil, dFName, dLName, dInitial, dHonorific, dInstitution,
dDepartment, dMailstop,
dAddress_Street, dAddress_City, dAddress_State, dAddress_Zip, dCountry,
dPhone_LocalNumber, dPhone_FaxNumber, dEmail]);
Authors.close;

```

{Get the Author number for the new record }

```

Authors.open;
Authors.last;
dANumber:=Authors.fieldbyname('ANumber').asString;
Authors.close;

```

```

{Send the Thankyou Paper page with the link to ADD Paper132.htm  }
send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
send ('<center><H2>Asilomar Conference on<BR>');
send ('Signals, Systems, & Computers</H2>');
send ('<P><HR>');

send ('<P>Thank you for your input ' + dHonorific + ' ' + dFName + ' ' + dLName + '. ');
send ('Your information has been added to our records. ');

send ('<FORM ACTION="../../cgi-win/d_submit.exe" METHOD="POST">');
send ('<INPUT TYPE="hidden" Name="ANumber" Value="" + dANumber + "">');
send ('<P><INPUT TYPE="submit" Value="Continue with the Submission Process">');
send ('</CENTER>');
send ('<P><hr><P><IMG SRC="../../prototype/asil2b0.gif" align=left alt="Asilomar
facility">');
send ('Return to the <A HREF="http://131.120.20.70/index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');

closeStdout;
closeApp( application ); { don't leave form around }

end;  {with CGIEnvData1}

end;  {FormCreate}

end.

```

Edit Author information page: **Edit132.exe**

unit D132edit;

{ WHEN CALLED: This program is activated from D13.exe when the author name is found in the AUTHOR Table. }

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type

TForm1 = class(TForm)
 DataSource1: TDataSource;
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 TempAuthor: TTable;

 procedure FormCreate(Sender: TObject);

private

 { Private declarations }

public

 { Public declarations }

end;

var

Form1: TForm1;

dANumber: string;
dANumberint: integer;
dFName: string;
dLName: string;
dInitial: string;
dHonorific: string;
dInstitution: string;
dDepartment: string;
dMailstop: string;
dAddress_Street: string;
dAddress_City: string;
dAddress_State: string;

```

dAddress_Zip: string;
dCountry: string;
dPhone_AreaCode: string;
dPhone_LocalNumber: string;
dPhone_FaxNumber: string;
dEmail: string;

```

implementation

```
{ $R *.DFM }
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
    with CGIEnvData1 do
    begin
```

```
        {Standard Header}
```

```
        webSiteINIFilename:=paramstr(1);
```

```
        application.onException:=cgiErrorHandler;
```

```
        application.processMessages;
```

```
        createStdout;
```

```
        sendPrologue;
```

```
        dANumber:= getsmallfield ('ANumber');
```

```
        dFName:= getsmallfield ('FName');
```

```
        dLName:= getsmallfield ('LName');
```

```
        dInitial:= getsmallfield ('Initial');
```

```
        dHonorific:= getsmallfield ('Honorific');
```

```
        dInstitution:= getsmallfield ('Institution');
```

```
        dDepartment:= getsmallfield ('Department');
```

```
        dMailstop:= getsmallfield ('Mailstop');
```

```
        dAddress_Street:= getsmallfield ('Address_Street');
```

```
        dAddress_City:= getsmallfield ('Address_City');
```

```
        dAddress_State:= getsmallfield ('Address_State');
```

```
        dAddress_Zip:= getsmallfield ('Address_Zip');
```

```
        dCountry:= getsmallfield ('Country');
```

```
        dPhone_LocalNumber:= getsmallfield ('Phone_LocalNumber');
```

```
        dPhone_FaxNumber:= getsmallfield ('Phone_Faxnumber');
```

```
        dEmail:= getsmallfield ('Email');
```

```
        dANumberint:= StrToInt(dANumber);    { Convert the dANumber back to an integer }
```

with TempAuthor do begin

{Open TempAuthor table and append new record }

open;

AppendRecord([nil, dANumberint, dFName, dLName, dInitial, dHonorific, dInstitution,
dDepartment, dMailstop,
dAddress_Street, dAddress_City, dAddress_State, dAddress_Zip, dCountry,
dPhone_LocalNumber, dPhone_FaxNumber, dEmail]);

close;

end; {with TempAuthor} .

{Send the Thankyou Paper page with link to ADD Paper132.htm }

send ('<HTML><HEAD>');

SendTitle('Asilomar Signals, Systems, and Computers');

send ('</HEAD><BODY bgcolor=FFFFFF>');

send ('<center><H2>Asilomar Conference on
');

send ('Signals, Systems, & Computers</H2>');

send ('<P><HR>');

send ('<P>Thank you for your input ' + dHonorific + ' ' + dFName + ' ' + dLName + ' ');

send ('Your information has been stored in order to update our records.');

send (' To protect your record, changes will not become permanent until verified by our
staff.');

send ('<FORM ACTION=" ../cgi-win/d_submit.exe" METHOD="POST">');

send ('<INPUT TYPE="hidden" Name="ANumber" Value="" + dANumber + "">');

send ('<P><INPUT TYPE="submit" Value="Continue with the Submission Process">');

send ('</CENTER>');

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar
facility">');

send ('Return to the Asilomar');

send ('Conference On-line Submission Page');

send ('</BODY></HTML>');

closeStdout;

closeApp(application); { don't leave form around }

end; {with CGIEnvData1}

end; {FormCreate}

end.

More than one Author's name found: **Multi132.exe**

unit D132mult;

{ WHEN CALLED: This program is activated from D13.exe when more than one author name match is found in the AUTHOR Table. }

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DB, DBTables, Cgi, Cgidb;

type

TForm1 = class(TForm)
 CGIDB1: TCGIDB;
 CGIEnvData1: TCGIEnvData;
 Authors: TTable;
 DataSource1: TDataSource;
 procedure FormCreate(Sender: TObject);

private

 { Private declarations }

public

 { Public declarations }

end;

var

 Form1: TForm1;

implementation

{ \$R *.DFM }

procedure TForm1.FormCreate(Sender: TObject);

var

 dANumber: string;
 dbFName: string;
 dbLName: string;
 dbInitial: string;
 dbHonorific: string;
 dbInstitution: string;
 dbDepartment: string;

```

dbMailstop: string;
dbAddress_Street: string;
dbAddress_City: string;
dbAddress_State: string;
dbAddress_Zip: string;
dbCountry: string;
dbPhone_LocalNumber: string;
dbPhone_FaxNumber: string;
dbEmail: string;

TC: TDataSet;

begin {TFormCreate}

with CGIEnvData1 do begin

    {Standard Header}
    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
    createStdout;
    sendPrologue;

    { Get the query string from the address}
    dANumber := CGIQueryString^;

    { Get the record information }
    with Authors do begin

        { Move to proper record}
        open;
        first;
        while fieldByName('ANumber').asString <> dANumber do
            next;

        { Retrieve record }
        dbFName := fieldByName('FName').AsString;
        dbLName := fieldByName('LName').AsString;
        dbInitial := fieldByName('Initial').AsString;
        dbHonorific := fieldByName('Honorific').AsString;
        dbInstitution := fieldByName('Institution').AsString;
        dbDepartment := fieldByName('Department').AsString;
        dbMailstop := fieldByName('Mailstop').AsString;
        dbAddress_Street := fieldByName('Address_Street').AsString;
        dbAddress_City := fieldByName('Address_City').AsString;

```

```

dbAddress_State := fieldName('Address_State').AsString;
dbAddress_Zip := fieldName('Address_Zip').AsString;
dbCountry := fieldName('Country').AsString;
dbPhone_LocalNumber := fieldName('Phone_LocalNumber').AsString;
dbPhone_FaxNumber := fieldName('Phone_FaxNumber').AsString;
dbEmail := fieldName('Email').AsString;
Close;
end; {With Authors}

```

{Create new page}

```

send('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send('</HEAD><BODY bgcolor=FFFFFF>');

send('<center><H1>Edit Author Information</H1></center>');
send('<P>We have located your record in the database. Please ensure that');
send(' your record is up-to-date by reviewing the information below and');
send(' making any changes that are necessary.<BR><HR>');

send('<P><FORM ACTION="edit132.exe" METHOD="POST">');
send('<INPUT NAME="ANumber" TYPE="hidden" Value="'+ dANumber +'>');
send('<B>First Name: </B><INPUT NAME="FName" Size="25" TYPE="text"');
VALUE="'+ dbFName +'>');
send('<B> Last Name: </B><INPUT NAME="LName" Size="25" TYPE="text"');
Value="'+ dbLName +'>');
send('<B> Middle Initial: </B><INPUT NAME="Initial" Size="2" TYPE="text" Value="'+
+ dbInitial +'><HR>');
send('<B>Honorific Title:</B><INPUT NAME="Honorific" Size="5" TYPE="text"');
Value="'+ dbHonorific +'>');
send('<P><B>Institution: </B><INPUT NAME="Institution" Size="60" TYPE="text">');
send(' Value="'+ dbInstitution +'>');
send('<br><B>Department: </B><INPUT NAME="Department" Size="60"');
TYPE="text" Value="'+ dbDepartment +'>');
send('<br><B>Mail Stop: </B><INPUT NAME="Mailstop" Size="30" TYPE="text"');
Value="'+ dbMailstop +'>');
send('<P><B>Street address: </B><INPUT NAME="Address_Street" Size="40"');
TYPE="text">');
send('Value="'+ dbAddress_Street +'>');
send('<br><B>City: </B><INPUT NAME="Address_City" Size="20" TYPE="text"');
Value="'+ dbAddress_City +'>');
send('<B>State: </B><INPUT NAME="Address_State" Size="18" TYPE="text"');
Value="'+ dbAddress_State +'>');
send('<br><B>Zip Code: </B><INPUT NAME="Address_Zip" Size="10"

```



```

TYPE="text" Value="" + dbAddress_Zip + ">');
    send('<B>Country:    </B><INPUT NAME="Country" Size="25" TYPE="text"
Value="" + dbCountry + "> ');
    send('<P><B>Telephone number (with Area/Country codes):    </B><INPUT
NAME="Phone_LocalNumber" Size="21">');
    send(' TYPE="text"Value="" + dbPhone_LocalNumber + ">');
    send('<br><B>Fax number (with Area/Country codes):    </B><INPUT
NAME="Phone_Faxnumber" Size="21">');
    send(' TYPE="text"Value="" + dbPhone_FaxNumber + ">');
    send('<br><B>E-mail:    </B><INPUT NAME="Email" Size="50" TYPE="text"
Value="" + dbEmail + ">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Submit Information"> <INPUT
TYPE="reset">');
    send('VALUE="Reset to Original Information"></CENTER>');

    send('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar
facility">');
    send('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
    send('Conference On-line Submission Page</A>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

    end; {with cgiEnvData1 do}

end; {Procedure TForm1.FormCreate}

end.

```

```
unit Dsubmit;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    Query1: TQuery;
    Keywords: TTable;
    Query1Name: TStringField;
    KeywordsNumber: TIntegerField;
    KeywordsName: TStringField;
    Query1Number: TIntegerField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  dbANumber: string;
  dbOption: string;
  i: integer;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do begin
```

```

webSiteINIFilename:=paramstr(1);
application.onException:=cgiErrorHandler;
application.processMessages;

```

```

createStdout;
sendPrologue;

```

```

dbANumber:= getsmallfield('ANumber');
send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
send ('<center><H2>Asilomar Conference on<BR>');
send ('Signals, Systems, & Computers</center></H2>');
send('<P><HR>');

```

```

send('<H1>New Article Submission</H1>');
send('<P> Please provide the following information about the article.');
```

```

send('<P><FORM METHOD="POST" ACTION=" ../cgi-win/d_paper.exe ">');
send('<INPUT TYPE="hidden" Name="ANumber" Value="'+ dbANumber +' ">');
send('<B>Title: </B><INPUT NAME="title" Size="80" TYPE="text">');
```

{Display Keywords on pull-down menu by taking values from the KEYWORD Table}

{Pull Keywords from KEYWORD Table }

```

with Query1 do begin
  close;
  sql.clear;
  sql.add('Select * FROM Keyword ');
  open;

```

```

{Move records from query result to Keyword #1 Pull-down menu}
Keywords.open;
Keywords.first;

```

```

send('<P><B>Keyword #1: </B><SELECT NAME="Keyword1">');
```

```

while not Keywords.EOF do begin
  dbOption:= Keywords.fieldByName('Name').AsString;
  send('<OPTION> ' +dbOption+ ');
  Keywords.next;
end; {for all records in the query result}
send('</SELECT>');
```

{Move records from query result to Keyword #2 Pull-down menu}

```
Keywords.first;
send('<P><B>Keyword #2: </B><SELECT NAME="Keyword2">');
send('<OPTION> None');
```

```
while not Keywords.EOF do begin
  dbOption:= Keywords.fieldByName('Name').AsString;
  send('<OPTION> ' +dbOption+ ');
  Keywords.next;
end; {for all records in the query result}
send('</SELECT>');
```

{Move records from query result to Keyword #3 Pull-down menu}
Keywords.first;

```
send('<P><B>Keyword #3: </B><SELECT NAME="Keyword3">');
send('<OPTION> None');
```

```
while not Keywords.EOF do begin
  dbOption:= Keywords.fieldByName('Name').AsString;
  send('<OPTION> ' +dbOption+ ');
  Keywords.next;
end; {for all records in the query result}
send('</SELECT>');
end; {with Query1}
```

{Also end displaying Keyword pull-down menus}

```
send('<P><B>Abstract: <p></B><TEXTAREA NAME="abstract" COLS=70 Rows=10>');
send('Please enter your abstract here by either');
send('typing or copying over this text.</TEXTAREA>');
```

```
send('<P><h1>Other Author Information</h1>If there are multiple authors for this article, ');
send('please enter the following information. ');
send('In the boxes marked "Name Position" please enter the position in which you wish ');
send('each author"s name to appear when the abstract is printed.');
```

```
send('<br><B>Position of the Contact Author"s Name:</B>');
send('<Select NAME="contactOrder" ><option> 1<option> 2<option> 3<option> 4<option> 5<option> 6</select>');
```

```
sendhr;
send('<center><h2>Other Authors</h2></center><hr>');
send('<B>Name Position: </B><Select NAME="order2" ><option> 1<option selected> 2');
send('<option> 3<option> 4<option> 5<option> 6</select>');
send('<B>First Name: </B><INPUT NAME="fname2" Size="25" TYPE="text">');
send('<B>Middle Initial: </B><INPUT NAME="initial2" Size="1" TYPE="text">');
send('<BR><B>Last Name: </B><INPUT NAME="lname2" Size="25" TYPE="text">');
```

```

send ('<br><B>Institution:  </B><INPUT NAME="institution2" Size="60" TYPE="text">');

sendhr;
send ('<B>Name Position: </B><Select NAME="order3" ><option> 1<option> 2');
send ('<option selected> 3<option> 4<option> 5<option> 6</select>');
send ('<B>First Name:  </B><INPUT NAME="fname3" Size="25" TYPE="text">');
send ('<B>Middle Initial: </B><INPUT NAME="initial3" Size="1" TYPE="text">');
send ('<BR><B>Last Name:  </B><INPUT NAME="lname3" Size="25" TYPE="text">');
send ('<BR><B>Institution:  </B><INPUT NAME="institution3" Size="60"
TYPE="text">');

```

```

sendhr;
send ('<B>Name Position: </B><Select NAME="order4" ><option> 1<option> 2<option>');
send (' 3<option selected> 4<option> 5<option> 6</select>');
send ('<B>First Name:  </B><INPUT NAME="fname4" Size="25" TYPE="text">');
send ('<B>Middle Initial: </B><INPUT NAME="initial4" Size="1" TYPE="text">');
send ('<BR><B>Last Name:  </B><INPUT NAME="lname4" Size="25" TYPE="text">');
send ('<BR><B>Institution:  </B><INPUT NAME="institution4" Size="60"
TYPE="text">');

```

```

sendhr;
send ('<B>Name Position: </B><Select NAME="order5" ><option> 1<option> 2');
send ('<option> 3<option> 4<option selected> 5<option> 6</select>');
send ('<B>First Name:  </B><INPUT NAME="fname5" Size="25" TYPE="text">');
send ('<B>Middle Initial: </B><INPUT NAME="initial5" Size="1" TYPE="text">');
send ('<BR><B>Last Name: </B><INPUT NAME="lname5" Size="25" TYPE="text">');
send ('<BR><B>Institution:  </B><INPUT NAME="institution5" Size="60"
TYPE="text">');

```

```

sendhr;
send ('<B>Name Position: </B><Select NAME="order6" ><option> 1<option> 2');
send ('<option> 3<option> 4<option> 5<option selected> 6</select>');
send ('<B>First Name:  </B><INPUT NAME="fname6" Size="25" TYPE="text">');
send ('<B>Middle Initial: </B><INPUT NAME="initial6" Size="1" TYPE="text">');
send ('<BR><B>Last Name: </B><INPUT NAME="lname6" Size="25" TYPE="text">');
send ('<BR><B>Institution:  </B><INPUT NAME="institution6" Size="60"
TYPE="text">');

```

```

send('<P><CENTER><INPUT TYPE="submit" Value="Submit Paper Information"> ');
send('<INPUT TYPE="reset" VALUE="Clear this form"></CENTER>');

```

```

send('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar
facility">');
send('Return to the <A HREF=" ../index.html">Asilomar');
send('Conference Home Page</A>');

```

```
send('</BODY></HTML>');

send('</BODY></HTML>');
closeStdout;
closeApp( application ); { don't leave form around }
end; {with CGIEnvData1}

end; {FormCreate}

end.
```

Article information confirmation page: **Dpaper.exe**

unit Dpaper;

{ WHEN CALLED: This program is activated upon the Submit Paper Information. }

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type

TForm1 = class(TForm)
 CGIDB1: TCGIDB;
 CGIEnvData1: TCGIEnvData;
 DataSource1: TDataSource;
 Authors: TTable;

 procedure FormCreate(Sender: TObject);

private

 { Private declarations }

public

 { Public declarations }

end;

var

 Form1: TForm1;

 theabstract : TStringList;

 i: integer;

 dANumber: string;

 dANumberint: integer;

 dFName: string;

 dLName: string;

 dInitial: string;

 dHonorific: string;

 dtitle: string;

 dkeyword1: string;

 dkeyword2: string;

dkeyword3: string;
dabstract: string;
dcontactOrder: string;

dfname2, dfname2in: string;
dlname2, dlname2in: string;
dinitial2: string;
dinstitution2: string;
dorder2: string;

dfname3, dfname3in: string;
dlname3, dlname3in: string;
dinitial3: string;
dinstitution3: string;
dorder3: string;

dfname4, dfname4in: string;
dlname4, dlname4in: string;
dinitial4: string;
dinstitution4: string;
dorder4: string;

dfname5, dfname5in: string;
dlname5, dlname5in: string;
dinitial5: string;
dinstitution5: string;
dorder5: string;

dfname6, dfname6in: string;
dlname6, dlname6in: string;
dinitial6: string;
dinstitution6: string;
dorder6: string;

dbFName : string;
dbLName : string;
dbInitial : string;
dbHonorific : string;

function MixCaseStr(S: string): string;

implementation

{ \$R *.DFM }


```

{*****}
function MixCaseStr(S: string): string;
{ This function converts a string to Mixed case.
  That is Capital first letter and the rest lower case. }
var
  i: integer;

begin

  if (S[1] >= 'a') and (S[1] <= 'z') then
    Dec(S[1], 32);
  for i := 2 to Length(S) do
    if (S[i] >= 'A') and (S[i] <= 'Z') and (S[i-1] <> ' ') and (S[i-1] <> '-') and (S[i-1] <> 'c') then
      Inc(S[i], 32);
    MixCaseStr := S;

end;
{*****}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do
    begin

      webSiteINIFilename:=paramstr(1);
      application.onException:=cgiErrorHandler;
      application.processMessages;

      createStdout;
      sendPrologue;

      { Get the fields from the FORM }

      dANumber:= getsmallfield ('ANumber');
      dANumberint:= strtoint (dANumber);
      dtitle:= getsmallfield ('title');
      dkeyword1:= getsmallfield ('keyword1');
      dkeyword2:= getsmallfield ('keyword2');
      dkeyword3:= getsmallfield ('keyword3');

      theabstract := TStringList.create;
      CGIEnvData1.getTextArea( 'abstract', theabstract );

      dcontactOrder:= getsmallfield ('contactOrder');

```

```

{get data on second author position}
dfname2in:= getsmallfield ('fname2');
dlname2in:= getsmallfield ('lname2');

if (dfname2in = "") and (dlname2in = "") then dorder2:= "    {Remove the number if there is no
name}
    else begin

        {get the rest of the info}
        dorder2:= getsmallfield ('order2');
        dinitial2:= getsmallfield ('initial2');
        dinstitution2:= getsmallfield ('institution2');

        {Format the Names}
        dfname2:= MixCaseStr (dfname2in);
        dlname2:= MixCaseStr (dlname2in)
        end; {if}

{get data on third author position}
dfname3in:= getsmallfield ('fname3');
dlname3in:= getsmallfield ('lname3');

if (dfname3in = "") and (dlname3in = "") then dorder3:= "    {Remove the number if there is no
name}
    else begin

        {get the rest of the info}
        dorder3:= getsmallfield ('order3');
        dinitial3:= getsmallfield ('initial3');
        dinstitution3:= getsmallfield ('institution3');

        {Format the Names}
        dfname3:= MixCaseStr (dfname3in);
        dlname3:= MixCaseStr (dlname3in)
        end; {if}

{get data on fourth author position}
dfname4in:= getsmallfield ('fname4');
dlname4in:= getsmallfield ('lname4');

if (dfname4in = "") and (dlname4in = "") then dorder4:= "    {Remove the number if there is
no name}
    else begin

```

```

{get the rest of the info}
dorder4:= getsmallfield ('order4');
dinitial4:= getsmallfield ('initial4');
dinstitution4:= getsmallfield ('institution4');

```

```

{Format the Names}
dfname4:= MixCaseStr (dfname4in);
dlname4:= MixCaseStr (dlname4in)
end; {if}

```

```

{get data on fifth author position}
dfname5in:= getsmallfield ('fname5');
dlname5in:= getsmallfield ('lname5');

```

```

if (dfname5in = "") and (dlname5in = "") then dorder5:= "    {Remove the number if there is no
name}
else begin

```

```

{get the rest of the info}
dorder5:= getsmallfield ('order5');
dinitial5:= getsmallfield ('initial5');
dinstitution5:= getsmallfield ('institution5');

```

```

{Format the Names}
dfname5:= MixCaseStr (dfname5in);
dlname5:= MixCaseStr (dlname5in)
end; {if}

```

```

{get data on sixth author position}
dfname6in:= getsmallfield ('fname6');
dlname6in:= getsmallfield ('lname6');

```

```

if (dfname6in = "") and (dlname6in = "") then dorder6:= "    {Remove the number if there is no
name}
else begin

```

```

{get the rest of the info}
dorder6:= getsmallfield ('order6');
dinitial6:= getsmallfield ('initial6');
dinstitution6:= getsmallfield ('institution6');

```

```

{Format the Names}
dfname6:= MixCaseStr (dfname6in);

```

```

    dlname6:= MixCaseStr (dlname6in)
    end; {if}

    {if there are no other authors, then do not place a number in contactAuthorNumber}
    if (dorder2 = "") and (dorder3 = "") and (dorder4 = "") and (dorder5 = "") and (dorder6 = "") then
dcontactOrder:= "";

{ Get Author Name information for next HTML page }
    with Authors do begin

        { Move to proper record }
        open;
        first;
        while fieldByName('ANumber').asString <> dANumber do
            next;

        { Retrieve Name information }
        edit;
        dbFName := fieldByName('FName').AsString;
        dbLName := fieldByName('LName').AsString;
        dbInitial := fieldByName('Initial').AsString;
        dbHonorific := fieldByName('Honorific').AsString;
        close;
        end; {With Authors}

{ Send the Confirm html page }
        send ('<HTML><HEAD>');
        SendTitle('Asilomar Signals, Systems, and Computers');
        send ('</HEAD><BODY bgcolor=FFFFFF>');
        send ('<center><H2>Asilomar Conference on<BR>');
        send ('Signals, Systems, & Computers</H2>');
        send ('<P><HR>');

        send ('<P>Thank you for your input ' + dbHonorific + ' ' + dbFName + ' ' + dbLName + '. ');
        send ('Please confirm that the following information is correct before it is added to our
records. ');
        send ('<strong>This information can not be changed after it is submitted this
time!</strong></center>');

        send ('<P><FORM METHOD="POST" ACTION=" ../cgi-win/d2paper.exe ">');
        send ('<INPUT TYPE="hidden" Name="ANumber" Value="" + dANumber + ">');
        send ('<B>Title: </B><INPUT NAME="title" Size="80" TYPE="text" Value="" + dtitle
+ ">');

```

```

send ('<P><B>Keyword #1: </B><INPUT NAME="keyword1" SIZE="50" TYPE="TEXT"
VALUE="" + dkeyword1 + ">');
send ('<P><B>Keyword #2: </B><INPUT NAME="keyword2" SIZE="50" TYPE="TEXT"
VALUE="" + dkeyword2 + ">');
send ('<P><B>Keyword #3: </B><INPUT NAME="keyword3" SIZE="50" TYPE="TEXT"
VALUE="" + dkeyword3 + ">');

send ('<P><B>Abstract: </B><p><TEXTAREA NAME="abstract" COLS=70 Rows=10>');
for i := 0 to theabstract.count - 1 do
    send( theabstract.strings[i] );
send ('</TEXTAREA>');

send ('<P><h1>Other Author Information</h1>If there are multiple authors for this article, ');
send ('please enter the following information. It is assumed that you are the "Contact
Autohor" for');
send ('this paper. In the boxes marked "Name Position" please enter the position in which
you wish ');
send ('each authors name to appear when the abstracts are printed. ');
send ('<br><B>Position of YOUR Name:</B>');
send ('<INPUT NAME="contactOrder" Size="2" TYPE="text" Value="" + dcontactOrder
+ ">');

sendhr;
send ('<B>Name Position: </B><INPUT NAME="order2" Size="2" TYPE="text" Value="" +
dorder2 + ">');
send ('<B>First Name: </B><INPUT NAME="fname2" Size="25" TYPE="text" Value="" +
dfname2 + ">');
send ('<B>Middle Initial: </B><INPUT NAME="initial2" Size="1" TYPE="text" Value="" +
dinitial2 + ">');
send ('<BR><B>Last Name: </B><INPUT NAME="lname2" Size="25" TYPE="text"
Value="" + dlname2 + ">');
send ('<BR><B>Institution: </B><INPUT NAME="institution2" Size="60" TYPE="text"
Value="" + dinstitution2 + ">');

sendhr;
send ('<B>Name Position: </B><INPUT NAME="order3" Size="2" TYPE="text" Value="" +
dorder3 + ">');
send ('<B>First Name: </B><INPUT NAME="fname3" Size="25" TYPE="text" Value="" +
dfname3 + ">');
send ('<B>Middle Initial: </B><INPUT NAME="initial3" Size="1" TYPE="text" Value="" +
dinitial3 + ">');
send ('<BR><B>Last Name: </B><INPUT NAME="lname3" Size="25" TYPE="text"
Value="" + dlname3 + ">');
send ('<br><B>Institution: </B><INPUT NAME="institution3" Size="60" TYPE="text"
Value="" + dinstitution3 + ">');

```

```

sendhr;
send ('<B>Name Position: </B><INPUT NAME="order4" Size="2" TYPE="text" Value="" +
dorder4 +"">');
send ('<B>First Name: </B><INPUT NAME="fname4" Size="25" TYPE="text" Value="" +
dfname4 +"">');
send ('<B>Middle Initial: </B><INPUT NAME="initial4" Size="1" TYPE="text" Value="" +
dinitial4 +"">');
send ('<BR><B>Last Name: </B><INPUT NAME="lname4" Size="25" TYPE="text"
Value="" + dlname4 +"">');
send ('<BR><B>Institution: </B><INPUT NAME="institution4" Size="60" TYPE="text"
Value="" + dinstitution4 +"">');

```

```

sendhr;
send ('<B>Name Position: </B><INPUT NAME="order5" Size="2" TYPE="text" Value="" +
dorder5 +"">');
send ('<B>First Name: </B><INPUT NAME="fname5" Size="25" TYPE="text" Value="" +
dfname5 +"">');
send ('<B>Middle Initial: </B><INPUT NAME="initial5" Size="1" TYPE="text" Value="" +
dinitial5 +"">');
send ('<BR><B>Last Name: </B><INPUT NAME="lname5" Size="25" TYPE="text"
Value="" + dlname5 +"">');
send ('<BR><B>Institution: </B><INPUT NAME="institution5" Size="60" TYPE="text"
Value="" + dinstitution5 +"">');

```

```

sendhr;
send ('<B>Name Position: </B><INPUT NAME="order6" Size="2" TYPE="text" Value="" +
dorder6 +"">');
send ('<B>First Name: </B><INPUT NAME="fname6" Size="25" TYPE="text" Value="" +
dfname6 +"">');
send ('<B>Middle Initial: </B><INPUT NAME="initial6" Size="1" TYPE="text" Value="" +
dinitial6 +"">');
send ('<BR><B>Last Name: </B><INPUT NAME="lname6" Size="25" TYPE="text"
Value="" + dlname6 +"">');
send ('<BR><B>Institution: </B><INPUT NAME="institution6" Size="60" TYPE="text"
Value="" + dinstitution6 +"">');

```

```

sendhr;
send ('<P><CENTER><INPUT TYPE="submit" Value="Confirm Submission
Information"><INPUT TYPE="reset">');
send (' VALUE="Clear this form"></CENTER>');

```

```

send('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar
facility">');
send('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send('Conference On-line Submission Page</A>');

```

```
send ('</BODY></HTML>');  
closeStdout;  
closeApp( application ); { don't leave form around }  
  
end; {with CGIEnvData1}  
  
end; {with FormCreate}  
  
end. {Application}
```

Confirmation page: **D2paper.exe**

unit D2_paper;

{ WHEN CALLED: This program is activated upon the Submission Information Confirmation. }

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, Cgldb, Cgi, DB, DBTables, Starsock, Inetmail;

type

TForm1 = class(TForm)

Articles: TTable;

DataSource1: TDataSource;

CGIEnvData1: TCGIEnvData;

CGIDB1: TCGIDB;

Authors: TTable;

procedure FormCreate(Sender: TObject);

{ procedure SendMail(DataSet: TDataSet); }

{ procedure onmailinfo(Sender: TObject; info: MailInfo; addinfo: String); }

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

theabstract : TStringList;

dANumber: string;

dANumberint: integer;

dtitle: string;

dkeyword1: string;

dkeyword2: string;

dkeyword3: string;

dabstract: string;

dcontactOrder: string;

dfname2in: string;
dlname2in: string;
dfname2: string;
dlname2: string;
dinitial2: string;
dinstitution2: string;
dorder2: string;

dfname3in: string;
dlname3in: string;
dfname3: string;
dlname3: string;
dinitial3: string;
dinstitution3: string;
dorder3: string;

dfname4in: string;
dlname4in: string;
dfname4: string;
dlname4: string;
dinitial4: string;
dinstitution4: string;
dorder4: string;

dfname5in: string;
dlname5in: string;
dfname5: string;
dlname5: string;
dinitial5: string;
dinstitution5: string;
dorder5: string;

dfname6in: string;
dlname6in: string;
dfname6: string;
dlname6: string;
dinitial6: string;
dinstitution6: string;
dorder6: string;

dPaper_Number: string;

dbFName : string;

```

dbLName : string;
dbHonorific : string;
dbEmail : string;

addinfo: string;

function MixCaseStr(S: string): string;

implementation

{$R *.DFM}

{*****}
function MixCaseStr(S: string): string;
{ This function converts a string to Mixed case.
That is Capital first letter and the rest lower case. }
var
  i: integer;

begin

  if (S[1] >= 'a') and (S[1] <= 'z') then
    Dec(S[1], 32);
  for i := 2 to Length(S) do
    if (S[i] >= 'A') and (S[i] <= 'Z') and (S[i-1] <> ' ') and (S[i-1] <> '-') and (S[i-1] <> 'c') then
      Inc(S[i], 32);
  MixCaseStr := S;

end;
{*****}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do
    begin

      {Initialize HTML page }

      webSiteINIFilename:=paramstr(1);
      application.onException:=cgiErrorHandler;
      application.processMessages;

      createStdout;
      sendPrologue;

```

```

{ Get data }
dANumber:= getsmallfield ('ANumber');
dANumberint:= strtoint (dANumber);
dtitle:= getsmallfield ('title');
dkeyword1:= getsmallfield ('keyword1');
dkeyword2:= getsmallfield ('keyword2');
dkeyword3:= getsmallfield ('keyword3');

theabstract := TStringList.create;
CGIEnvData1.getTextArea( 'abstract', theabstract);

dcontactOrder:= getsmallfield ('contactOrder');

{get data on second author position}
dfname2in:= getsmallfield ('fname2');
dlname2in:= getsmallfield ('lname2');

if (dfname2in = "") and (dlname2in = "") then dorder2:= "    {Remove the number if there is no
name}
    else begin

        {get the rest of the info}
        dorder2:= getsmallfield ('order2');
        dinitial2:= getsmallfield ('initial2');
        dinstitution2:= getsmallfield ('institution2');

        {Format the Names}
        dfname2:= MixCaseStr (dfname2in);
        dlname2:= MixCaseStr (dlname2in)
        end; {if}

{get data on third author position}
dfname3in:= getsmallfield ('fname3');
dlname3in:= getsmallfield ('lname3');

if (dfname3in = "") and (dlname3in = "") then dorder3:= "    {Remove the number if there is no
name}
    else begin

        {get the rest of the info}
        dorder3:= getsmallfield ('order3');
        dinitial3:= getsmallfield ('initial3');
        dinstitution3:= getsmallfield ('institution3');

```

```

    {Format the Names}
    dfname3:= MixCaseStr (dfname3in);
    dlname3:= MixCaseStr (dlname3in)
    end; {if}

{get data on fourth author position}
dfname4in:= getsmallfield ('fname4');
dlname4in:= getsmallfield ('lname4');

if (dfname4in = "") and (dlname4in = "") then dorder4:= "    {Remove the number if there is
no name}
    else begin

        {get the rest of the info}
        dorder4:= getsmallfield ('order4');
        dinitial4:= getsmallfield ('initial4');
        dinstitution4:= getsmallfield ('institution4');

        {Format the Names}
        dfname4:= MixCaseStr (dfname4in);
        dlname4:= MixCaseStr (dlname4in)
        end; {if}

{get data on fifth author position}
dfname5in:= getsmallfield ('fname5');
dlname5in:= getsmallfield ('lname5');

if (dfname5in = "") and (dlname5in = "") then dorder5:= "    {Remove the number if there is no
name}
    else begin

        {get the rest of the info}
        dorder5:= getsmallfield ('order5');
        dinitial5:= getsmallfield ('initial5');
        dinstitution5:= getsmallfield ('institution5');

        {Format the Names}
        dfname5:= MixCaseStr (dfname5in);
        dlname5:= MixCaseStr (dlname5in)
        end; {if}

{get data on sixth author position}

```

```

dfname6in:= getsmallfield ('fname6');
dlname6in:= getsmallfield ('lname6');

if (dfname6in = "") and (dlname6in = "") then dorder6:= "    {Remove the number if there is no
name}
    else begin

        {get the rest of the info}
        dorder6:= getsmallfield ('order6');
        dinitial6:= getsmallfield ('initial6');
        dinstitution6:= getsmallfield ('institution6');

        {Format the Names}
        dfname6:= MixCaseStr (dfname6in);
        dlname6:= MixCaseStr (dlname6in);
        end; {if}

    {if there are no other authors, then do not place a number in contactAuthorNumber}
    if (dorder2 = "") and (dorder3 = "") and (dorder4 = "")and (dorder5 = "")and (dorder6 = "") then
dcontactOrder:= "";

{ Place Data into SUBMISSION Table}
with Articles do
    begin

        {Append new submission data to SUBMISSION Table}
        open;
        AppendRecord([nil, dtitle, nil, nil, dANumberint, nil, nil, nil, dkeyword1, dkeyword2,
dkeyword3,
            nil, dcontactOrder, dFName2, dLName2, dInitial2, dInstitution2, dOrder2, dFName3,
dLName3,
            dInitial3, dInstitution3, dOrder3, dFName4, dLName4, dInitial4, dInstitution4,
dOrder4,
            dFName5, dLName5, dInitial5, dInstitution5, dOrder5, dFName6, dLName6,
dInitial6, dInstitution6,
            dOrder6]);

        edit;
        CGIDB1.StringlistToMemo(theabstract, fieldbyname('Abstract'));
        theabstract.free;

        {Get the Paper_Number for presentation as the Registration #.}
        last;
        dPaper_Number:=fieldbyname('Paper_Number').asString;
        close;

```

```

    end; {with Articles}

{ Get Name Information for next html page }
with Authors do begin

    { Move to proper record }
    open;
    first;
    while fieldByName('ANumber').asString <> dANumber do
        next;

    { Get Name Data }
    edit;
    dbFName := fieldByName('FName').AsString;
    dbLName := fieldByName('LName').AsString;
    dbHonorific := fieldByName('Honorific').AsString;
    dbEmail := fieldByName('Email').AsString;
    close;
    end; {With Authors}

{
    SendMail1.TO_Address:= dbEmail;
    SendMail1.TO_Name:= dbHonorific + ' ' + dbFName + ' ' + dbLName;
    SendMail1.action:=Send_Mail;}

{ Send the Final page }
send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
send ('<center><H2>Asilomar Conference on<BR>');
send ('Signals, Systems, & Computers</center></H2>');
sendhr;
sendhdr ('1', 'Confirm New Article Submission');
send ('<P>Thank you ');
send (dbHonorific + ' ' + dbFName + ' ' + dbLName + '. ');

send ('<P> Your article information has been added to the conference');
send ('records. <b>Your Registration Number is: '+dPaper_Number+'<br></b>');

send ('<i>Be sure that you write this number down</i> or print out a copy of this page. ');
send ('You will need this number when you send in your extended summary. ');
sendhdr ('3','The filename for your extended summary will be the registration number!');
send ('For example, if your Registration Number is 15, the file which contains your summary
should be named ');

```

```

send ('<B>15.ps</B> if it's a postscript file, or <B>15.txt</B> if it's an ASCII file.');
```

send ('<P>You must now FTP your extended summary. You may use your own');
 send ('FTP software or the program you down-loaded from the <A>');
 send ('HREF="http://131.120.20.70/prototype/proc122.htm">Procedures Page.');
 send ('Be sure you include the Title of the Article and the Author names');
 send (' in the text of the summary.');

send ('<P>Log on to our FTP Server using the following information:
');
 send ('server address: 131.120.20.70
 ');
 send ('userID: guest
 ');
 send ('password: asilconf
');

send ('<P>Thank you for using our online submission process. We hope');
 send ('that you found it helpful. Please send any comments to');
 send ('mikec@nps.navy.mil');

send('<P><hr><P><IMG SRC="../../prototype/asil2b0.gif" align=left alt="Asilomar
 facility">');
 send('Return to the Asilomar');
 send('Conference On-line Submission Page');
 send ('</BODY></HTML>');

CGIEnvData1.closeStdout;
 closeApp(application);

end; {with CGIEnvData1}

end; {FormCreate}

```

{procedure TForm1.SendMail(DataSet: TDataset);

var

    dbFName : string;
    dbLName : string;
    dbHonorific : string;
    dbEmail : string;

begin

    with Authors do begin
        open;
        first;
        edit;
    
```

```

    moveby(dANumberint-1);
    dbFName := fieldByName('FName').AsString;
    dbLName := fieldByName('LName').AsString;
    dbHonorific := fieldByName('Honorific').AsString;
    dbEmail := fieldByName('Email').AsString;
    close;
    end; {with Authors}

{   SendMail1.TO_Address:= dbEmail;
    SendMail1.TO_Name:= dbHonorific + '' + dbFName + '' + dbLName;

    SendMail1.action:= Send_Mail;

end; {Sendmail}

{procedure TForm1.onmailinfo(Sender: TObject; info: MailInfo;
    addinfo: String);
begin
    CGIEnvData1.send ('<P>');
    CGIEnvData1.sendhdr ('1', 'Mail Status');

    case info of

        Available :
        begin
            if SendMail1.status = Available then
                CGIEnvData1.send ('Message was sent.')

        else
        begin
            CGIEnvData1.send( 'mailform.exe : something went wrong<br>' );
            CGIEnvData1.send('Sendmail status is: ');
            case SendMail1.status of
                ServerConnected: CGIEnvData1.send( 'ServerConnected: ' );
                InvalidServer: CGIEnvData1.send( 'InvalidServer: ' );
                NoWinsock: CGIEnvData1.send( 'NoWinsock: ' );
                NoMailServer: CGIEnvData1.send( 'NoMailServer: ' );
                MailDestinationError: CGIEnvData1.send( 'MailDestinationError: ' );
                MailOriginError: CGIEnvData1.send( 'MailOriginError: ' );
                ProtocolError: CGIEnvData1.send( 'Protocol Error: ' );
                SocketError: CGIEnvData1.send( 'SocketError: ' );
            end;
        end;
    end;
end;

```



```

ReadError: CGIEnvData1.send( 'ReadError: ' );
WriteError: CGIEnvData1.send( 'WriteError: ' );
AlreadyBusy: CGIEnvData1.send( 'AlreadyBusy: ' );
Busy: CGIEnvData1.send( 'Busy: ' );
PrematureDisconnect: CGIEnvData1.send( 'PrematureDisconnect: ' );
end;
CGIEnvData1.send('info is: ');
case info of
  ServerConnected: CGIEnvData1.send( 'ServerConnected: ' );
  ServerDisconnected: CGIEnvData1.send( 'Server Disconnected: ' );
  SendingMessage: CGIEnvData1.send( 'SendingMessage: ' );
  PrematureDisconnect: CGIEnvData1.send( 'PrematureDisconnect: ' );
  Recipient: CGIEnvData1.send( 'Recipient: ' );
  BadAddress: CGIEnvData1.send( 'BadAddress: ' );
  ProtocolError: CGIEnvData1.send( 'Protocol Error: ' );
  SocketError: CGIEnvData1.send( 'SocketError: ' );
  ReadError: CGIEnvData1.send( 'ReadError: ' );
  WriteError: CGIEnvData1.send( 'WriteError: ' );
end;

CGIEnvData1.send( addinfo );
end; {else}

{ end; {if}
{ end; {case}

{ CGIEnvData1.closeStdout;
  closeApp( application );

end; }

end.

```


APPENDIX F. ADMINISTRATION SUB-SYSTEM CODE

This appendix contains the HTML and Delphi code for the HTML documents and Delphi executables that comprise the Administration Sub-system. The relationships of the programs is depicted in Figure F-1.

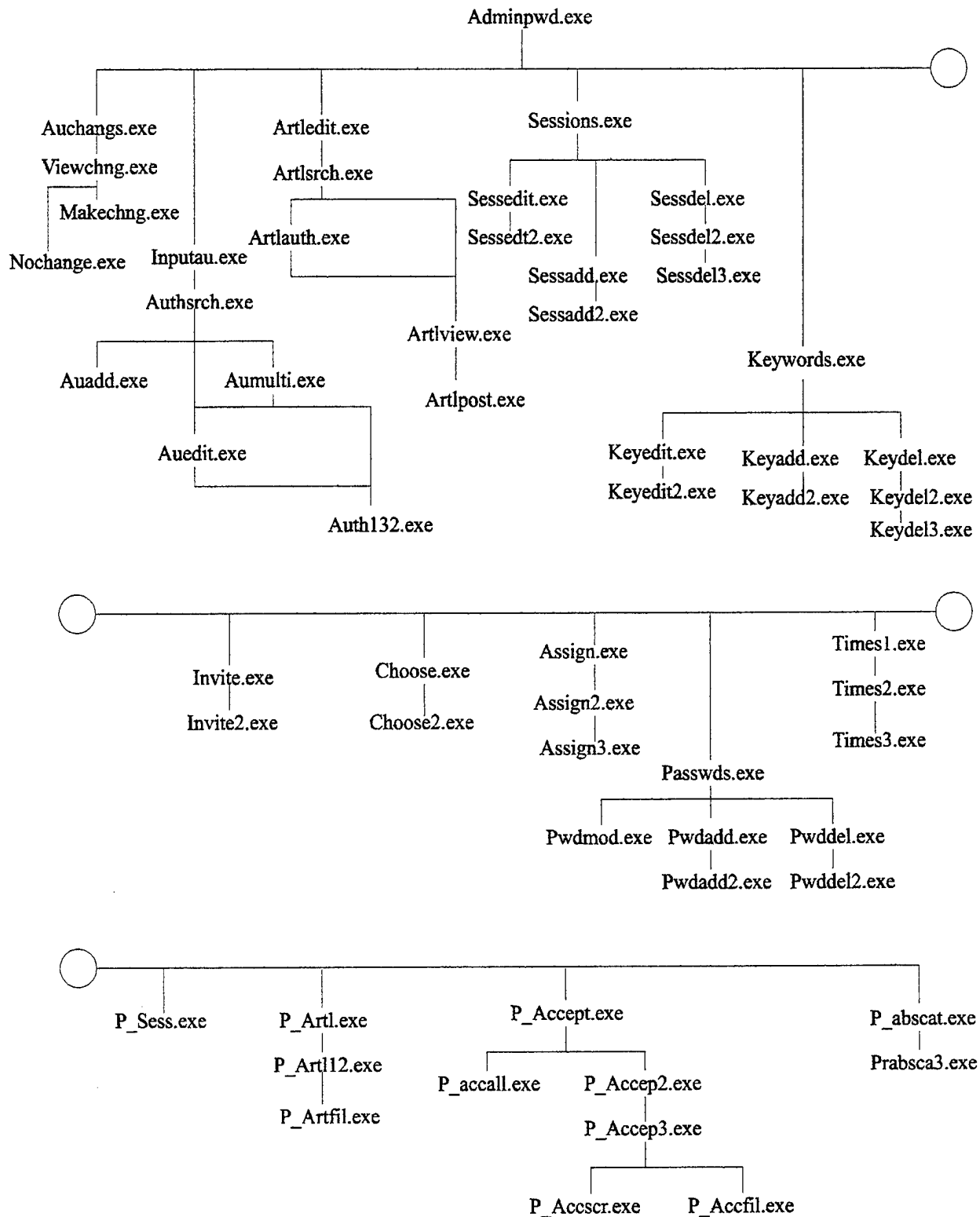


Figure F-1. Administration Sub-system.

UserID and Password verification page: **Admin20.htm**

```
<HTML>

<HEAD><TITLE>Asilomar Signals, Systems, and Computers</TITLE></HEAD>

<BODY bgcolor=FFFFFF>

<center><H2>Asilomar Conference on<BR>
Signals, Systems, & Computers</center></H2>

<H1>System Administration</H1>

<P> This page is intended for the exclusive use of the System Administrator for the Asilomar
Conference on Signals, Systems, & Computers.

<FORM ACTION="../../cgi-win/Adminpwd.exe " METHOD="POST">

<HR><CENTER>

<B>User name:  </B><INPUT NAME="userid" Size="20" TYPE="text">

<B>Password:  </B><INPUT NAME="pwd" Size="20" TYPE="password">

<INPUT NAME="flag" Size="3" TYPE="hidden" Value="0">

<P><INPUT TYPE="submit" Value="Submit Password"> <INPUT TYPE="reset"
VALUE="Clear Values"></CENTER>

<P><hr><P>

<IMG SRC="asil2b0.gif " align=left alt="Asilomar facility">

Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar
Conference On-line Submission Page</A>

</BODY></HTML>
```

System Administration Main Menu: **Adminpwd.exe**

```
unit Adm_pwd;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgi, Cgidb;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Table1: TTable;
    Query1: TQuery;
    CGIDB1: TCGIDB;
    CGIEnvData1: TCGIEnvData;
    Table1UserID: TStringField;
    Table1Password: TStringField;
    Query1Password: TStringField;

    procedure FormCreate(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  dUserID: string;
  dpwd: string;
  dbpassword: string;
  dflag: string;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

begin
```

with CGIEnvData1 do begin

```
webSiteINIFilename:=paramstr(1);  
application.onException:=cgiErrorHandler;  
application.processMessages;
```

```
createStdout;  
sendPrologue;
```

```
{ Get fields from search page }  
dUserID:= getsmallfield ('UserID');  
dpwd:= getsmallfield ('pwd');  
dflag:= getsmallfield ('flag');  
send ('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

```
end;
```

{Retrieve Password of the given User}

with Query1 do begin

```
close;  
sql.clear;  
sql.add('Select*FROM Admin WHERE UserID = "' + dUserID + '"');  
open;
```

```
Table1.Open;  
Table1.First;  
dbpassword := fieldByName('Password').AsString;  
Table1.Close;  
end;
```

with cgiEnvData1 do begin

{Determine whether password was correct}

```
if (dbpassword = dpwd) or (dflag = '1') then begin      {password was correct or flag is  
set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;  
send('<center><h3>Database Functions</h3>');
```

{Review changes to Author data Button}

```

send('<P><FORM ACTION="../cgi-win/auchangs.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><INPUT TYPE="submit" Value="Review Changes to Author
Information"></FORM>');

```

{Modify Authors Button}

```

send('<P><FORM ACTION="../cgi-win/inputau.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><INPUT TYPE="submit" Value=" Edit Author Information  "
Size="30"></FORM>');

```

{Modify Articles Button}

```

send('<P><FORM ACTION="../cgi-win/ArtlEdit.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><INPUT TYPE="submit" Value=" Edit Article data  "
Size="30"></FORM>');

```

{Modify Sessions Button}

```

send('<P><FORM ACTION="../cgi-win/Sessions.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><INPUT TYPE="submit" Value=" Edit Sessions  " ></FORM>');

```

{Modify Keywords Button}

```

send('<P><FORM ACTION="../cgi-win/Keywords.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><INPUT TYPE="submit" Value=" Edit Keywords  " ></FORM>');

```

{Invite Articles Button}

```

send('<P><FORM ACTION="../cgi-win/Invite.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value=" Invite Articles
"></CENTER></FORM>');

```

{Accept/reject Articles Button}

```

send('<P><FORM ACTION="../cgi-win/Choose.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value=" Accept / Reject Articles
"></CENTER></FORM>');

```

{Assign Articles to Sessions Button}

```

send('<P><FORM ACTION="../cgi-win/Assign.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value=" Assign Articles to Sessions "
></center></FORM>');

```


{Assign Presentation Times to Articles Button}

```

send('<P><FORM ACTION=" ../cgi-win/Times.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Assign Presentation Times to
Articles"></CENTER></FORM>');

```

{Modify Passwords Button}

```

send('<P><FORM ACTION=" ../cgi-win/Passwds.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value=" Modify Passwords "
></center></FORM>');

```

{Print Session List Button}

```

send('<hr><center><h3>Print Functions</h3></center>');
send('<P><FORM ACTION=" ../cgi-win/P_sess.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Print Session
list"></CENTER></FORM>');

```

{Print Article Information Button}

```

send('<P><FORM ACTION=" ../cgi-win/p_artl.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Print Article
Information"></CENTER></FORM>');

```

{Print Article accept/reject Button}

```

send('<P><FORM ACTION=" ../cgi-win/p_accept.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Print Article Accept/Reject
notice"></CENTER></FORM>');

```

{Print Abstract Catalog Button}

```

send('<P><FORM ACTION=" ../cgi-win/p_abscat.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Print Abstract
Catalog"></CENTER></FORM>');
end

```

```

else begin    {Password **incorrect**}

```

```

send('<center><H2>Asilomar Conference on Signals, Systems, &
Computers</H2></center>');
sendhr;
send('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');

```

```

    send ('<P>Please ensure that you are authorized to access this information. ');
    send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to
login again.</A>');
    send ('<P>If you are not authorized to access this information, please note that a log is');
    send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
    send ('prosecution for your actions.<BR>');
    end;

```

```

    send('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar
facility">');
    send('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
    send('Conference On-line Submission Page</A>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

```

```

    end; {with cgiEnvData1 do}
end; {FormCreate}
end.

```

Author Changes: **AuChange.exe**

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DBTables, DB, Cgidb, Cgi;

type

```
TForm1 = class(TForm)
  CGIEnvData1: TCGIEnvData;
  CGIDB1: TCGIDB;
  DataSource1: TDataSource;
  TempAuthors: TTable;
  Query1: TQuery;
  Query1FName: TStringField;
  Query1LName: TStringField;
  Query1Editnumber: TIntegerField;
  Query1Honorific: TStringField;
  Query1Institution: TStringField;
  Query1Department: TStringField;
  Query1ANumber: TIntegerField;
  Query1Initial: TStringField;
  Query1Mailstop: TStringField;
  Query1Address_Street: TStringField;
  Query1Address_City: TStringField;
  Query1Address_Zip: TStringField;
  Query1Phone_LocalNumber: TStringField;
  Query1Phone_FaxNumber: TStringField;
  Query1Email: TStringField;
  Query1Country: TStringField;
  Query1Address_State: TStringField;
  procedure FormCreate(Sender: TObject);
  procedure CGIDBOnSendingHotField(currentRecord: TDataset;
    var s: OpenString);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

Form1: TForm1;

implementation

```
{SR *.DFM}
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
var
```

```
    count: integer;
```

```
    countstr: string;
```

```
    dflag: string;           {valid program call variable}
```

```
begin
```

```
    with CGIEnvData1 do begin
```

```
        webSiteINIFilename:=paramstr(1);
```

```
        application.onException:=cgiErrorHandler;
```

```
        application.processMessages;
```

```
        createStdout;
```

```
        sendPrologue;
```

```
        { HTML page header }
```

```
        send ('<HTML><HEAD>');
```

```
        SendTitle('Asilomar Signals, Systems, and Computers');
```

```
        send ('</HEAD><BODY bgcolor=FFFFFF>');
```

```
{Determine whether flag is valid}
```

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin  {flag is valid}
```

```
    send('<center><H2>Asilomar Conference System Administration</H2></center>');
```

```
    sendhr;
```

```
    {Display all enrties in TEMPAUTHOR Table, ordered by ANumber, with Editnumber  
HotLink}
```

```
    with query1 do begin
```

```
        prepare;  { Optimizes query }
```

```
        close;    { should already be closed }
```

```
        sql.clear;
```

```
        sql.add( 'Select * FROM TempAuth ' );
```

```
        sql.add( 'Order by ANumber ' );
```

```

open;
count:= Recordcount;
countstr:= inttostr (count);
send ('<P>There are '+ countstr +' records to review.<P>');
CGIDB1.DrawTable;
send ('<P>');
end; {with query1 do}

sendhr;
send('<P><FORM ACTION="../../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end {flag is valid}

else begin  {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF="../../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC="../../prototype/asil2b0.gif" align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {flag **incorrect**}

end; {with cgiEnvData1 do}

end; {FormCreate}

{Send Editnumber as a Hot Link}

```

```
procedure TForm1.CGIDBOnSendingHotField(currentRecord: TDataset;  
  var s: OpenString);  
begin  
  s:='<A HREF=" ../cgi-win/ViewChng.exe?'+  
currentrecord.fieldbyName('Editnumber').AsString + ">"'  
  + currentrecord.FieldbyName('Editnumber').AsString + '</A>';  
  
end;  
end.
```

View changes: **ViewChng.exe**

```
unit View_chg;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    TempAuthors: TTable;
    Authors: TTable;
    procedure FormCreate(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
var
  dANumber: string;
  dENumber: string;
  db1FName: string;
  db1LName: string;
  db1Initial: string;
  db1Honorific: string;
  db1Institution: string;
  db1Department: string;
  db1Mailstop: string;
  db1Address_Street: string;
  db1Address_City: string;
```

```
db1Address_State: string;  
db1Address_Zip: string;  
db1Country: string;  
db1Phone_LocalNumber: string;  
db1Phone_FaxNumber: string;  
db1Email: string;
```

```
db2FName: string;  
db2LName: string;  
db2Initial: string;  
db2Honorific: string;  
db2Institution: string;  
db2Department: string;  
db2Mailstop: string;  
db2Address_Street: string;  
db2Address_City: string;  
db2Address_State: string;  
db2Address_Zip: string;  
db2Country: string;  
db2Phone_LocalNumber: string;  
db2Phone_FaxNumber: string;  
db2Email: string;
```

```
TC: TDataSet;
```

```
begin {TFormCreate}
```

```
with CGIEnvData1 do begin
```

```
  {Standard Header}
```

```
  webSiteINIFilename:=paramstr(1);  
  application.onException:=cgiErrorHandler;  
  application.processMessages;  
  createStdout;  
  sendPrologue;  
  send('<HTML><HEAD>');  
  SendTitle('Asilomar Signals, Systems, and Computers');  
  send('</HEAD><BODY bgcolor=FFFFFF>');
```

```
  send('<center><H1>Compare Author Information</H1></center>');  
  send('<P>Here is the old record along with the submitted changes.');
```

```
  sendHR;
```

```
  { Get the query string from the address}
```

```
  dENumber := CGIQueryString^;
```


{ Get the record information from the TempAuthor table }
with TempAuthors do begin

{ Move to proper record }

open;

first;

while fieldByName('Editnumber').asString <> dENumber do
 next;

{ Retrieve record }

dANumber := fieldByName('ANumber').AsString;

db2FName := fieldByName('FName').AsString;

db2LName := fieldByName('LName').AsString;

db2Initial := fieldByName('Initial').AsString;

db2Honorific := fieldByName('Honorific').AsString;

db2Institution := fieldByName('Institution').AsString;

db2Department := fieldByName('Department').AsString;

db2Mailstop := fieldByName('Mailstop').AsString;

db2Address_Street := fieldByName('Address_Street').AsString;

db2Address_City := fieldByName('Address_City').AsString;

db2Address_State := fieldByName('Address_State').AsString;

db2Address_Zip := fieldByName('Address_Zip').AsString;

db2Country := fieldByName('Country').AsString;

db2Phone_LocalNumber := fieldByName('Phone_LocalNumber').AsString;

db2Phone_FaxNumber := fieldByName('Phone_FaxNumber').AsString;

db2Email := fieldByName('Email').AsString;

Close;

end; **{with TempAuthors}**

{ Get the record information from the Author table }
with Authors do begin

{ Move to proper record }

open;

first;

while fieldByName('ANumber').asString <> dANumber do
 next;

{ Retrieve record }

db1FName := fieldByName('FName').AsString;

db1LName := fieldByName('LName').AsString;

db1Initial := fieldByName('Initial').AsString;

db1Honorific := fieldByName('Honorific').AsString;

```

db1Institution := fieldByName('Institution').AsString;
db1Department := fieldByName('Department').AsString;
db1Mailstop := fieldByName('Mailstop').AsString;
db1Address_Street := fieldByName('Address_Street').AsString;
db1Address_City := fieldByName('Address_City').AsString;
db1Address_State := fieldByName('Address_State').AsString;
db1Address_Zip := fieldByName('Address_Zip').AsString;
db1Country := fieldByName('Country').AsString;
db1Phone_LocalNumber := fieldByName('Phone_LocalNumber').AsString;
db1Phone_FaxNumber := fieldByName('Phone_FaxNumber').AsString;
db1Email := fieldByName('Email').AsString;
Close;
end; {with Authors}

```

```

{Create new page}
send('<P><FORM ACTION="..../cgi-win/MakeChng.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<INPUT TYPE="hidden" Name="ANumber" Value="'+ dANumber +'>');
send('<INPUT TYPE="hidden" Name="ENumber" Value="'+ dENumber +'>');
send('<B>OLD First Name: </B><INPUT NAME="OFName" Size="25" TYPE="text"
VALUE="'+ db1FName +'>');
    send('<br><B>NEW First Name: </B><INPUT NAME="FName" Size="25"
TYPE="text" VALUE="'+ db2FName +'>');
    send('<P><B>OLD Last Name: </B><INPUT NAME="OLName" Size="25"
TYPE="text" Value="'+ db1LName +'>');
    send('<br><B>NEW Last Name: </B><INPUT NAME="LName" Size="25"
TYPE="text" Value="'+ db2LName +'>');
    send('<P><B>OLD Middle Initial: </B><INPUT NAME="OInitial" Size="2"
TYPE="text" Value="'+ db1Initial +'>');
    send('<br><B>NEW Middle Initial: </B><INPUT NAME="Initial" Size="2" TYPE="text"
Value="'+ db2Initial +'>');
    send('<P><B>OLD Honorific Title:</B><INPUT NAME="OHonorific" Size="5"
TYPE="text" Value="'+ db1Honorific +'>');
    send('<br><B>NEW Honorific Title:</B><INPUT NAME="Honorific" Size="5"
TYPE="text" Value="'+ db2Honorific +'>');
    send('<P><B>OLD Institution: </B><INPUT NAME="OInstitution" Size="60"
TYPE="text">');
    send(' Value="'+ db1Institution +'>');
    send('<br><B>NEW Institution: </B><INPUT NAME="Institution" Size="60"
TYPE="text">');
    send(' Value="'+ db2Institution +'>');
    send('<P><B>OLD Department: </B><INPUT NAME="ODepartment" Size="60"
TYPE="text" Value="'+ db1Department +'>');
    send('<br><B>NEW Department: </B><INPUT NAME="Department" Size="60"

```

```

TYPE="text" Value="" + db2Department + ">");
    send('<P><B>OLD Mail Stop: </B><INPUT NAME="OMailstop" Size="30"
TYPE="text" Value="" + db1Mailstop + ">");
    send('<br><B>NEW Mail Stop: </B><INPUT NAME="Mailstop" Size="30"
TYPE="text" Value="" + db2Mailstop + ">");
    send('<P><B>OLD Street address: </B><INPUT NAME="OAddress_Street" Size="40"
TYPE="text");
    send('Value="" + db1Address_Street + ">');
    send('<br><B>NEW Street address: </B><INPUT NAME="Address_Street" Size="40"
TYPE="text");
    send('Value="" + db2Address_Street + ">');
    send('<P><B>OLD City: </B><INPUT NAME="OAddress_City" Size="20"
TYPE="text" Value="" + db1Address_City + ">");
    send('<br><B>NEW City: </B><INPUT NAME="Address_City" Size="20"
TYPE="text" Value="" + db2Address_City + ">");
    send('<P><B>OLD State: </B><INPUT NAME="OAddress_State" Size="2"
TYPE="text" Value="" + db1Address_State + ">");
    send('<br><B>NEW State: </B><INPUT NAME="Address_State" Size="2"
TYPE="text" Value="" + db2Address_State + ">");
    send('<P><B>OLD Zip Code: </B><INPUT NAME="OAddress_Zip" Size="10"
TYPE="text" Value="" + db1Address_Zip + ">");
    send('<br><B>NEW Zip Code: </B><INPUT NAME="Address_Zip" Size="10"
TYPE="text" Value="" + db2Address_Zip + ">");
    send('<P><B>OLD Country: </B><INPUT NAME="OCountry" Size="25" TYPE="text"
Value="" + db1Country + "> ');
    send('<br><B>NEW Country: </B><INPUT NAME="Country" Size="25" TYPE="text"
Value="" + db2Country + "> ');
    send('<P><B>OLD Telephone number (with Area/Country codes): </B><INPUT
NAME="OPhone_LocalNumber" Size="21");
    send(' TYPE="text"Value="" + db1Phone_LocalNumber + ">');
    send('<br><B>NEW Telephone number (with Area/Country codes): </B><INPUT
NAME="Phone_LocalNumber" Size="21");
    send(' TYPE="text"Value="" + db2Phone_LocalNumber + ">');
    send('<P><B>OLD Fax number (with Area/Country codes): </B><INPUT
NAME="OPhone_Faxnumber" Size="21");
    send(' TYPE="text"Value="" + db1Phone_FaxNumber + ">');
    send('<br><B>NEW Fax number (with Area/Country codes): </B><INPUT
NAME="Phone_Faxnumber" Size="21");
    send(' TYPE="text"Value="" + db2Phone_FaxNumber + ">');
    send('<P><B>OLD E-mail: </B><INPUT NAME="OEmail" Size="50" TYPE="text"
Value="" + db1Email + ">');
    send('<br><B>NEW E-mail: </B><INPUT NAME="Email" Size="50" TYPE="text"
Value="" + db2Email + ">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Accept
Changes"></CENTER></FORM>');

```

```

send('<P><FORM ACTION=" ../cgi-win/Nochange.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<INPUT TYPE="hidden" Name="ENumber" Value="'+ dENumber +' ">');
send('<P><CENTER><INPUT TYPE="submit" Value="Do NOT Modify this Record.
Discard Changes."></CENTER></FORM>');

```

```

sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send('</BODY></HTML>');
closeStdout;
closeApp( application );

```

```

end; {with cgiEnvData1 do}

```

```

end; {Procedure TForm1.FormCreate}

```

```

end.

```

Make the Change: **MakeChng.exe**

```
unit Make_chg;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Authors: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    TempAuthors: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  dANumber: string;
  dENumber: string;
  dANumberint: integer;
  dFName: string;
  dLName: string;
  dInitial: string;
  dHonorific: string;
  dInstitution: string;
  dDepartment: string;
  dMailstop: string;
  dAddress_Street: string;
  dAddress_City: string;
  dAddress_State: string;
  dAddress_Zip: string;
  dCountry: string;
  dPhone_AreaCode: string;
  dPhone_LocalNumber: string;
  dPhone_FaxNumber: string;
  dEmail: string;
```

dflag: string; **{valid program call}**

implementation

{ \$R *.DFM }

procedure TForm1.FormCreate(Sender: TObject);
begin

 with CGIEnvData1 do
 begin

{Standard Header}

 webSiteINIFilename:=paramstr(1);
 application.onException:=cgiErrorHandler;
 application.processMessages;
 createStdout;
 sendPrologue;

{Standard Header}

 send ('<HTML><HEAD>');
 SendTitle('Asilomar Signals, Systems, and Computers');
 send ('</HEAD><BODY bgcolor=FFFFFF>');

{Determine whether flag is valid}

dflag:= getsmallfield ('flag');

if (dflag = '1') then begin **{flag is valid}**

 send('<center><H2>Asilomar Conference System Administration</H2></center>');
 sendhr;

{Retrieve data from Form}

 dANumber:= getsmallfield ('ANumber');
 dENumber:= getsmallfield ('ENumber');
 dFName:= getsmallfield ('FName');
 dLName:= getsmallfield ('LName');
 dInitial:= getsmallfield ('Initial');
 dHonorific:= getsmallfield ('Honorific');
 dInstitution:= getsmallfield ('Institution');
 dDepartment:= getsmallfield ('Department');
 dMailstop:= getsmallfield ('Mailstop');
 dAddress_Street:= getsmallfield ('Address_Street');

```

dAddress_City:= getsmallfield ('Address_City');
dAddress_State:= getsmallfield ('Address_State');
dAddress_Zip:= getsmallfield ('Address_Zip');
dCountry:= getsmallfield ('Country');
dPhone_LocalNumber:= getsmallfield ('Phone_LocalNumber');
dPhone_FaxNumber:= getsmallfield ('Phone_Faxnumber');
dEmail:= getsmallfield ('Email');

```

{Add new information to AUTHOR Table}

with Authors do begin

{ Move to proper record }

```

open;
first;
while fieldbyName('ANumber').asstring <> dANumber do
    next;

```

{Update record }

```

edit;
SetFields([nil, dFName, dLName, dInitial, dHonorific, dInstitution, dDepartment, dMailstop,
dAddress_Street, dAddress_City, dAddress_State, dAddress_Zip, dCountry,
dPhone_LocalNumber, dPhone_FaxNumber, dEmail]);
post;
close;
end;  {with Authors}

```

{Remove entry from TEMPAUTHOR Table}

With TempAuthors do begin

{ Move to proper record }

```

open;
first;
while fieldbyName('Editnumber').asstring <> dENumber do
    next;

```

{Delete Record}

```

delete;
close;
end; {With TempAuthors}

```

{Send the Thankyou page }

```

send ('<P>The information about ' + dHonorific + ' ' + dFName + ' ' + dLName + ' ');
send ('has been updated in our records. ');
send ('</CENTER>');

```

```

    send('<P><FORM ACTION="../../cgi-win/AuChangs.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Review changes to another
Author"></CENTER></FORM>');

    sendhr;
    send('<P><FORM ACTION="../../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is valid}

else begin    {flag **incorrect**}

    send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
    sendhr;
    send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
    send ('<P>Please ensure that you are authorized to access this information. ');
    send ('<P>If you made an error, then please <A HREF="../../prototype/admin20.htm">try to login
again.</A>');
    send ('<P>If you are not authorized to access this information, please note that a log is');
    send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
    send ('prosecution for your actions.<BR>');

    send ('<P><hr><P><IMG SRC="../../prototype/asil2b0.gif" align=left alt="Asilomar facility">');
    send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
    send ('Conference On-line Submission Page</A>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

    end; {flag **incorrect**}

    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );
    end; {with CGIEnvData1}
end; {FormCreate}

end. {Application}

Do not make changes: NoChanges.exe

unit No_chang;

```


interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type

TForm1 = class(TForm)
 DataSource1: TDataSource;
 TempAuthors: TTable;
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 procedure FormCreate(Sender: TObject);
private
 { Private declarations }
public
 { Public declarations }
end;

var

Form1: TForm1;
dENumber:string;
dflag: string; {valid program call}

implementation

{SR *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin

 with CGIEnvData1 do
 begin

{Standard Header}

 webSiteINIFilename:=paramstr(1);
 application.onException:=cgiErrorHandler;
 application.processMessages;
 createStdout;
 sendPrologue;

 send ('<HTML><HEAD>');
 SendTitle('Asilomar Signals, Systems, and Computers');
 send ('</HEAD><BODY bgcolor=FFFFFF>');

{Determine whether flag is valid}

```

dflag:= getsmallfield ('flag');

if (dflag = '1') then begin  {flag is valid}

send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

dENumber := getsmallfield ('ENumber');

{Delete information from TEMPAUTHOR Table}
with TempAuthors do begin

  { Move to proper record}
  open;
  first;
  while fieldbyname('Editnumber').asString <> dENumber do
    next;

  {Delete record }
  delete;
  close;
end;  {with TempAuthors}

{Send the Confirmation}
send ('<P>The temporary changes to the Author data have been deleted from our records.');
```

send('<P><FORM ACTION=" ../cgi-win/AuChangs.exe" METHOD="POST">');
 send('<INPUT TYPE="hidden" Name="flag" Value="1">');
 send('<P><CENTER><INPUT TYPE="submit" Value="Review changes to another
 Author"></CENTER></FORM>');

```

  sendhr;
  send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
  send('<INPUT TYPE="hidden" Name="flag" Value="1">');
  send('<P><CENTER><INPUT TYPE="submit" Value="Go to System Admin Options  

  page"></CENTER></FORM>');
end {flag is valid}

else begin  {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information.');
```

```

send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

    end; {flag **incorrect**}

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}

end; {FormCreate}

end. {Application}

```

Input Author Name: **Inputau.exe**

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, Cgldb, Cgi, DBTables, DB;

type

TForm1 = class(TForm)
 DataSource1: TDataSource;
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;

 procedure FormCreate(Sender: TObject);

private

 { Private declarations }

public

 { Public declarations }

end;

var

 Form1: TForm1;

 dflag: string; **{valid program call}**

implementation

 {\$R *.DFM}

 procedure TForm1.FormCreate(Sender: TObject);

begin

 with CGIEnvData1 do

 begin

 webSiteINIFilename:=paramstr(1);

 application.onException:=cgiErrorHandler;

 application.processMessages;

 createStdout;

 sendPrologue;

 send ('<HTML><HEAD>');

 SendTitle('Asilomar Signals, Systems, and Computers');

```

send ('</HEAD><BODY bgcolor=FFFFFF>');

{Determine whether flag is valid}

dflag:= getsmallfield ('flag');

if (dflag = '1') then begin  {flag is valid}

send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

send('<H1>Edit Author Information</H1>');

send('<P>To locate an author name in the database, please enter the name below: ');
send('<P><FORM METHOD="POST" ACTION=" ../cgi-win/AuthSrch.exe ">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');

send('<B>First Name:  </B><INPUT NAME="first" Size="20" TYPE="TEXT"> ');
send('<B>Last Name:  </B><INPUT NAME="last" Size="20" TYPE="TEXT">');
send('<B>Middle Initial: </B><INPUT NAME="initial" Size="2" TYPE="TEXT">');

send('<P><CENTER><INPUT TYPE="submit" Value="submit author name">');
send('<INPUT TYPE="reset" VALUE="Clear this form"></CENTER>');
send ('</form>');
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is valid}

else begin  {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL.  You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

```

```
send ('<P><hr><P><IMG SRC="../../prototype/asil2b0.gif" align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
```

```
end; {flag **incorrect**}
```

```
send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}
```

```
end; {Procedure FormCreate}
```

```
end.
```

Add a new Author: **AuAdd.exe**

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type

TForm1 = class(TForm)
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 DataSource1: TDataSource;
 Authors: TTable;

 procedure FormCreate(Sender: TObject);

private

 { Private declarations }

public

 { Public declarations }

end;

var

 Form1: TForm1;

 dflag: string; **{valid program call}**

{Database Variables}

dANumber: string;
dFName: string;
dLName: string;
dInitial: string;
dHonorific: string;
dInstitution: string;
dDepartment: string;
dMailstop: string;
dAddress_Street: string;
dAddress_City: string;
dAddress_State: string;
dAddress_Zip: string;
dCountry: string;
dPhone_AreaCode: string;
dPhone_LocalNumber: string;
dPhone_FaxNumber: string;

dEmail: string;

implementation

{ \$R *.DFM }

procedure TForm1.FormCreate(Sender: TObject);
begin

with CGIEnvData1 do begin

{Standard Header}

webSiteINIFilename:=paramstr(1);
application.onException:=cgiErrorHandler;
application.processMessages;
createStdout;
sendPrologue;

send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');

{Determine whether flag is valid}

dflag:= getsmallfield ('flag');

if (dflag = '1') then begin **{flag is valid}**

send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

{Get data from FORM}

dFName:= getsmallfield ('FName');
dLName:= getsmallfield ('LName');
dInitial:= getsmallfield ('Initial');
dHonorific:= getsmallfield ('Honorific');
dInstitution:= getsmallfield ('Institution');
dDepartment:= getsmallfield ('Department');
dMailstop:= getsmallfield ('Mailstop');
dAddress_Street:= getsmallfield ('Address_Street');
dAddress_City:= getsmallfield ('Address_City');
dAddress_State:= getsmallfield ('Address_State');
dAddress_Zip:= getsmallfield ('Address_Zip');
dCountry:= getsmallfield ('Country');


```
dPhone_LocalNumber:= getsmallfield ('Phone_LocalNumber');
dPhone_FaxNumber:= getsmallfield ('Phone_Faxnumber');
dEmail:= getsmallfield ('Email');
```

{Append data to AUTHOR Table}

```
Authors.open;
Authors.AppendRecord([nil, dFName, dLName, dInitial, dHonorific, dInstitution,
dDepartment, dMailstop,
dAddress_Street, dAddress_City, dAddress_State, dAddress_Zip, dCountry, nil{old
areacode},
dPhone_LocalNumber, dPhone_FaxNumber, dEmail]);
Authors.close;
```

{Send the Thankyou page }

```
send ('<P>The information about ' + dHonorific + ' ' + dFName + ' ' + dLName + ' ');
send ('has been added to our records. ');
send ('</CENTER>');
```

```
send ('<P><FORM ACTION=" ../cgi-win/InputAu.exe" METHOD="POST">');
send ('<INPUT TYPE="hidden" Name="flag" Value="1">');
send ('<P><CENTER><INPUT TYPE="submit" Value="Edit another
Author"></CENTER></FORM>');
sendhr;
send ('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send ('<INPUT TYPE="hidden" Name="flag" Value="1">');
send ('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
```

end {flag is valid}

else begin {flag **incorrect**}

```
send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');
```

```
send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
```

```
send ('Conference On-line Submission Page</A>');
```

```
    end; {flag **incorrect**}
```

```
send ('</BODY></HTML>');
```

```
closeStdout;
```

```
closeApp( application );
```

```
end; {with CGIEnvData1}
```

```
end; {FormCreate}
```

```
end. {Application}
```

Multiple Matches: AuMulti.exe

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type

TForm1 = class(TForm)
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 DataSource1: TDataSource;
 Table1: TTable;
 procedure FormCreate(Sender: TObject);
private
 { Private declarations }
public
 { Public declarations }
end;

var

Form1: TForm1;

implementation

{ \$R *.DFM }

procedure TForm1.FormCreate(Sender: TObject);

var

 {Database Variables}
 dANumber: string;
 dbFName: string;
 dbLName: string;
 dbInitial: string;
 dbHonorific: string;
 dbInstitution: string;
 dbDepartment: string;
 dbMailstop: string;
 dbAddress_Street: string;
 dbAddress_City: string;
 dbAddress_State: string;
 dbAddress_Zip: string;
 dbCountry: string;

```
dbPhone_LocalNumber: string;  
dbPhone_FaxNumber: string;  
dbEmail: string;
```

```
TC: TDataSet;
```

```
begin {TFormCreate}
```

```
with CGIEnvData1 do begin
```

```
  {Standard Header}
```

```
  webSiteINIFilename:=paramstr(1);  
  application.onException:=cgiErrorHandler;  
  application.processMessages;  
  createStdout;  
  sendPrologue;
```

```
  { Get the query string from the address}
```

```
  dANumber := CGIQueryString^;
```

```
  { Get the record information }
```

```
  with Table1 do begin
```

```
    { Move to proper record}
```

```
    open;  
    first;  
    while fieldByName('ANumber').asString <> dANumber do  
      next;
```

```
    { Retrieve record }
```

```
    dbFName := fieldByName('FName').AsString;  
    dbLName := fieldByName('LName').AsString;  
    dbInitial := fieldByName('Initial').AsString;  
    dbHonorific := fieldByName('Honorific').AsString;  
    dbInstitution := fieldByName('Institution').AsString;  
    dbDepartment := fieldByName('Department').AsString;  
    dbMailstop := fieldByName('Mailstop').AsString;  
    dbAddress_Street := fieldByName('Address_Street').AsString;  
    dbAddress_City := fieldByName('Address_City').AsString;  
    dbAddress_State := fieldByName('Address_State').AsString;  
    dbAddress_Zip := fieldByName('Address_Zip').AsString;  
    dbCountry := fieldByName('Country').AsString;  
    dbPhone_LocalNumber := fieldByName('Phone_LocalNumber').AsString;  
    dbPhone_FaxNumber := fieldByName('Phone_FaxNumber').AsString;  
    dbEmail := fieldByName('Email').AsString;
```

```
Close;
end;
```

{Create new page}

```
send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

```
send ('<center><H1>Edit Author Information</H1></center>');
send ('<P>We have located the record in the database.');
```

```
sendhr;
```

{Send form with Author information}

```
send ('<P><FORM ACTION="AuEdit.exe" METHOD="POST">');
send ('<INPUT NAME="ANumber" TYPE="hidden" Value="" + dANumber + ">');
    send ('<B>First Name: </B><INPUT NAME="FName" Size="25" TYPE="text"
VALUE="" + dbFName + ">');
    send ('<B> Last Name: </B><INPUT NAME="LName" Size="25" TYPE="text"
Value="" + dbLName + ">');
    send ('<B> Middle Initial: </B><INPUT NAME="Initial" Size="2" TYPE="text" Value=""
+ dbInitial + "><HR>');
    send ('<B>Honorific Title:</B><INPUT NAME="Honorific" Size="5" TYPE="text"
Value="" + dbHonorific + ">');
    send ('<P><B>Institution: </B><INPUT NAME="Institution" Size="60" TYPE="text";
    send (' Value="" + dbInstitution + ">');
    send ('<br><B>Department: </B><INPUT NAME="Department" Size="60"
TYPE="text" Value="" + dbDepartment + ">');
    send ('<br><B>Mail Stop: </B><INPUT NAME="Mailstop" Size="30" TYPE="text"
Value="" + dbMailstop + ">');
    send ('<P><B>Street address: </B><INPUT NAME="Address_Street" Size="40"
TYPE="text");
    send ('Value="" + dbAddress_Street + ">');
    send ('<br><B>City: </B><INPUT NAME="Address_City" Size="20" TYPE="text"
Value="" + dbAddress_City + ">');
    send ('<B>State: </B><INPUT NAME="Address_State" Size="2" TYPE="text"
Value="" + dbAddress_State + ">');
    send ('<br><B>Zip Code: </B><INPUT NAME="Address_Zip" Size="10"
TYPE="text" Value="" + dbAddress_Zip + ">');
    send ('<B>Country: </B><INPUT NAME="Country" Size="25" TYPE="text"
Value="" + dbCountry + "> ');
    send ('<P><B>Telephone number (with Area/Country codes): </B><INPUT
NAME="Phone_LocalNumber" Size="21";
    send (' TYPE="text"Value="" + dbPhone_LocalNumber + ">');
    send ('<br><B>Fax number (with Area/Country codes): </B><INPUT
```

```

NAME="Phone_Faxnumber" Size="21");
    send(' TYPE="text"Value="" + dbPhone_FaxNumber + ">');
    send('<br><B>E-mail:      </B><INPUT NAME="Email" Size="50" TYPE="text"
Value="" + dbEmail + ">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Submit Information"> <INPUT
TYPE="reset">');
    send('VALUE="Reset to Original Information"></CENTER>');

    send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

    end; {with cgiEnvData1 do}

end; {Procedure TForm1.FormCreate}

end.

```

Edit an Author: **AuEdit.exe**

```
unit Au_edit;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Authors: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    AuthorsANumber: TIntegerField;
    AuthorsLName: TStringField;
    AuthorsFName: TStringField;
    AuthorsInitial: TStringField;
    AuthorsHonorific: TStringField;
    AuthorsInstitution: TStringField;
    AuthorsDepartment: TStringField;
    AuthorsMailstop: TStringField;
    AuthorsAddress_Street: TStringField;
    AuthorsAddress_City: TStringField;
    AuthorsAddress_State: TStringField;
    AuthorsAddress_Zip: TStringField;
    AuthorsCountry: TStringField;
    AuthorsPhone_LocalNumber: TStringField;
    AuthorsPhone_FaxNumber: TStringField;
    AuthorsEmail: TStringField;

    procedure FormCreate(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

{Database Variables}
```

```

dANumber: string;
dANumberint: integer;
dFName: string;
dLName: string;
dInitial: string;
dHonorific: string;
dInstitution: string;
dDepartment: string;
dMailstop: string;
dAddress_Street: string;
dAddress_City: string;
dAddress_State: string;
dAddress_Zip: string;
dCountry: string;
dPhone_AreaCode: string;
dPhone_LocalNumber: string;
dPhone_FaxNumber: string;
dEmail: string;

```

```

dflag: string;           {valid program call}

```

```

implementation

```

```

{$R *.DFM}

```

```

procedure TForm1.FormCreate(Sender: TObject);

```

```

begin

```

```

    with CGIEnvData1 do
    begin

```

```

        {Standard Header}

```

```

        webSiteINIFilename:=paramstr(1);
        application.onException:=cgiErrorHandler;
        application.processMessages;
        createStdout;
        sendPrologue;

```

```

        send('<HTML><HEAD>');
        SendTitle('Asilomar Signals, Systems, and Computers');
        send('</HEAD><BODY bgcolor=FFFFFF>');

```

```

{Determine whether flag is valid}

```



```

dflag:= getsmallfield ('flag');

if (dflag = '1') then begin   {flag is set}

send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

  {Get data from Form}
  dANumber:= getsmallfield ('ANumber');
  dFName:= getsmallfield ('FName');
  dLName:= getsmallfield ('LName');
  dInitial:= getsmallfield ('Initial');
  dHonorific:= getsmallfield ('Honorific');
  dInstitution:= getsmallfield ('Institution');
  dDepartment:= getsmallfield ('Department');
  dMailstop:= getsmallfield ('Mailstop');
  dAddress_Street:= getsmallfield ('Address_Street');
  dAddress_City:= getsmallfield ('Address_City');
  dAddress_State:= getsmallfield ('Address_State');
  dAddress_Zip:= getsmallfield ('Address_Zip');
  dCountry:= getsmallfield ('Country');
  dPhone_LocalNumber:= getsmallfield ('Phone_LocalNumber');
  dPhone_FaxNumber:= getsmallfield ('Phone_Faxnumber');
  dEmail:= getsmallfield ('Email');

  dANumberint:= StrToInt(dANumber);    { Convert the dANumber back to an integer }

  with Authors do begin

    { Move to proper record}
    open;
    first;
    while fieldbyname('ANumber').asString <> dANumber do
      next;

    {Update record }
    edit;
    SetFields([dANumberint, dFName, dLName, dInitial, dHonorific, dInstitution, dDepartment,
dMailstop,
dAddress_Street, dAddress_City, dAddress_State, dAddress_Zip, dCountry,
dPhone_LocalNumber, dPhone_FaxNumber, dEmail]);
    post;
    close;
  end;   {with Authors}

```

```

{Send the Thankyou page }
send ('<P>The information about ' + dHonorific + ' ' + dFName + ' ' + dLName + ' ');
send ('has been updated in our records. ');
send ('</CENTER>');

send ('<P><FORM ACTION=" ../cgi-win/InputAu.exe" METHOD="POST">');
send ('<INPUT TYPE="hidden" Name="flag" Value="1">');
send ('<P><CENTER><INPUT TYPE="submit" Value="Edit another
Author"></CENTER></FORM>');
sendhr;
send ('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send ('<INPUT TYPE="hidden" Name="flag" Value="1">');
send ('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');

end {flag is valid}

else begin {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

    end; {flag **incorrect**}

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}

end; {FormCreate}

```

end. {Application}

Edit an Article: **ArtlEdit.exe**

```
unit Art_edit;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Query1: TQuery;
    Table1: TTable;
    Table2: TTable;
    Table2ANumber: TIntegerField;
    Table2FName: TStringField;
    Table2LName: TStringField;
    Table2Initial: TStringField;
    Table2Honorific: TStringField;
    Table2Institution: TStringField;
    Table2Department: TStringField;
    Table2Mailstop: TStringField;
    Table2Address_Street: TStringField;
    Table2Address_City: TStringField;
    Table2Address_State: TStringField;
    Table2Address_Zip: TStringField;
    Table2Country: TStringField;
    Table2Phone_LocalNumber: TStringField;
    Table2Phone_AreaCode: TStringField;
    Table2Phone_FaxNumber: TStringField;
    Table2Email: TStringField;
    Table3: TTable;
    Table3Number: TIntegerField;
    Table3Neumonic: TStringField;
    Query1Neumonic: TStringField;
    DataSource2: TDataSource;
    Query2: TQuery;
    Table4: TTable;
    Query1Number: TIntegerField;
    Query2Name: TStringField;
    Table4Number: TIntegerField;
```

```

Table4Name: TStringField;
Query2Number: TIntegerField;
Table1Paper_Number: TIntegerField;
Table1Title: TStringField;
Table1Invited: TStringField;
Table1Accepted: TStringField;
Table1ContactAuthorNumber: TIntegerField;
Table1Session: TStringField;
Table1OrderInSession: TStringField;
Table1PresentationTime: TStringField;
Table1Keyword1: TStringField;
Table1Keyword2: TStringField;
Table1Keyword3: TStringField;
Table1Abstract: TMemoField;
Table1ContactOrder: TStringField;
Table1FName2: TStringField;
Table1LName2: TStringField;
Table1Initial2: TStringField;
Table1Institution2: TStringField;
Table1Order2: TStringField;
Table1FName3: TStringField;
Table1LName3: TStringField;
Table1Initial3: TStringField;
Table1Institution3: TStringField;
Table1Order3: TStringField;
Table1FName4: TStringField;
Table1LName4: TStringField;
Table1Initial4: TStringField;
Table1Institution4: TStringField;
Table1Order4: TStringField;
Table1FName5: TStringField;
Table1LName5: TStringField;
Table1Initial5: TStringField;
Table1Institution5: TStringField;
Table1Order5: TStringField;
Table1FName6: TStringField;
Table1LName6: TStringField;
Table1Initial6: TStringField;
Table1Institution6: TStringField;
Table1Order6: TStringField;
procedure FormCreate(Sender: TObject);

```

```
private
```

```
{ Private declarations }
```

```
public
```

```
    { Public declarations }  
end;
```

```
var  
  Form1: TForm1;  
  dflag: string;      {Valid program call}  
  dbOption: string;   {Which action is requested}  
  i:integer;          {loop control variable}
```

```
implementation
```

```
{ $R *.DFM }
```

```
procedure TForm1.FormCreate(Sender: TObject);  
begin
```

```
  with CGIEnvData1 do begin
```

```
    webSiteINIFilename:=paramstr(1);  
    application.onException:=cgiErrorHandler;  
    application.processMessages;
```

```
    createStdout;  
    sendPrologue;
```

```
    send ('<HTML><HEAD>');  
    SendTitle('Asilomar Signals, Systems, and Computers');  
    send ('</HEAD><BODY bgcolor=FFFFFF>');
```

```
{Determine whether flag is valid }
```

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin  {flag is valid}
```

```
  send('<center><H2>Asilomar Conference System Administration</H2></center>');  
  sendhr;
```

```
    send('In order to select the Article you would like to view, you will have to select it from a  
list of articles.');
```

```
    send('You have three options for building this list:<BR>');
```

```
{Display ALL Articles }
```

```
  sendhr;
```

```

sendhdr ('3', 'Choose from a list of ALL the Articles?');
send('<P><FORM ACTION=" ../cgi-win/ArtlSrch.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<INPUT TYPE="hidden" Name="Type" Value="all">');
send('<P><CENTER><INPUT TYPE="submit" Value="Show me a list of ALL of the Paper
Titles"></CENTER></FORM>');

```

{Search by Article characteristics}

```

send('<P><FORM METHOD="POST" ACTION=" ../cgi-win/ArtlSrch.exe">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">'); {send flag for verification}
send('<INPUT TYPE="hidden" Name="Type" Value="article">');

```

```

sendhr;
sendhdr ('3', 'Choose from a list of Articles with these characteristics?');
send('<P><center><B>Invited Papers: </B><SELECT NAME="invited">');
send('<OPTION> Do not search on this field');
send('<OPTION> Yes');
send('<OPTION> No');
send('</SELECT>');

```

```

send('<P><B>Accepted Papers: </B><SELECT NAME="accepted">');
send('<OPTION> Do not search on this field');
send('<OPTION> Yes');
send('<OPTION> No');
send('</SELECT>');

```

{Pull Sessions from SESSION Table and display on pull-down menu}

with Query1 do begin

```

close;
sql.clear;
sql.add('Select Number, Neumonic FROM Session');
open;

```

{Build option menu from query results }

```

Table3.open;
Table3.first;

```

```

send('<P><B>Sessions: </B><SELECT NAME="Session">');
send('<OPTION> Do not search on this field');

```

for i := 1 to Recordcount do begin

```

dbOption:= Table3.fieldByName('Neumonic').AsString;
send('<OPTION> ' +dbOption+ ');
Table3.next;
end; {for all records in the query result}

```

```
send('</SELECT>');
```

```
Table1.close;
```

```
end; {withQuery1}
```

```
{Pull Keywords from KEYWORD Table and display on pull-down menu}
```

```
with Query2 do begin
```

```
  close;
```

```
  sql.clear;
```

```
  sql.add('Select * FROM Keyword ');
```

```
  open;
```

```
{Move records from query result to stringlist}
```

```
Table4.open;
```

```
Table4.first;
```

```
send('<P><B>Keyword: </B><SELECT NAME="Keyword">');
```

```
send('<OPTION> Do not search on this field');
```

```
while not Table4.EOF do begin
```

```
  dbOption:= Table4.fieldByName('Name').AsString;
```

```
  send('<OPTION> ' +dbOption+ "');
```

```
  Table4.next;
```

```
end; {for all records in the query result}
```

```
send('</SELECT>');
```

```
end; {withQuery2}
```

```
send('<P><INPUT TYPE="submit" Value="Show me a list of Articles like this"> ');
```

```
send('<INPUT TYPE="reset" VALUE="Reset Article Fields"></CENTER></form>');
```

```
{Search by Author Name}
```

```
send('<P><FORM ACTION=" ../cgi-win/ArtlSrch.exe" METHOD="POST">');
```

```
send('<INPUT TYPE="hidden" Name="flag" Value="1">'); {send flag for verification}
```

```
send('<INPUT TYPE="hidden" Name="Type" Value="author">');
```

```
sendhdr;
```

```
sendhdr('3', 'Choose from a list of Articles by this Author?');
```

```
send ('<P><CENTER><B>Author First Name: </B><INPUT NAME="fname" Size="25"  
TYPE="text">');
```

```
  send ('<P><B>Author Last Name: </B><INPUT NAME="lname" Size="25"  
TYPE="text">');
```

```
send('<P><INPUT TYPE="submit" Value="Find an Article by this Author"> ');
```

```
send('<INPUT TYPE="reset" VALUE="Clear Names"></CENTER></form>');
```



```

sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');

send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end {flag is valid}

else begin {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {flag **incorrect**}

end; {with cgiEnvData1 do}

end; {FormCreate}

end. {Application}

```

Search for the Article: **ArtlSrch.exe**

unit Art_srch;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type

TForm1 = class(TForm)
 DataSource1: TDataSource;
 Query1: TQuery;
 Table1: TTable;
 Table2: TTable;
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 Query2: TQuery;
 DataSource2: TDataSource;
 Query1Title: TStringField;
 Query2Title: TStringField;
 Query1Paper_Number: TIntegerField;
 CGIDB2: TCGIDB;
 Query2Paper_Number: TIntegerField;
 Table3: TTable;
 DataSource3: TDataSource;
 Query3: TQuery;
 Query3ANumber: TIntegerField;
 CGIDB3: TCGIDB;
 Query3FName: TStringField;
 Query3LName: TStringField;
 Query3Initial: TStringField;
 Query3Honorific: TStringField;
 Query3Institution: TStringField;
 CGIDB4: TCGIDB;
 DataSource4: TDataSource;
 Table4: TTable;
 Query4: TQuery;
 Query4Paper_Number: TIntegerField;
 Query4Title: TStringField;
 Table1Paper_Number: TIntegerField;
 Table1Title: TStringField;
 Table1Invited: TStringField;
 Table1Accepted: TStringField;

Table1ContactAuthorNumber: TIntegerField;
 Table1Session: TStringField;
 Table1OrderInSession: TStringField;
 Table1PresentationTime: TStringField;
 Table1Keyword1: TStringField;
 Table1Keyword2: TStringField;
 Table1Keyword3: TStringField;
 Table1Abstract: TMemoField;
 Table1ContactOrder: TStringField;
 Table1FName2: TStringField;
 Table1LName2: TStringField;
 Table1Initial2: TStringField;
 Table1Institution2: TStringField;
 Table1Order2: TStringField;
 Table1FName3: TStringField;
 Table1LName3: TStringField;
 Table1Initial3: TStringField;
 Table1Institution3: TStringField;
 Table1Order3: TStringField;
 Table1FName4: TStringField;
 Table1LName4: TStringField;
 Table1Initial4: TStringField;
 Table1Institution4: TStringField;
 Table1Order4: TStringField;
 Table1FName5: TStringField;
 Table1LName5: TStringField;
 Table1Initial5: TStringField;
 Table1Institution5: TStringField;
 Table1Order5: TStringField;
 Table1FName6: TStringField;
 Table1LName6: TStringField;
 Table1Initial6: TStringField;
 Table1Institution6: TStringField;
 Table1Order6: TStringField;
 Table2Paper_Number: TIntegerField;
 Table2Title: TStringField;
 Table2Invited: TStringField;
 Table2Accepted: TStringField;
 Table2ContactAuthorNumber: TIntegerField;
 Table2Session: TStringField;
 Table2OrderInSession: TStringField;
 Table2PresentationTime: TStringField;
 Table2Keyword1: TStringField;
 Table2Keyword2: TStringField;
 Table2Keyword3: TStringField;

Table2Abstract: TMemoField;
 Table2ContactOrder: TStringField;
 Table2FName2: TStringField;
 Table2LName2: TStringField;
 Table2Initial2: TStringField;
 Table2Institution2: TStringField;
 Table2Order2: TStringField;
 Table2FName3: TStringField;
 Table2LName3: TStringField;
 Table2Initial3: TStringField;
 Table2Institution3: TStringField;
 Table2Order3: TStringField;
 Table2FName4: TStringField;
 Table2LName4: TStringField;
 Table2Initial4: TStringField;
 Table2Institution4: TStringField;
 Table2Order4: TStringField;
 Table2FName5: TStringField;
 Table2LName5: TStringField;
 Table2Initial5: TStringField;
 Table2Institution5: TStringField;
 Table2Order5: TStringField;
 Table2FName6: TStringField;
 Table2LName6: TStringField;
 Table2Initial6: TStringField;
 Table2Institution6: TStringField;
 Table2Order6: TStringField;
 Table4Paper_Number: TIntegerField;
 Table4Title: TStringField;
 Table4Invited: TStringField;
 Table4Accepted: TStringField;
 Table4ContactAuthorNumber: TIntegerField;
 Table4Session: TStringField;
 Table4OrderInSession: TStringField;
 Table4PresentationTime: TStringField;
 Table4Keyword1: TStringField;
 Table4Keyword2: TStringField;
 Table4Keyword3: TStringField;
 Table4Abstract: TMemoField;
 Table4ContactOrder: TStringField;
 Table4FName2: TStringField;
 Table4LName2: TStringField;
 Table4Initial2: TStringField;
 Table4Institution2: TStringField;
 Table4Order2: TStringField;

```

Table4FName3: TStringField;
Table4LName3: TStringField;
Table4Initial3: TStringField;
Table4Institution3: TStringField;
Table4Order3: TStringField;
Table4FName4: TStringField;
Table4LName4: TStringField;
Table4Initial4: TStringField;
Table4Institution4: TStringField;
Table4Order4: TStringField;
Table4FName5: TStringField;
Table4LName5: TStringField;
Table4Initial5: TStringField;
Table4Institution5: TStringField;
Table4Order5: TStringField;
Table4FName6: TStringField;
Table4LName6: TStringField;
Table4Initial6: TStringField;
Table4Institution6: TStringField;
Table4Order6: TStringField;
Table3ANumber: TIntegerField;
Table3LName: TStringField;
Table3FName: TStringField;
Table3Initial: TStringField;
Table3Honorific: TStringField;
Table3Institution: TStringField;
Table3Department: TStringField;
Table3Mailstop: TStringField;
Table3Address_Street: TStringField;
Table3Address_City: TStringField;
Table3Address_State: TStringField;
Table3Address_Zip: TStringField;
Table3Country: TStringField;
Table3Phone_LocalNumber: TStringField;
Table3Phone_FaxNumber: TStringField;
Table3Email: TStringField;
procedure FormCreate(Sender: TObject);
procedure CGIDB1OnSendingHotField(currentRecord: TDataset;
  var s: OpenString);
procedure CGIDB2OnSendingHotField(currentRecord: TDataset;
  var s: OpenString);
procedure CGIDB3SendingHotField(currentRecord: TDataset;
  var s: OpenString);
procedure CGIDB4SendingHotField(currentRecord: TDataset;
  var s: OpenString);

```

```

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  function MixCaseStr(S: string): string;

implementation

{$R *.DFM}

{*****}
function MixCaseStr(S: string): string;
{ This function converts a string to Mixed case.
That is Capital first letter and the rest lower case. }
var
  i: integer;

begin

  if (S[1] >= 'a') and (S[1] <= 'z') then
    Dec(S[1], 32);
  for i := 2 to Length(S) do
    if (S[i] >= 'A') and (S[i] <= 'Z') and (S[i-1] <> ' ') and (S[i-1] <> '-') and (S[i-1] <> 'c') then
      Inc(S[i], 32);
    MixCaseStr := S;

end;
{*****}

procedure TForm1.FormCreate(Sender: TObject);
var
  ANumberlist: Tstringlist; {list of ANumber retrieved from AUTHOR Table which match
query name request}

  i, count: integer;      {loop variable}
  TempANumber: string;
  temp:string;

  dbANumber: string;

```

```
countstr: string;  
dPaperNumber: string;  
dflag: string;  
dtype: string;
```

```
dinvited: string;  
daccepted: string;  
dfnamein: string;  
dlnamein: string;  
dfname: string;  
dlname: string;  
dSession: string;  
dkeyword: string;
```

```
MySelector: integer;  
ANumberint: integer;
```

```
TC: TDataSet;
```

```
begin
```

```
  with CGIEnvData1 do begin
```

```
    {Standard Header}
```

```
    webSiteINIFilename:=paramstr(1);  
    application.onException:=cgiErrorHandler;  
    application.processMessages;  
    createStdout;  
    sendPrologue;
```

```
    {HTML Header}
```

```
    send ('<HTML><HEAD>');  
    SendTitle('Asilomar Signals, Systems, and Computers');  
    send ('</HEAD><BODY bgcolor=FFFFFF>');
```

```
{Determine whether flag is valid}
```

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin  {flag is set}
```

```
  send('<center><H2>Asilomar Conference System Administration</H2></center>');  
  sendhr;
```

{ Get type from search page }

dtype:= getsmallfield ('type');

{ Flag to determine type of search }

{Decide what action is required}

if dtype = 'all' then MySelector := 1;

if dtype = 'article' then MySelector := 2;

if dtype = 'author' then MySelector := 3;

case MySelector of

1 : begin **{ALL}**

with Query1 do begin

prepare; { Optimizes query }

close;

sql.clear;

sql.add('Select * FROM Submissi');

sql.add('Order by Title ');

open;

count:= Recordcount;

countstr:= inttostr (count);

send('<P><center>There are ' +countstr+ ' records in the Submission Table.<P>');

CGIDB1.drawtable; {display all titles}

send ('</center>');

end; **{ with Query1}**

end; **{if all}**

2: begin **{Article}**

dinvited:= getsmallfield ('invited');

daccepted:= getsmallfield ('accepted');

dsession:= getsmallfield ('session');

dkeyword:= getsmallfield ('keyword');

if (dinvited = 'Do not search on this field') and (daccepted = 'Do not search on this field')
and (dsession = 'Do not search on this field') and (dkeyword = 'Do not search on this
field')

then begin

{another ALL search}

with Query1 do begin

prepare; { Optimizes query }

close;

sql.clear;


```

sql.add('Select * FROM Submissi');
sql.add( 'Order by Title ' );
open;
count:= Recordcount;
countstr:= inttostr (count);
send('<P><center>There are '+countstr+ ' records in the Submission Table.<P>');
CGIDB1.drawtable; {display all titles}
send ('</center>');
end; { with Query1}
end {if nothing selected}

```

```

else begin {article}

```

```

{Do the article query requested}
{Find the Titles that match the request}
with Query2 do begin

```

```

close;
sql.clear;
sql.add('Select*FROM Submissi WHERE ');

```

```

{Invited}
if dinvited = 'Yes' then
  sql.add ('Invited = "y"')
else if dinvited = 'No' then
  sql.add ('Invited = "n"');

```

```

if (daccepted <> 'Do not search on this field')and (sql.count = 2) then
  sql.add (' and '); {need an 'and' between statements}

```

```

{Accepted}
if daccepted = 'Yes' then
  sql.add ('accepted = "y"')
else if daccepted = 'No' then
  sql.add ('accepted = "n"');

```

```

{Session}
if (dsession <> 'Do not search on this field') and (sql.count >= 2) then
  sql.add (' and '); {need an 'and' between statements}

```

```

if (dsession <> 'Do not search on this field') then
  sql.add ('Session = "' +dsession+ '"');

```

```

{Keyword}
if (dkeyword <> 'Do not search on this field') and (sql.count >= 2) then

```

```

    sql.add (' and ');    {need an 'and' between statements}

    if (dkeyword <> 'Do not search on this field') then
        sql.add ('(Keyword1 = "' + dkeyword + '" or Keyword2 = "' + dkeyword + '" or Keyword3 =
"' + dkeyword + '"'));

    open;
    send ('<center>');
    if recordcount = 0 then begin
        sendhdr('2', 'There were no records that matched your query!');
        send('You may <a href=" ../cgi-win/ArtlEdit.exe">Do another Query</a> of the
Database. ');
        send ('</center><p>');
        end

    else begin
        CGIDB2.drawTable;
        send ('</center><p>');
        send('Here are the Titles that match your request. Click on the Title Name for the record
you would like ');
        send('to view.<p>If you believe the record you are looking for is under another name,');
        send(' you may <a href=" ../cgi-win/ArtlEdit.exe">Do another Query</a> of the
Database. ');
        end;

    end; {query2}
    end; {else}
    end; {case article}

3: begin {author}

    {Get Author Name}
    dFNamein:= getsmallfield ('fname');
    dfname:= MixCaseStr(dFNamein);

    dLNamein:= getsmallfield ('lname');
    dlname:= MixCaseStr(dLNamein);

    {Find out if name is in contact authors. Only the last name is required,
but the first name will help narrow the search. }

    if (dlname <> "") then begin

```

{Search for name in AUTHOR Table}

with Query3 do begin

close;

sql.clear;

{If both the first and last name is given, search for match on both}

if (dfname <> "") then

sql.add('Select * FROM Author WHERE FName = "' + dfname + '" and LName = "' + dlname + '"')

{Only the last name is given, so search only on it}

else

sql.add('Select ANumber FROM Author WHERE LName = "' + dlname + '"');

{execute the Query}

open;

{Display results}

send('<P><center>');

If Recordcount > 0 then begin

CGIDB3.DrawTable;

send('<P>If the Author sought was not the Contact Author for the paper, their name will not appear');

send(' on this list. To continue the search, click on the "Not a Contact Author" button below.');

send('<P><FORM ACTION=" ../cgi-win/ArtlAuth.exe" METHOD="POST">');

send('<INPUT TYPE="hidden" Name="flag" Value="1">');

send('<INPUT TYPE="hidden" Name="getauthor" Value="NotContactAuthor">');

send('<INPUT TYPE="hidden" Name="fname" Value="' + dfname + '">');

send('<INPUT TYPE="hidden" Name="lname" Value="' + dlname + '">');

send('<P><CENTER><INPUT TYPE="submit" Value=" Not a Contact Author "></CENTER></FORM>');

end

else begin **{The name is not a ontact Author so do the search on Other Authers only}**

with Query4 do begin

close;

sql.clear;

sql.add('Select*FROM Submissi WHERE ');

{If there is a First name given, then search for first and last name match in Other Authors}

```

        if dfname <> " then
            sql.add ('(LName2 = "" +dlname+ "" and FName2 ="" +dfname+ "") or (LName3 =
""
            +dlname+ "" and FName3 ="" +dfname+ "") or (LName4 = ""+dlname+ "" and
FName4 =""
            +dfname+ "") or (LName5 = "" +dlname+ "" and FName5 = "" +dfname+ "") or
(LName6 = "" +dlname+ "" and FName6 ="" +dfname+ ""))

```

{Otherwise, search only for last name match in Other Authors}

```

        else sql.add ('(LName2 = "" +dlname+ "" or LName3 = ""+dlname+ "" or LName4 =
""
            +dlname+ "" or LName5 = "" +dlname+ "" or LName6 = ""+dlname+ ""));

```

open;

If Recordcount >0 then CGIDB4.DrawTable

```

        else send("There are no Authors in the database with the name <B>' + dfname + '' +
dlname + '</B>');

```

```

        send('<P><FORM ACTION=" ../cgi-win/ArtlEdit.exe" METHOD="POST">');
        send('<INPUT TYPE="hidden" Name="flag" Value="1">');
        send('<P><CENTER><INPUT TYPE="submit" Value="Query another
Article"></CENTER></FORM>');
        end; {with query4}

```

end; {if recordcount > 0}

end; {withQuery3}

end {If dlname <>" then}

else begin

```

        sendhdr('2', 'You must enter a last name!');
        send('<P><FORM ACTION=" ../cgi-win/ArtlEdit.exe" METHOD="POST">');
        send('<INPUT TYPE="hidden" Name="flag" Value="1">');
        send('<P><CENTER><INPUT TYPE="submit" Value="Query another
Article"></CENTER></FORM>');

```

end; {If dlname <>" else}

end; {case author}

end; {case}

{ HTML Footer}

```

    sendhr;
    send('<P><FORM ACTION="../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );
end {flag is valid}

else begin    {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF="../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC="../prototype/asil2b0.gif" align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

    end; {flag **incorrect**}

end; {with cgiEnvData1 do}

end; {Procedure TForm1.FormCreate}

procedure TForm1.CGIDB1OnSendingHotField(currentRecord: TDataset;
var s: OpenString);
begin
    s:='<A HREF="../cgi-win/ArtlView.exe?'+
currentrecord.fieldByName('Paper_Number').AsString + "'>
+ currentrecord.FieldByName('Title').AsString + '</A>';

```

```

end; {Procedure TForm1.CGIDB1OnSendingHotField}

procedure TForm1.CGIDB2OnSendingHotField(currentRecord: TDataset;
  var s: OpenString);
begin
  s:='<A HREF="../cgi-win/ArtlView.exe?'+
currentrecord.fieldbyname('Paper_Number').AsString + '>'
  + currentrecord.Fieldbyname('Title').AsString + '</A>';

end; {Procedure TForm1.CGIDB2OnSendingHotField}

procedure TForm1.CGIDB3SendingHotField(currentRecord: TDataset;
  var s: OpenString);
begin
  s:='<A HREF="../cgi-win/ArtlAuth.exe?'+ currentrecord.fieldbyname('ANumber').AsString +
">'
  + currentrecord.Fieldbyname('ANumber').AsString + '</A>';

end; {Procedure TForm1.CGIDB3OnSendingHotField}

procedure TForm1.CGIDB4SendingHotField(currentRecord: TDataset;
  var s: OpenString);
begin
  s:='<A HREF="../cgi-win/ArtlView.exe?'+
currentrecord.fieldbyname('Paper_Number').AsString + '>'
  + currentrecord.Fieldbyname('Title').AsString + '</A>';

end; {Procedure TForm1.CGIDB4OnSendingHotField}

end.

```

unit Art_auth;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type

TForm1 = class(TForm)
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 DataSource1: TDataSource;
 Query1: TQuery;
 Table1: TTable;
 Query1Paper_Number: TIntegerField;
 Query1Title: TStringField;
 Query2: TQuery;
 DataSource2: TDataSource;
 Table3: TTable;
 Table2: TTable;
 DataSource3: TDataSource;
 Query2Paper_Number: TIntegerField;
 Query2Title: TStringField;
 CGIDB2: TCGIDB;
 Table1Paper_Number: TIntegerField;
 Table1Title: TStringField;
 Table1Invited: TStringField;
 Table1Accepted: TStringField;
 Table1ContactAuthorNumber: TIntegerField;
 Table1Session: TStringField;
 Table1OrderInSession: TStringField;
 Table1PresentationTime: TStringField;
 Table1Keyword1: TStringField;
 Table1Keyword2: TStringField;
 Table1Keyword3: TStringField;
 Table1Abstract: TMemoField;
 Table1ContactOrder: TStringField;
 Table1FName2: TStringField;
 Table1LName2: TStringField;
 Table1Initial2: TStringField;
 Table1Institution2: TStringField;
 Table1Order2: TStringField;

Table1FName3: TStringField;
 Table1LName3: TStringField;
 Table1Initial3: TStringField;
 Table1Institution3: TStringField;
 Table1Order3: TStringField;
 Table1FName4: TStringField;
 Table1LName4: TStringField;
 Table1Initial4: TStringField;
 Table1Institution4: TStringField;
 Table1Order4: TStringField;
 Table1FName5: TStringField;
 Table1LName5: TStringField;
 Table1Initial5: TStringField;
 Table1Institution5: TStringField;
 Table1Order5: TStringField;
 Table1FName6: TStringField;
 Table1LName6: TStringField;
 Table1Initial6: TStringField;
 Table1Institution6: TStringField;
 Table1Order6: TStringField;
 Table3Paper_Number: TIntegerField;
 Table3Title: TStringField;
 Table3Invited: TStringField;
 Table3Accepted: TStringField;
 Table3ContactAuthorNumber: TIntegerField;
 Table3Session: TStringField;
 Table3OrderInSession: TStringField;
 Table3PresentationTime: TStringField;
 Table3Keyword1: TStringField;
 Table3Keyword2: TStringField;
 Table3Keyword3: TStringField;
 Table3Abstract: TMemoField;
 Table3ContactOrder: TStringField;
 Table3FName2: TStringField;
 Table3LName2: TStringField;
 Table3Initial2: TStringField;
 Table3Institution2: TStringField;
 Table3Order2: TStringField;
 Table3FName3: TStringField;
 Table3LName3: TStringField;
 Table3Initial3: TStringField;
 Table3Institution3: TStringField;
 Table3Order3: TStringField;
 Table3FName4: TStringField;
 Table3LName4: TStringField;


```

Table3Initial4: TStringField;
Table3Institution4: TStringField;
Table3Order4: TStringField;
Table3FName5: TStringField;
Table3LName5: TStringField;
Table3Initial5: TStringField;
Table3Institution5: TStringField;
Table3Order5: TStringField;
Table3FName6: TStringField;
Table3LName6: TStringField;
Table3Initial6: TStringField;
Table3Institution6: TStringField;
Table3Order6: TStringField;
procedure FormCreate(Sender: TObject);
procedure CGIDB1SendingHotField(currentRecord: TDataset;
  var s: OpenString);
procedure CGIDB2SendingHotField(currentRecord: TDataset;
  var s: OpenString);
procedure CGIDB3SendingHotField(currentRecord: TDataset;
  var s: OpenString);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  dflag: string;           {valid program call}

  dANumber, dANumberstr: string;
  dFName : string;
  dLName : string;

  dGetAuthor: string;

  dANumberint, count1, count2: integer;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin

```

with CGIEnvData1 do begin

{Standard Header}

```
webSiteINIFilename:=paramstr(1);
application.onException:=cgiErrorHandler;
application.processMessages;
createStdout;
sendPrologue;
```

{HTML Header}

```
send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

{Determine whether flag is valid}

```
dflag:= getsmallfield ('flag');
```

if (dflag = '1') then begin **{flag is set}**

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;
```

{Find Author name to use in search for article}

{Get fields from search page.

If a contact author was not found, then will get the string 'NotContactAuthor' and will proceed with a search of the Other Authors

If a contact author was found, then will get the query string with the 'ANumber'. }

try

{ Get the FORM information }

```
dGetAuthor:= getsmallfield ('GetAuthor');
```

```
dfname:= getsmallfield ('fname');
```

```
dlname:= getsmallfield ('lname');
```

finally

{Get the ANumber string from the address}

```
dANumber := CGIQueryString^;
```

```
end;
```

if dGetAuthor <> cginotfound then begin

{There was NOT a Contact Author match}

{ if dGetAuthor = 'NotContactAuthor' then begin }

{Find the Titles that match the name in Other Authors only}
with Query1 do begin

{query on names in Other Authors only}
close;
sql.clear;
sql.add('Select*FROM Submissi WHERE ');

{If there is a First name given, then search for first and last name match in Other Authors}

if dfname <> "" then
sql.add ('(LName2 = "" + dlname+ "" and FName2 = "" + dfname+ "") or (LName3 = ""
+ dlname+ "" and FName3 = "" + dfname+ "") or (LName4 = "" + dlname+ "" and FName4
= ""
+ dfname+ "") or (LName5 = "" + dlname+ "" and FName5 = "" + dfname+ "") or (LName6
= ""
+ dlname+ "" and FName6 = "" + dfname+ ""))

{Otherwise, search only for last name match in Other Authors}
else sql.add ('(LName2 = "" + dlname+ "" or LName3 = "" + dlname+ "" or LName4 =
"" + dlname+ "" or LName5 = ""
+ dlname+ "" or LName6 = "" + dlname+ ""));

{execute the query}
open;

{Display the results}
if recordcount > 0 then begin

send('<center>');
CGIDB1.DrawTable;
send('</center>');

send('<P><FORM ACTION=" ../cgi-win/Artlledit.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Search for another
Article"></CENTER></FORM>');

end

else begin {No Authors found}

```

        send('There are no Articles in the database with the Author <B>' + dfname + '' +
dlname + '</B>');
        send('<P><FORM ACTION="../../cgi-win/ArtlEdit.exe" METHOD="POST">');
        send('<INPUT TYPE="hidden" Name="flag" Value="1">');
        send('<P><CENTER><INPUT TYPE="submit" Value="Query another
Article"></CENTER></FORM>');
        end; {if}
    close;
    end; {with query1}

end {ifdGetAuthor = 'NotContactAuthor' then}

else begin {There was a contact author match}

    {Find the Titles that match the name in Other Authors *and* in
ContactAuthorNumber}

    {Get name of author}
    with table2 do begin

        open;
        first;
        while fieldbyname('ANumber').asstring <> dANumber do
            next;

        dfname:= fieldbyname ('FName').asstring;
        dlname:= fieldbyname ('LName').asstring;
        close;
        end; {with tabl2}

    {Search for Titles with contactauthornumber}
    with Query2 do begin

        {execute the query}
        dANumberint:= strtoint (dANumber);
        Query2.ParamByName('ANumber').AsInteger := dANumberint;
        open;

        count1:= recordcount;
        {Display the results}
        if recordcount > 0 then begin

            send('<P><h4>The Author ' + dfname + '' + dlname + ' was a Contact Author for these
Articles.</h4>');

```

```

send('<center>');
CGIDB2.DrawTable;
send('</center>');
end;

```

end; {With Query2}

with Query1 do begin

```

{query on names in Other Authors only}
close;
sql.clear;
sql.add('Select*FROM Submissi WHERE ');

```

```

sql.add ('(LName2 = "" + dlname+ "" and FName2 ="" +dfname+ "" ) or (LName3 = ""
+dlname+ "" and FName3 ="" +dfname+ "" ) or (LName4 = ""+dlname+ "" and FName4
= ""
+dfname+ "" ) or (LName5 = "" +dlname+ "" and FName5 = "" +dfname+ "" ) or (LName6
= ""
+dlname+ "" and FName6 ="" +dfname+ ""));

```

```

{execute the query}
open;
count2:= recordcount;

```

```

{Display the results}
if recordcount > 0 then begin

```

```

    send('<P><h4>The Author ' +dfname+ ' ' +dlname+ ' was not the Contact Author for
these Articles.</h4>');
    send('<center>');
    CGIDB1.DrawTable;
    send('</center>');
    end;

```

end; {with Query1}

```

if (count1 = 0) and (count2 = 0) then begin

```

```

    send('There are no Articles in the database with the Author <B>' + dfname + ' ' +
dlname + '</B>');
    send('<P><FORM ACTION=" ../cgi-win/ArtlEdit.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');

```

```

        send('<P><CENTER><INPUT TYPE="submit" Value="Query another
Article"></CENTER></FORM>');
        end; {if}

end; {else}

{ HTML Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin Options
page"></CENTER></FORM>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end {flag is set}

else begin    {Password **incorrect**}

send('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send('Conference On-line Submission Page</A>');
send('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {Password **incorrect**}

end; {with CGIEnvData}

end; {Tform1}

```

```

procedure TForm1.CGIDB1SendingHotField(currentRecord: TDataset;
  var s: OpenString);
begin
  s:='<A HREF=" ../cgi-win/ArtlView.exe?"'+
currentrecord.fieldbyName('Paper_Number').AsString + ">"'
  + currentrecord.FieldbyName('Title').AsString + '</A>';

end;

```

```

procedure TForm1.CGIDB2SendingHotField(currentRecord: TDataset;
  var s: OpenString);
begin
  s:='<A HREF=" ../cgi-win/ArtlView.exe?"'+
currentrecord.fieldbyName('Paper_Number').AsString + ">"'
  + currentrecord.FieldbyName('Title').AsString + '</A>';

end;

```

```

procedure TForm1.CGIDB3SendingHotField(currentRecord: TDataset;
  var s: OpenString);
begin
  s:='<A HREF=" ../cgi-win/ArtlView.exe?"'+
currentrecord.fieldbyName('Paper_Number').AsString + ">"'
  + currentrecord.FieldbyName('Title').AsString + '</A>';

end;

```

```

end.

```

View the Article: **ArtlView.exe**

```
unit Art_view;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Table1: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    Table2: TTable;
    Query1: TQuery;
    DataSource2: TDataSource;
    Keywords: TTable;
    KeywordsNumber: TIntegerField;
    KeywordsName: TStringField;
    Query1Number: TIntegerField;
    Query1Name: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  theabstract : TStringList;
  i: integer;

  dPaper_Number: string;
  dbOption: string;

  dbtitle: string;
  dbinvited: string;
  dbaccepted: string;
  dbANumber: string;
```


dbSession: string;
dbOrderInSession,
dbPresentationTime: string;

dbkeyword1: string;
dbkeyword2: string;
dbkeyword3: string;
dbabstract: string;
dbcontactOrder: string;

dbfname2: string;
dblname2: string;
dbinitial2: string;
dbinstitution2: string;
dborder2: string;

dbfname3: string;
dblname3: string;
dbinitial3: string;
dbinstitution3: string;
dborder3: string;

dbfname4: string;
dblname4: string;
dbinitial4: string;
dbinstitution4: string;
dborder4: string;

dbfname5: string;
dblname5: string;
dbinitial5: string;
dbinstitution5: string;
dborder5: string;

dbfname6: string;
dblname6: string;
dbinitial6: string;
dbinstitution6: string;
dborder6: string;

dbFName : string;
dbLName : string;
dbHonorific : string;

implementation

```
{ $R *.DFM }
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
  with CGIEnvData1 do
```

```
    begin
```

```
      webSiteINIFilename:=paramstr(1);
```

```
      application.OnException:=cgErrorHandler;
```

```
      application.ProcessMessages;
```

```
      createStdout;
```

```
      sendPrologue;
```

```
  { Get the Paper Number Selected }
```

```
  dPaper_Number:= CGIQueryString^;
```

```
  { Get the Paper Record from the SUBMISSION Table }
```

```
  with Table1 do begin
```

```
    { Move to proper record }
```

```
    open;
```

```
    first;
```

```
    while fieldByName('Paper_Number').asString < dPaper_Number do
```

```
      next;
```

```
    { Retrieve record }
```

```
    dbtitle:= fieldByName ('title').asString;
```

```
    dbinvited:= fieldByName ('invited').asString;
```

```
    dbaccepted:= fieldByName ('accepted').asString;
```

```
    dbANumber:= fieldByName ('ContactAuthorNumber').asString;
```

```
    dbSession:= fieldByName ('Session').asString;
```

```
    dbOrderInSession:= fieldByName ('OrderInSession').asString;
```

```
    dbPresentationTime:= fieldByName ('PresentationTime').asString;
```

```
    dbkeyword1:= fieldByName ('keyword1').asString;
```

```
    dbkeyword2:= fieldByName ('keyword2').asString;
```

```
    dbkeyword3:= fieldByName ('keyword3').asString;
```

```
    CGIDB1.memoToStringList (fieldbyname ('Abstract'), theabstract);
```

```
    dbcontactOrder:= fieldByName ('contactOrder').asString;
```

```
    dborder2:= fieldByName ('order2').asString;
```

```
dbfname2:= fieldByName ('fname2').asstring;
dbinitial2:= fieldByName ('initial2').asstring;
dblname2:= fieldByName ('lname2').asstring;
dbinstitution2:= fieldByName ('institution2').asstring;
```

```
dborder3:= fieldByName ('order3').asstring;
dbfname3:= fieldByName ('fname3').asstring;
dbinitial3:= fieldByName ('initial3').asstring;
dblname3:= fieldByName ('lname3').asstring;
dbinstitution3:= fieldByName ('institution3').asstring;
```

```
dborder4:= fieldByName ('order4').asstring;
dbfname4:= fieldByName ('fname4').asstring;
dbinitial4:= fieldByName ('initial4').asstring;
dblname4:= fieldByName ('lname4').asstring;
dbinstitution4:= fieldByName ('institution4').asstring;
```

```
dborder5:= fieldByName ('order5').asstring;
dbfname5:= fieldByName ('fname5').asstring;
dbinitial5:= fieldByName ('initial5').asstring;
dblname5:= fieldByName ('lname5').asstring;
dbinstitution5:= fieldByName ('institution5').asstring;
```

```
dborder6:= fieldByName ('order6').asstring;
dbfname6:= fieldByName ('fname6').asstring;
dbinitial6:= fieldByName ('initial6').asstring;
dblname6:= fieldByName ('lname6').asstring;
dbinstitution6:= fieldByName ('institution6').asstring;
end;
```

{ Get Author Name information from the AUTHOR Table }

with Table2 do begin

{ Move to proper record }

open;

first;

while fieldByName('ANumber').asstring <> dbANumber do
 next;

{ Retrieve Author Name information }

dbFName := fieldByName('FName').AsString;

dbLName := fieldByName('LName').AsString;

dbHonorific := fieldByName('Honorific').AsString;

close;

```

end;

{ Send the html page }
send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
send ('<center><H2>Asilomar Conference System Administration</H2></center><HR>');

send ('<P>Here is the paper submitted by ' + dbHonorific + ' ' + dbFName + ' ' + dbLName + '. ');

send ('<P><FORM METHOD="POST" ACTION="..cgi-win/ArtlPost.exe ">');
send ('<INPUT TYPE="hidden" Name="flag" Value="1">');
send ('<INPUT TYPE="hidden" Name="Paper_Number" Value="" + dPaper_Number + ">');
send ('<B>Title: </B><INPUT NAME="title" Size="80" TYPE="text" Value="" + dbtitle
+ ">');
send ('<P><B>Invited: </B><INPUT NAME="Invited" Size="2" TYPE="text" Value="" +
dbinvited + ">');
send ('<B>Accepted: </B><INPUT NAME="Accepted" Size="2" TYPE="text" Value="" +
dbaccepted + ">');
send ('<B>Session: </B><INPUT NAME="Session" Size="5" TYPE="text" Value="" +
dbSession + ">');
send ('<B>Order in Session: </B><INPUT NAME="OrderInSession" Size="3" TYPE="text"
Value="" + dbOrderInSession + ">');
send ('<B>Presentation Time: </B><INPUT NAME="PresentationTime" Size="8"
TYPE="text" Value="" + dbPresentationTime + ">');

{Pull Keywords from KEYWORD Table }
with Query1 do begin
close;
sql.clear;
sql.add('Select * FROM Keyword ');
open;

{Move records from query result to Keyword #1 Pull-down menu}
Keywords.open;
Keywords.first;

send ('<P><B>Keyword #1: </B><SELECT NAME="Keyword1">');
send ('<OPTION SELECTED> ' + dbkeyword1 + ');

while not Keywords.EOF do begin
dbOption:= Keywords.fieldByName('Name').AsString;
send ('<OPTION> ' + dbOption + ');
Keywords.next;

```

```
end; {for all records in the query result}
send('</SELECT>');
```

{Move records from query result to Keyword #2 Pull-down menu}

```
Keywords.first;
```

```
send('<P><B>Keyword #2: </B><SELECT NAME="Keyword2">');
if dbkeyword2 = 'None' then send('<OPTION> None')
  else send('<OPTION SELECTED> ' + dbkeyword2 + "');
```

```
while not Keywords.EOF do begin
  dbOption:= Keywords.fieldByName('Name').AsString;
  send('<OPTION> ' + dbOption + " ');
  Keywords.next;
end; {for all records in the query result}
send('</SELECT>');
```

{Move records from query result to Keyword #3 Pull-down menu}

```
Keywords.first;
```

```
send('<P><B>Keyword #3: </B><SELECT NAME="Keyword3">');
if dbkeyword3 = 'None' then send('<OPTION> None')
  else send('<OPTION SELECTED> ' + dbkeyword3 + "');
```

```
while not Keywords.EOF do begin
  dbOption:= Keywords.fieldByName('Name').AsString;
  send('<OPTION> ' + dbOption + " ');
  Keywords.next;
end; {for all records in the query result}
send('</SELECT>');
```

```
Keywords.close;
end; {query1}
```

```
send ('<P><B>Abstract: </B><p><TEXTAREA NAME="abstract" COLS=70 Rows=10>');
for i := 0 to theabstract.count - 1 do
  send( theabstract.strings[i] );
send ('</TEXTAREA>');
```

```
theabstract.free; {release the memory held by 'theabstract'}
```

```
sendhdr ('1', 'Other Author Information');
send ('<B>Name Position of the contact author:');
```

```
send ('<INPUT NAME="contactOrder" Size="2" TYPE="text" Value="" + dbcontactOrder
+ ">');
```

```
sendhr;
```

```
send ('<B>Name Position: </B><INPUT NAME="order2" Size="2" TYPE="text" Value="" +
dborder2 + ">');
```

```
send ('<B>First Name: </B><INPUT NAME="fname2" Size="25" TYPE="text" Value="" +
dbfname2 + ">');
```

```
send ('<B>Middle Initial: </B><INPUT NAME="initial2" Size="1" TYPE="text" Value="" +
dbinitial2 + ">');
```

```
send ('<BR><B>Last Name: </B><INPUT NAME="lname2" Size="25" TYPE="text"
Value="" + dblname2 + ">');
```

```
send ('<BR><B>Institution:</B><INPUT NAME="institution2" Size="60" TYPE="text"
Value="" + dbinstitution2 + ">');
```

```
sendhr;
```

```
send ('<B>Name Position: </B><INPUT NAME="order3" Size="2" TYPE="text" Value="" +
dborder3 + ">');
```

```
send ('<B>First Name: </B><INPUT NAME="fname3" Size="25" TYPE="text" Value="" +
dbfname3 + ">');
```

```
send ('<B>Middle Initial: </B><INPUT NAME="initial3" Size="1" TYPE="text" Value="" +
dbinitial3 + ">');
```

```
send ('<BR><B>Last Name: </B><INPUT NAME="lname3" Size="25" TYPE="text"
Value="" + dblname3 + ">');
```

```
send ('<br><B>Institution:</B><INPUT NAME="institution3" Size="60" TYPE="text"
Value="" + dbinstitution3 + ">');
```

```
sendhr;
```

```
send ('<B>Name Position: </B><INPUT NAME="order4" Size="2" TYPE="text" Value="" +
dborder4 + ">');
```

```
send ('<B>First Name: </B><INPUT NAME="fname4" Size="25" TYPE="text" Value="" +
dbfname4 + ">');
```

```
send ('<B>Middle Initial: </B><INPUT NAME="initial4" Size="1" TYPE="text" Value="" +
dbinitial4 + ">');
```

```
send ('<BR><B>Last Name: </B><INPUT NAME="lname4" Size="25" TYPE="text"
Value="" + dblname4 + ">');
```

```
send ('<BR><B>Institution:</B><INPUT NAME="institution4" Size="60" TYPE="text"
Value="" + dbinstitution4 + ">');
```

```
sendhr;
```

```
send ('<B>Name Position: </B><INPUT NAME="order5" Size="2" TYPE="text" Value="" +
dborder5 + ">');
```

```
send ('<B>First Name: </B><INPUT NAME="fname5" Size="25" TYPE="text" Value="" +
dbfname5 + ">');
```

```
send ('<B>Middle Initial: </B><INPUT NAME="initial5" Size="1" TYPE="text" Value="" +
```

```

dbinitial5 +"">');
    send ('<BR><B>Last Name: </B><INPUT NAME="lname5" Size="25" TYPE="text"
Value="" + dbname5 +"">');
    send ('<BR><B>Institution:</B><INPUT NAME="institution5" Size="60" TYPE="text"
Value="" + dbinstitution5 +"">');

    sendhr;
    send ('<B>Name Position: </B><INPUT NAME="order6" Size="2" TYPE="text" Value="" +
dborder6 +"">');
    send ('<B>First Name: </B><INPUT NAME="fname6" Size="25" TYPE="text" Value="" +
dbfname6 +"">');
    send ('<B>Middle Initial: </B><INPUT NAME="initial6" Size="1" TYPE="text" Value="" +
dbinitial6 +"">');
    send ('<BR><B>Last Name: </B><INPUT NAME="lname6" Size="25" TYPE="text"
Value="" + dbname6 +"">');
    send ('<BR><B>Institution:</B><INPUT NAME="institution6" Size="60" TYPE="text"
Value="" + dbinstitution6 +"">');

    send ('<P><CENTER><INPUT TYPE="submit" Value="Record these changes to Article
Information"> <INPUT TYPE="reset"">');
    send (' VALUE="Reset to Original Values"></CENTER></FORM>');

    sendhr;
    send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application ); { don't leave form around }

    end; {with CGIEnvData1}

end; {with FormCreate}

end.

```

Post the Article: **ArtlPost.exe**

```
unit Art_post;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Table1: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    Table1Paper_Number: TIntegerField;
    Table1Title: TStringField;
    Table1Invited: TStringField;
    Table1Accepted: TStringField;
    Table1ContactAuthorNumber: TIntegerField;
    Table1Session: TStringField;
    Table1OrderInSession: TStringField;
    Table1PresentationTime: TStringField;
    Table1Keyword1: TStringField;
    Table1Keyword2: TStringField;
    Table1Keyword3: TStringField;
    Table1ContactOrder: TStringField;
    Table1FName2: TStringField;
    Table1LName2: TStringField;
    Table1Initial2: TStringField;
    Table1Institution2: TStringField;
    Table1Order2: TStringField;
    Table1FName3: TStringField;
    Table1LName3: TStringField;
    Table1Initial3: TStringField;
    Table1Institution3: TStringField;
    Table1Order3: TStringField;
    Table1FName4: TStringField;
    Table1LName4: TStringField;
    Table1Initial4: TStringField;
    Table1Institution4: TStringField;
    Table1Order4: TStringField;
    Table1FName5: TStringField;
    Table1LName5: TStringField;
```



```

Table1Initial5: TStringField;
Table1Institution5: TStringField;
Table1Order5: TStringField;
Table1FName6: TStringField;
Table1LName6: TStringField;
Table1Initial6: TStringField;
Table1Institution6: TStringField;
Table1Order6: TStringField;
Table1Abstract: TMemoField;
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  theabstract : TStringList;
  dANumber: string;
  dANumberint: integer;
  dPaper_Number: string;
  dtitle: string;
  dinvited: string;
  daccepted: string;
  dsession: string;
  dOrderInSession, dPresentationTime: string;

  dkeyword1: string;
  dkeyword2: string;
  dkeyword3: string;
  dabstract: string;
  dcontactOrder: string;

  dfname2: string;
  dlname2: string;
  dinitial2: string;
  dinstitution2: string;
  dorder2: string;

  dfname3: string;
  dlname3: string;
  dinitial3: string;
  dinstitution3: string;
  dorder3: string;

```

dfname4: string;
dlname4: string;
dinitial4: string;
dinstitution4: string;
dorder4: string;

dfname5: string;
dlname5: string;
dinitial5: string;
dinstitution5: string;
dorder5: string;

dfname6: string;
dlname6: string;
dinitial6: string;
dinstitution6: string;
dorder6: string;

dbFName : string;
dbLName : string;
dbHonorific : string;
dbEmail : string;

addinfo: string;

dflag: string; **{valid program call}**

implementation

{ \$R *.DFM }

procedure TForm1.FormCreate(Sender: TObject);
begin
 with CGIEnvData1 do begin

{Initialize HTML page }
 webSiteINIFilename:=paramstr(1);
 application.onException:=cgiErrorHandler;
 application.processMessages;

 createStdout;
 sendPrologue;

```
send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

{Determine whether flag is valid}

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;
```

{ Get data }

```
dPaper_Number:= getsmallfield ('Paper_Number');
```

```
dtitle:= getsmallfield ('title');
dinited:= getsmallfield ('invited');
daccepted:= getsmallfield ('accepted');
dsession:= getsmallfield ('session');
dOrderInSession:= getsmallfield ('OrderInSession');
dPresentationTime:= getsmallfield ('PresentationTime');
dkeyword1:= getsmallfield ('keyword1');
dkeyword2:= getsmallfield ('keyword2');
dkeyword3:= getsmallfield ('keyword3');
```

```
theabstract := TStringList.create;
theabstract.clear;
```

```
CGIEnvData1.getTextArea( 'abstract', theabstract);
```

```
dcontactOrder:= getsmallfield ('contactOrder');
```

```
dorder2:= getsmallfield ('order2');
dfname2:= getsmallfield ('fname2');
dlname2:= getsmallfield ('lname2');
dinitial2:= getsmallfield ('initial2');
dinstitution2:= getsmallfield ('institution2');
```

```
dorder3:= getsmallfield ('order3');
dfname3:= getsmallfield ('fname3');
dlname3:= getsmallfield ('lname3');
dinitial3:= getsmallfield ('initial3');
dinstitution3:= getsmallfield ('institution3');
```

```

dorder4:= getsmallfield ('order4');
dfname4:= getsmallfield ('fname4');
dlname4:= getsmallfield ('lname4');
dinitial4:= getsmallfield ('initial4');
dinstitution4:= getsmallfield ('institution4');

```

```

dorder5:= getsmallfield ('order5');
dfname5:= getsmallfield ('fname5');
dlname5:= getsmallfield ('lname5');
dinitial5:= getsmallfield ('initial5');
dinstitution5:= getsmallfield ('institution5');

```

```

dorder6:= getsmallfield ('order6');
dfname6:= getsmallfield ('fname6');
dlname6:= getsmallfield ('lname6');
dinitial6:= getsmallfield ('initial6');
dinstitution6:= getsmallfield ('institution6');

```

```

{ Place Data into SUBMISSION Table}
with Table1 do begin

```

```

    { Move to proper record}
    open;
    first;
    if fieldbyname('Paper_Number').asstring <> dPaper_Number then
    repeat
    next;
    until fieldbyname('Paper_Number').asstring = dPaper_Number;

```

```

    {Place edited submission data to SUBMISSION Table}
    edit;
    SetFields([nil, dtitle, dinvited, daccepted, nil, dsession, dOrderInSession,
dPresentationTime, dkeyword1, dkeyword2,
    dkeyword3, nil, dcontactOrder, dFName2, dLName2, dInitial2, dInstitution2, dOrder2,
dFName3, dLName3,
    dInitial3, dInstitution3, dOrder3, dFName4, dLName4, dInitial4, dInstitution4, dOrder4,
    dFName5, dLName5, dInitial5, dInstitution5, dOrder5, dFName6, dLName6, dInitial6,
dInstitution6,
    dOrder6]);

    edit;
    CGIDB1.StringlistToMemo(theabstract, fieldbyname('Abstract'));
    theabstract.free;

```

```
post;
close;
end; {with Table1}
```

```
{ Send the Final page }
```

```
sendhdr ('2', 'Article Successfully Edited');
```

```
send ('<P>The article titled <B>' + dtitle + '</B> has been updated in the Database.<P>');
```

```
send('<P><FORM ACTION="..\cgi-win/ArtlEdit.exe" METHOD="POST">');
```

```
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
```

```
send('<P><CENTER><INPUT TYPE="submit" Value="Edit another
Article"></CENTER></FORM><P>');
```

```
sendhr;
```

```
send('<P><FORM ACTION="..\cgi-win/Adminpwd.exe" METHOD="POST">');
```

```
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
```

```
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
```

```
send ('</BODY></HTML>');
```

```
CGIEnvData1.closeStdout;
```

```
closeApp( application );
```

```
end {flag is set}
```

```
else begin {flag **incorrect**}
```

```
send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
```

```
sendhr;
```

```
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
```

```
send ('<P>Please ensure that you are authorized to access this information.');
```

```
send ('<P>If you made an error, then please <A HREF="..\prototype/admin20.htm">try to login
again.</A>');
```

```
send ('<P>If you are not authorized to access this information, please note that a log is');
```

```
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
```

```
send ('prosecution for your actions.<BR>');
```

```
send ('<P><hr><P><IMG SRC="..\prototype/asil2b0.gif" align=left alt="Asilomar facility">');
```

```
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
```

```
send ('Conference On-line Submission Page</A>');
```

```
send ('</BODY></HTML>');
```

```
closeStdout;  
closeApp( application );  
  
    end; {flag **incorrect**}  
  
    end; {with CGIEnvData1}  
  
    end; {FormCreate}  
  
end. {Application}
```

Choose Session: **Session.exe**

```
unit Session;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DBTables, DB;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Query1: TQuery;
    Sessions: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    SessionsNumber: TIntegerField;
    SessionsNeumonic: TStringField;
    Query1Number: TIntegerField;
    Query1Neumonic: TStringField;
    procedure FormCreate(Sender: TObject);
    procedure CGIDB1SendingHotField(currentRecord: TDataset;
      var s: OpenString);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  dflag: string;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    {Standard Header}
    webSiteINIFilename:=paramstr(1);
```

```
application.onException:=cgiErrorHandler;  
application.processMessages;
```

```
createStdout;  
sendPrologue;  
send('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send('</HEAD><BODY bgcolor=FFFFFF>');
```

{Determine whether flag is valid}

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

{send instruction;}

```
send('<center>These are the current Session Neumonics.<br>');  
send('To edit a single Session Neumonic, simply click on the Neumonic that you would like  
to change.<BR>');  
send(' Other options are available at the bottom of the page.<BR>');
```

{Retrieve current Session Information}

```
with Query1 do begin
```

```
close;  
sql.clear;  
sql.add('Select*FROM Session ');  
open;
```

```
if recordcount > 0 then begin  
CGIDB1.DrawTable;  
send('</center>');  
end {if recordcount > 0}
```

```
else  
sendhdr ('2','There are no Sessions in the Database!');
```

```
end; {with Query1}
```


{Button to Add keywords}

```
send('<P><FORM ACTION="../cgi-win/SessAdd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Add a New
Session"></CENTER></FORM>');
```

{Button to Delete keywords}

```
send('<P><FORM ACTION="../cgi-win/SessDel.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Delete
Session(s)"></CENTER></FORM>');
```

{Standard Footer}

```
sendhr;
send('<P><FORM ACTION="../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}
```

else begin {flag **incorrect**}

```
send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF="../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');
```

```
send ('<P><hr><P><IMG SRC="../prototype/asil2b0.gif" align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );
```

end; {flag **incorrect**}

```
send ('</BODY></HTML>');
closeStdout;
```

```

closeApp( application );
end; {with CGIEnvData1}

end; {FormCreate}

procedure TForm1.CGIDB1SendingHotField(currentRecord: TDataset;
var s: OpenString);
begin
    {Hotlink for single session edit}
    s:='<A HREF=" ../cgi-win/SessEdit.exe?'+ currentrecord.fieldbyName('Number').AsString +
">'
    + currentrecord.FieldbyName('Neumonic').AsString + '</A>';

end;

end.

```

Edit a Session: SessEdit.exe

```
unit Ses_edit;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Sessions: TTable;
    SessionsNumber: TIntegerField;
    SessionsNeumonic: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  SessionNumber, Session: string;

begin
  with CGIEnvData1 do begin
    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
```

```
createStdout;  
sendPrologue;
```

{Find Session to edit}

```
{ Get field from search page }  
SessionNumber:= cgiQueryString^;
```

```
with Sessions do begin
```

```
{Prepare the table}  
open;  
first;
```

```
{find the correct Session}  
while SessionNumber <> fieldbyname('Number').asString do  
    next;
```

```
Session:= fieldbyname('Neumonic').asString;
```

```
end; {with Sessions}
```

{Send Session edit FORM to HTML page}

```
{Standard Header}  
send('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send('</HEAD><BODY bgcolor=FFFFFF>');  
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

```
send('Here is the Session you wanted to edit. Make changes and "Submit" them.<BR>');
```

```
send('<P><FORM ACTION=" ../cgi-win/Sessedt2.exe" METHOD="POST">');  
send('<INPUT TYPE="hidden" Name="flag" Value="1">');  
send('<INPUT TYPE="text" NAME="Session" SIZE="50" VALUE="" +Session+ "">');  
send('<INPUT TYPE="hidden" NAME="Sessionnumber" VALUE="" +SessionNumber+ "">');  
send('<P><CENTER><INPUT TYPE="submit" Value="Submit Change to Session">');  
send('<INPUT TYPE="reset" VALUE="Reset to Original Session"></CENTER></FORM>');
```

```
send('<P><FORM ACTION=" ../cgi-win/Sessions.exe" METHOD="POST">');  
send('<INPUT TYPE="hidden" Name="flag" Value="1">');  
send('<P><CENTER><INPUT TYPE="submit" Value="CANCEL - Do NOT  
change"></CENTER></FORM>');
```

```

{Standard Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {with cgiEnvData1 do}

end; {FormCreate}

end.

```

Edit Confirmation: **SesEdit2.exe**

```
unit Ses_edt2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Sessions: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    SessionsNumber: TIntegerField;
    SessionsNeumonic: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  dsession: string;
  dsessionnumber: string;

  dflag: string;           {valid program call}

begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
```

```
application.processMessages;
```

```
createStdout;  
sendPrologue;
```

```
{Standard Header}  
send ('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

```
{Determine whether flag is valid}
```

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin  {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

```
{ Get field from search page }  
dsession:= getsmallfield ('session');  
dsessionnumber:= getsmallfield ('sessionnumber');
```

```
{Put changed session into SESSION Table}
```

```
with Sessions do begin
```

```
  { Move to proper record }  
  open;  
  first;  
  while fieldbyName('Number').asstring <> dsessionnumber do  
    next;
```

```
  {Update record }  
  edit;  
  SetFields([nil, dsession]);  
  post;  
  close;  
end;  {with Sessions}
```

```
{Send the Confirmation page }
```

```
send ('<P>The Session has been changed to <B>' + dsession + '</B>');
```

```

send ('</CENTER><BR>');

send('<P><FORM ACTION=" ../cgi-win/Sessions.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Edit more
Sessions"></CENTER></FORM>');

{Standard Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}

else begin {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');

end; {flag **incorrect**}

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}
end; {FormCreate}

end.

```


Add a Session: SessAdd.exe

```
unit Sess_add;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
    sendPrologue;

    {Send Keyword ADD Form to HTML page}

    {Standard Header}
    send ('<HTML><HEAD>');
```

```

SendTitle('Asilomar Signals, Systems, and Computers');
send('</HEAD><BODY bgcolor=FFFFFF>');
send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

send('Fill in the Session Neumonics you would like to add.<BR>');
send('You may fill in all or none of these fields.<BR>');

send('<P><FORM ACTION=" ../cgi-win/SessAdd2.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><B>New Session: </B><INPUT TYPE="text" NAME="Session1" >');
send('<P><B>New Session: </B><INPUT TYPE="text" NAME="Session2" >');
send('<P><B>New Session: </B><INPUT TYPE="text" NAME="Session3" >');
send('<P><B>New Session: </B><INPUT TYPE="text" NAME="Session4" >');
send('<P><CENTER><INPUT TYPE="submit" Value="ADD these Sessions"></CENTER></FORM>');

send('<P><FORM ACTION=" ../cgi-win/Sessions.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="CANCEL - Do NOT Add any Sessions"></CENTER></FORM>');

{Standard Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin Options page"></CENTER></FORM>');
send('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {with cgiEnvData1 do}

end; {FormCreate}

end.

```

Addition Confirmation: SessAdd2.exe

```
unit Ses_add2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Sessions: TTable;
    SessionsNumber: TIntegerField;
    SessionsNeumonic: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  dsession1: string;
  dsession2: string;
  dsession3: string;
  dsession4: string;

  dbsession: string;

  dflag: string;           {valid program call}
```

```

    sessionPos: integer;

begin
    with CGIEnvData1 do begin

        webSiteINIFilename:=paramstr(1);
        application.onException:=cgiErrorHandler;
        application.processMessages;

        createStdout;
        sendPrologue;

        {Standard Header}
        send ('<HTML><HEAD>');
        SendTitle('Asilomar Signals, Systems, and Computers');
        send ('</HEAD><BODY bgcolor=FFFFFF>');

{Determine whether flag is valid}

        dflag:= getsmallfield ('flag');

        if (dflag = '1') then begin    {flag is set}

            send('<center><H2>Asilomar Conference System Administration</H2></center>');
            sendhr;

            { Get field from search page }
            dsession1:= getsmallfield ('session1');
            dsession2:= getsmallfield ('session2');
            dsession3:= getsmallfield ('session3');
            dsession4:= getsmallfield ('session4');

{ADD Sessions into Sessions Table}

            with Sessions do begin

                {Update record }
                open;

                if dsession1 <> " then begin

                    first;
                    sessionPos := CompareStr(dsession1, Fieldbyname ('neumonic').asstring);

```

while (not EOF) and (sessionPos < 0) do begin

 sessionPos := CompareStr(dsession1, Fieldbyname ('neumonic').asstring);
 end;

if sessionPos > 0 then begin

 insert;
 InsertRecord([nil, dsession1]);
 end

 else send ('The session neumonic ' + dsession1 + ' already exists');

end; {if session <> ''}

if dsession2 <> '' then begin

 first;
 sessionPos := CompareStr(dsession2, Fieldbyname ('neumonic').asstring);
 while (not EOF) and (sessionPos < 0) do begin

 sessionPos := CompareStr(dsession2, Fieldbyname ('neumonic').asstring);
 end;

if sessionPos > 0 then begin

 insert;
 InsertRecord([nil, dsession2]);
 end

 else send ('The session neumonic ' + dsession2 + ' already exists');

end; {if session <> ''}

close;

end; {with Sessions}

{Send the Thankyou page }

 sendhdr ('3', 'The following Sessions have been ADDED.
');
 send ('<P>' + dsession1 + '
');
 send ('" + dsession2 + '
');
 send ('" + dsession3 + '
');
 send ('" + dsession4 + '');

```

send ('</CENTER>');

send('<P><FORM ACTION=" ../cgi-win/Sessions.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Edit more
Sessions"></CENTER></FORM>');

{Standard Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}

else begin   {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');

end; {flag **incorrect**}

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}
end; {FormCreate}
end.

```

Delete Session(s): **SessDel.exe**

```
unit Sess_del;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Sessions: TTable;
    Query1: TQuery;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    SessionsNumber: TIntegerField;
    SessionsNeumonic: TStringField;
    Query1Number: TIntegerField;
    Query1Neumonic: TStringField;
    procedure FormCreate(Sender: TObject);
    procedure CGIDB1SendingHotField(currentRecord: TDataset;
      var s: OpenString);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  dflag: string;

begin
  with CGIEnvData1 do begin
```

{Standard Header}

```
webSiteINIFilename:=paramstr(1);
application.onException:=cgiErrorHandler;
application.processMessages;
createStdout;
sendPrologue;
send('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send('</HEAD><BODY bgcolor=FFFFFF>');
```

{ Get fields from search page }

```
dflag:= getsmallfield ('flag');
```

```
if dflag = '1' then begin
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');
```

```
sendhr;
```

```
send('<center>These are the current Sessions.<br>');
```

```
send('To DELETE a Session, simply select the Number next to the Session that you would  
like to DELETE.<BR>');
```

{Retrieve current Keywords}

```
with Query1 do begin
```

```
close;
```

```
sql.clear;
```

```
sql.add('Select*FROM Session ');
```

```
sql.add('Order By Number');
```

```
open;
```

{Check if any records were returned}

```
if recordcount > 0 then begin
```

{There are records returned}

```
Sessions.Open;
```

```
CGIDB1.DrawTable;
```

```
send('</center>');
```

```
Sessions.Close;
```

```
end {if recordcount > 0}
```

```
else
```

{No records returned}

```
sendhdr ('2','There are no Sessions in the Database to Delete!');
```

```
end; {with Query1}
```


{Delete all fields option}

```
send('<P><FORM ACTION="../cgi-win/SessDel2.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<INPUT TYPE="hidden" Name="DeleteAll" Value="Yes">');
send('<P><CENTER><INPUT TYPE="submit" Value="**** Delete ALL the Sessions
****"></CENTER></FORM>');
```

{HTML Footer}

```
sendhr;
send('<P><FORM ACTION="../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
```

end {flag is set}

else begin {flag **incorrect**}

```
send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF="../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');
```

```
send ('<P><hr><P><IMG SRC="../prototype/asil2b0.gif" align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
```

end; {flag **incorrect**}

```
send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}
```

end; {FormCreate}

```

procedure TForm1.CGIDB1SendingHotField(currentRecord: TDataset;
  var s: OpenString);
begin
  {Hotlink for single keyword edit}
  s:='<A HREF=" ../cgi-win/SessDel2.exe?' + currentrecord.fieldbyName('Number').AsString +
  ">'
  + currentrecord.FieldbyName('Number').AsString + '</A>';

end;

end.

```

Verification of Delete: SessDel2.exe

```
unit Ses_del2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Sessions: TTable;
    SessionsNumber: TIntegerField;
    SessionsNeumonic: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  SessionNumber, SessionName, dDeleteAll: string;

begin
  with CGIEnvData1 do begin
    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
```

```
createStdout;  
sendPrologue;
```

```
{Standard Header}  
send('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send('</HEAD><BODY bgcolor=FFFFFF>');  
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

{Find Session to Delete}

```
{ Get field from search page }  
try  
  dDeleteAll:= getsmallfield ('DeleteAll');  
finally  
  {Check to see if 'Delete all fields' is true}  
  SessionNumber:= cgiQueryString^;  
  end;
```

```
if getsmallfield ('DeleteAll') <> cginotfound then begin
```

```
  {Check to see if 'Delete all fields' is true}  
  dDeleteAll:= getsmallfield ('DeleteAll');
```

```
  if dDeleteAll = 'Yes' then begin
```

```
    with Sessions do begin
```

```
      sendhdr ('2', 'Are you sure you want to Delete ALL of the Sessions in the Database!');
```

```
      send('<P><FORM ACTION="../../cgi-win/Sessions.exe" METHOD="POST">');  
      send('<INPUT TYPE="hidden" Name="flag" Value="1">');  
      send('<P><CENTER><INPUT TYPE="submit" Value="CANCEL - Do NOT  
Delete"></CENTER></FORM>');  
      send('<P><FORM ACTION="../../cgi-win/SessDel3.exe" METHOD="POST">');  
      send('<INPUT TYPE="hidden" Name="flag" Value="1">');  
      send('<INPUT TYPE="hidden" NAME="sessionnumber" VALUE="delete">');  
      send('<INPUT TYPE="hidden" NAME="session" VALUE="delete">');  
      send('<P><CENTER><INPUT TYPE="submit" Value="*** YES - DELETE ALL  
Sessions in the Database! ***"></CENTER></FORM>');  
      end; {with Sessions}
```

```

    end; {if dDeleteAll}

end {if cginotfound}

else begin {Delete one record}

    with Sessions do begin

        {Prepare the table}
        open;
        first;

        {find the correct keyword}
        while SessionNumber <> fieldbyname('Number').asstring do
            next;

        {Get the Session Name}
        SessionName:= fieldbyname('Neumonic').asstring;
        end; {with Sessions}

{Send Session edit FORM to HTML page}

        send('Are you sure you want to DELETE the Session <B>' +SessionName+ ' </B><BR>');

        send('<P><FORM ACTION="../../cgi-win/SessDel3.exe" METHOD="POST">');
        send('<INPUT TYPE="hidden" Name="flag" Value="1">');
        send('<INPUT TYPE="hidden" NAME="sessionnumber" VALUE="" +SessionNumber+ "">');
        send('<INPUT TYPE="hidden" NAME="session" VALUE="" +SessionName+ "">');
        send('<P><CENTER><INPUT TYPE="submit" Value="Yes, DELETE this
Session"></CENTER></FORM>');

        send('<P><FORM ACTION="../../cgi-win/Sessions.exe" METHOD="POST">');
        send('<INPUT TYPE="hidden" Name="flag" Value="1">');
        send('<P><CENTER><INPUT TYPE="submit" Value="CANCEL - Do NOT
Delete"></CENTER></FORM>');
        end; {Delete one record}

{Standard Footer}
        sendhr;
        send('<P><FORM ACTION="../../cgi-win/Adminpwd.exe" METHOD="POST">');
        send('<INPUT TYPE="hidden" Name="flag" Value="1">');
        send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
        send ('</BODY></HTML>');
        closeStdout;

```

```
    closeApp( application );  
end; {with cgiEnvData1 do}  
end; {FormCreate}  
end.
```

Deletion Confirmation: SessDel3.exe

```
unit Ses_del3;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Sessions: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    SessionsNumber: TIntegerField;
    SessionsNeumonic: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  dSessionNumber, dSession: string;

  dflag: string;           {valid program call}

begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
```

```

application.processMessages;

createStdout;
sendPrologue;

{Standard Header}
send('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send('</HEAD><BODY bgcolor=FFFFFF>');

{Determine whether flag is valid}

dflag:= getsmallfield ('flag');

if (dflag = '1') then begin  {flag is set}

send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

    { Get field from search page }
    dSession:= getsmallfield ('Session');
    dSessionnumber:= getsmallfield ('Sessionnumber');

    if dSession = 'delete' then begin

        {Delete ALL the Sessions}
        with Sessions do begin

            open;
            first;
            while not EOF do begin
                delete;
            end; {while not EOF}

            end; {with Sessions}

            sendhdr ('2', 'ALL Sessions were deleted!');
            end {if Sdession = delete}

        else begin {Delete one Session}

{Delete Session from SESSION Table}

```


with Sessions do begin

```
{ Move to proper record}
open;
first;
while fieldbyName('Number').asstring < dSessionnumber do
    next;

{Delete record }
edit;
delete;
close;
end; {with Sessions}
```

{Send the Thankyou page for delete one record }

```
send ('<P>The Session <B> ' + dSession + '</B> has been DELETED.</CENTER><BR>');

send('<P><FORM ACTION=" ../cgi-win/SessDel.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="DELETE more
Sessions"></CENTER></FORM>');
```

end; {else Delete one Session}

```
send('<P><FORM ACTION=" ../cgi-win/Sessions.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Edit more
Sessions"></CENTER></FORM>');
```

{Standard Footer}

```
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}
```

else begin {flag **incorrect**}

```
send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
```

```

send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF="../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

```

```

send ('<P><hr><P><IMG SRC="../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');

```

```

end; {flag **incorrect**}

```

```

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}
end; {FormCreate}
end. {Application}

```

Choose a Keyword: **Keywords.exe**

```
unit Key_word;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Query1: TQuery;
    Table1: TTable;
    Query1Name: TStringField;
    Table1Number: TIntegerField;
    Table1Name: TStringField;
    Query1Number: TIntegerField;
    procedure FormCreate(Sender: TObject);
    procedure CGIDB1OnSendingHotField(currentRecord: TDataset;
      var s: OpenString);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  dflag: string;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    {Standard Header}
    webSiteINIFilename:=paramstr(1);
```

```
application.onException:=cgiErrorHandler;  
application.processMessages;
```

```
createStdout;  
sendPrologue;  
send('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send('</HEAD><BODY bgcolor=FFFFFF>');
```

{Determine whether flag is valid}

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin  {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

{Send intructions}

```
send('<center>These are the current Keywords.<br>');  
send('To edit a single Keyword, simply click on the Keyword that you would like to  
change.<BR>');  
send(' Other options are available at the bottom of the page.<BR>');
```

{Retrieve current Keywords}

```
with Query1 do begin
```

```
close;  
sql.clear;  
sql.add('Select*FROM Keyword ');  
sql.add('Order By Number');  
open;
```

```
if recordcount > 0 then begin
```

```
Table1.Open;  
CGIDB1.DrawTable;  
send('</center>');  
Table1.Close;  
end {if recordcount > 0}
```

```
else  
sendhdr ('2','There are no Keywords in the Database!');
```

```
end;  {with Query1}
```

{Button to Add keywords}

```

send('<P><FORM ACTION=" ../cgi-win/KeyAdd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Add a New
Keyword"></CENTER></FORM>');

```

{Button to Delete keywords}

```

send('<P><FORM ACTION=" ../cgi-win/KeyDel.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Delete
Keyword(s)"></CENTER></FORM>');

```

{HTML Footer}

```

sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}

```

```

else begin {flag **incorrect**}

```

```

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

```

```

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

```

```

end; {flag **incorrect**}

```

```

send ('</BODY></HTML>');
closeStdout;
closeApp( application );

```

```

end; {with CGIEnvData1}

end; {FormCreate}

procedure TForm1.CGIDB1OnSendingHotField(currentRecord: TDataset;
  var s: OpenString);
begin
  {Hotlink for single keyword edit}
  s:='<A HREF=" ../cgi-win/KeyEdit.exe?' + currentrecord.fieldbyName('Number').AsString +
  ">'
  + currentrecord.FieldbyName('Name').AsString + '</A>';

end;

end.

```

Edit a Keyword: **KeyEdit.exe**

```
unit Key_edit;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DBTables, DB;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Keywords: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    KeywordsNumber: TIntegerField;
    KeywordsName: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  KeyNumber, KeyName: string;

begin
  with CGIEnvData1 do begin
    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
```

```
createStdout;  
sendPrologue;
```

{Find Keyword to edit}

```
{ Get field from search page }  
KeyNumber:= cgiQueryString^;
```

```
with Keywords do begin
```

```
  {Prepare the table}  
  open;  
  first;
```

```
  {Move to the correct keyword}  
  while KeyNumber <> fieldbyname('Number').asstring do  
    next;
```

```
  KeyName:= fieldbyname('Name').asstring;
```

```
end; {with Keywords}
```

{Send Keyword edit FORM to HTML page}

```
{Standard Header}  
send ('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send ('</HEAD><BODY bgcolor=FFFFFF>');  
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

```
send('Here is the Keyword you wanted to edit. Make changes and "Submit" them.<BR>');
```

```
send('<P><FORM ACTION="../cgi-win/keyedit2.exe" METHOD="POST">');  
send('<INPUT TYPE="hidden" Name="flag" Value="1">');  
send('<INPUT TYPE="text" NAME="keyword" SIZE="50" VALUE="" +KeyName+ ">');  
send('<INPUT TYPE="hidden" NAME="keynumber" VALUE="" +KeyNumber+ ">');  
send('<P><CENTER><INPUT TYPE="submit" Value="Submit Change to Keyword">');  
send('<INPUT TYPE="reset" VALUE="Reset to Original  
Keyword"></CENTER></FORM>');
```

```
send('<P><FORM ACTION="../cgi-win/Keywords.exe" METHOD="POST">');  
send('<INPUT TYPE="hidden" Name="flag" Value="1">');  
send('<P><CENTER><INPUT TYPE="submit" Value="CANCEL - Do NOT  
change"></CENTER></FORM>');
```



```

{Standard Footer}
sendhr;
send('<P><FORM ACTION="..../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {with cgiEnvData1 do}

end; {FormCreate}

end. {Application}

```

Edit Confirmation: **KeyEdit2.exe**

```
unit Key_edt2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    keywords: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  dkeyword: string;
  dkeynumber: string;

  dflag: string;           {valid program call}

begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
```

```
createStdout;  
sendPrologue;
```

```
{Standard Header}  
send ('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

```
{Determine whether flag is valid}
```

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

```
{ Get field from search page }  
dkeyword:= getsmallfield ('keyword');  
dkeynumber:= getsmallfield ('keynumber');
```

```
{Put changed Keyword into KEYWORD Table}
```

```
with Keywords do begin
```

```
  { Move to proper record }  
  open;  
  first;  
  while fieldbyName('Number').asstring <> dkeynumber do  
    next;
```

```
  {Update record }  
  edit;  
  SetFields([nil, dkeyword]);  
  post;  
  close;  
end; {with Keywords}
```

```
{Send the Thankyou page }
```

```
send ('<P>The Keyword has been changed to <B> ' + dkeyword + '</B>.</CENTER><BR>');
```

```
send('<P><FORM ACTION=" ../cgi-win/Keywords.exe" METHOD="POST">');
```

```

    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Edit more
Keywords"></CENTER></FORM>');

```

{Standard Footer}

```

sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}

```

```

else begin    {flag **incorrect**}

```

```

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

```

```

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

```

```

    end; {flag **incorrect**}

```

```

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}
end; {FormCreate}
end. {Application}

```

Add a Keyword: **KeyAdd.exe**

```
unit Key_add;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Query1: TQuery;
    Table1: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  dflag: string;           {valid program call}

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
    sendPrologue;
```

{Standard Header}

```
send('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send('</HEAD><BODY bgcolor=FFFFFF>');
```

{Determine whether flag is valid}

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin  {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;
```

{Send Keyword ADD Form to HTML page}

```
send('Fill in the Keywords you would like to add.<BR>');
send('You may fill in all or none of these fields.<BR>');
```

```
send('<P><FORM ACTION=" ../cgi-win/KeyAdd2.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><B>New Keyword: </B><INPUT TYPE="text" NAME="keyword1" >');
send('<P><B>New Keyword: </B><INPUT TYPE="text" NAME="keyword2" >');
send('<P><B>New Keyword: </B><INPUT TYPE="text" NAME="keyword3" >');
send('<P><B>New Keyword: </B><INPUT TYPE="text" NAME="keyword4" >');
send('<P><CENTER><INPUT TYPE="submit" Value="ADD these
Keywords"></CENTER></FORM>');

send('<P><FORM ACTION=" ../cgi-win/Keywords.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="CANCEL - Do NOT Add any
Keywords"></CENTER></FORM>');
```

{Standard Footer}

```
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
```

```
end {flag is set}
```

```
else begin  {flag **incorrect**}
```

```
send('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
```

```

sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

```

```

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

```

```

    end; {flag **incorrect**}

```

```

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}

```

```

end; {FormCreate}

```

```

end. {Application}

```

Addition Confirmation: **KeyAdd2.exe**

```
unit Key_add2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Table1: TTable;
    Table1Number: TIntegerField;
    Table1Name: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  dflag: string;           {valid program call}

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  dkeyword1: string;
  dkeyword2: string;
  dkeyword3: string;
  dkeyword4: string;

begin
  with CGIEnvData1 do begin
```



```
webSiteINIFilename:=paramstr(1);  
application.onException:=cgiErrorHandler;  
application.processMessages;
```

```
createStdout;  
sendPrologue;
```

```
{Standard Header}  
send ('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

{Determine whether flag is valid}

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

```
{ Get field from search page }  
dkeyword1:= getsmallfield ('keyword1');  
dkeyword2:= getsmallfield ('keyword2');  
dkeyword3:= getsmallfield ('keyword3');  
dkeyword4:= getsmallfield ('keyword4');
```

{ADD Keywords into KEYWORD Table}

```
with Table1 do begin
```

```
{Update record }  
open;
```

```
if dkeyword1 <> " then  
AppendRecord([nil, dkeyword1]);
```

```
if dkeyword2 <> " then  
AppendRecord([nil, dkeyword2]);
```

```
if dkeyword3 <> " then  
AppendRecord([nil, dkeyword3]);
```

```

if dkeyword4 <> " then
AppendRecord([nil, dkeyword4]);

```

```

close;
end; {with table1}

```

{Send the Thankyou page }

```

sendhdr ('3', 'The following Keywords have been ADDED.<BR>');
send ('<P><B>' + dkeyword1 + '<BR>');
send (" + dkeyword2 + '<BR>');
send (" + dkeyword3 + '<BR>');
send (" + dkeyword4 + '</B>');

send ('</CENTER>');

send('<P><FORM ACTION=" ../cgi-win/Keywords.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Edit more
Keywords"></CENTER></FORM>');

```

{Standard Footer}

```

sendhdr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}

```

```

else begin {flag **incorrect**}

```

```

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhdr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

```

```

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');

```

```
send ('Conference On-line Submission Page</A>');  
send ('</BODY></HTML>');  
closeStdout;  
closeApp( application );
```

```
    end; {flag **incorrect**}
```

```
    send ('</BODY></HTML>');  
    closeStdout;  
    closeApp( application );  
    end; {with CGIEnvData1}
```

```
end; {FormCreate}
```

```
end. {Application}
```

Delete Keyword(s): **KeyDel.exe**

```
unit Key_del;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DBTables, DB;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Query1: TQuery;
    Table1: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    Table1Number: TIntegerField;
    Table1Name: TStringField;
    Query1Number: TIntegerField;
    Query1Name: TStringField;
    procedure FormCreate(Sender: TObject);
    procedure CGIDB1OnSendingHotField(currentRecord: TDataset;
      var s: OpenString);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  dflag: string;

begin
  with CGIEnvData1 do begin
```

{Standard Header}

```
webSiteINIFilename:=paramstr(1);  
application.onException:=cgiErrorHandler;  
application.processMessages;
```

```
createStdout;  
sendPrologue;
```

```
send ('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

{Determine whether flag is valid}

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin  {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

{Send Instructions}

```
send('<center>These are the current Keywords.<br>');  
send('To DELETE a Keyword, simply select the Number next to the Keyword that you would  
like to DELETE.<BR>');
```

{Retrieve current Keywords}

```
with Query1 do begin
```

```
close;  
sql.clear;  
sql.add('Select*FROM Keyword ');  
sql.add('Order By Number');  
open;
```

```
if recordcount > 0 then begin
```

```
Table1.Open;  
CGIDB1.DrawTable;  
send('</center>');  
Table1.Close;  
end {if recordcount > 0}
```

```
else
```

```
sendhdr ('2','There are no Keywords in the Database to Delete!');
```

```

end; {with Query1}

{Delete all fields option}
send('<P><FORM ACTION="../cgi-win/KeyDel2.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<INPUT TYPE="hidden" Name="DeleteAll" Value="Yes">');
send('<P><CENTER><INPUT TYPE="submit" Value="**** Delete ALL the Keywords
****"></CENTER></FORM>');

{HTML Footer}
sendhr;
send('<P><FORM ACTION="../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}

else begin {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF="../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC="../prototype/asil2b0.gif" align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {flag **incorrect**}

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}

```

end; {FormCreate}

procedure TForm1.CGI_DB1_OnSendingHotField(currentRecord: TDataset;
var s: OpenString);
begin

 {Hotlink for single keyword edit}

 s:='<A HREF=" ../cgi-win/KeyDel2.exe?'+ currentrecord.fieldbyName('Number').AsString +
 ">'
 + currentrecord.FieldbyName('Number').AsString + '';

end;

end.

Verification of Delete: **KeyDel2.exe**

```
unit Key_del2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Table1: TTable;
    Table1Number: TIntegerField;
    Table1Name: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
var
  KeyNumber,           {Number of the Keyword record}
  KeyName,             {Name of the Keyword}
  dDeleteAll,          {action flag - if set then delete all the Keywords}
  dflag: string;       {valid program call}

begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
```



```
application.processMessages;
```

```
createStdout;  
sendPrologue;
```

```
{Standard Header}  
send ('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

```
{Determine whether flag is valid}
```

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

```
{Find Keyword to Delete}
```

```
{ Get field from search page }  
try  
dDeleteAll:= getsmallfield ('DeleteAll');  
finally  
{Check to see if 'Delete all fields' is true}  
KeyNumber:= cgiQueryString^;  
end;
```

```
if getsmallfield ('DeleteAll') <> cginotfound then begin
```

```
{Check to see if 'Delete all fields' is true}  
dDeleteAll:= getsmallfield ('DeleteAll');
```

```
if dDeleteAll = 'Yes' then begin
```

```
with Table1 do begin
```

```
sendhdr ('2', 'Are you sure you want to Delete ALL of the Keywords in the Database!');
```

```
send('<P><FORM ACTION=" ../cgi-win/Keywords.exe" METHOD="POST">');  
send('<INPUT TYPE="hidden" Name="flag" Value="1">');  
send('<P><CENTER><INPUT TYPE="submit" Value="CANCEL - Do NOT  
Delete"></CENTER></FORM>');
```

```

send('<P><FORM ACTION=" ../cgi-win/KeyDel3.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<INPUT TYPE="hidden" NAME="keynumber" VALUE="delete">');
send('<INPUT TYPE="hidden" NAME="keyname" VALUE="delete">');
send('<P><CENTER><INPUT TYPE="submit" Value="*** YES - DELETE ALL
Keywords in the Database! ***"></CENTER></FORM>');
end; {with Table1}

```

```

end; {if dDeleteAll}

```

```

end {if cginotfound}

```

```

else begin {Delete one record}

```

```

    with Table1 do begin

```

```

        {Prepare the table}

```

```

        open;
        first;

```

```

        {find the correct keyword}

```

```

        if KeyNumber <> fieldbyname('Number').asString then begin

```

```

            {move to the next field until we find the correct one}
            repeat

```

```

                next;

```

```

            until KeyNumber = fieldbyname('Number').asString;
            end; {if keyword}

```

```

        {Get the Keyword Name}

```

```

        KeyName:= fieldbyname('Name').asString;
        end; {with Table1}

```

```

{Send Keyword edit FORM to HTML page}

```

```

send('Are you sure you want to DELETE the keyword <B>' +KeyName+ '!.</B><BR>');

```

```

send('<P><FORM ACTION=" ../cgi-win/KeyDel3.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<INPUT TYPE="hidden" NAME="keynumber" VALUE="" +KeyNumber+ "">');
send('<INPUT TYPE="hidden" NAME="keyname" VALUE="" +KeyName+ "">');
send('<P><CENTER><INPUT TYPE="submit" Value="Yes, DELETE this
Keyword"></CENTER></FORM>');

```

```

send('<P><FORM ACTION=" ../cgi-win/Keywords.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="CANCEL - Do NOT
Delete"></CENTER></FORM>');
end; {Delete one record}

{Standard Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}

else begin  {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {flag **incorrect**}

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}

end; {Formcreate}

end. {Application}

```

Deletion Confirmation: **KeyDel3.exe**

```
unit Key_del3;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Keywords: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    KeywordsNumber: TIntegerField;
    KeywordsName: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
var
  dKeyNumber, dKeyword: string;
  dflag: string;          {valid program call}

begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
```

```
createStdout;  
sendPrologue;
```

```
{Standard Header}  
send ('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

```
{Determine whether flag is valid}
```

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');  
sendhr;
```

```
{ Get field from search page }  
dkeyword:= getsmallfield ('keyname');  
dkeynumber:= getsmallfield ('keynumber');
```

```
if dkeyword = 'delete' then begin
```

```
with Keywords do begin
```

```
open;  
first;  
while not EOF do  
delete;
```

```
end; {with Keywords}
```

```
sendhdr ('2', 'ALL Keywords were deleted!');  
end {if dkeyword = delete}
```

```
else begin {Delete one keyword}
```

```
{Delete Keyword from KEYWORD Table}
```

```
with Keywords do begin
```

```
{ Move to proper record}
```

```

open;
first;
while fieldbyName('Number').asstring <> dkeynumber do
    next;

```

```

{Delete record }
edit;
delete;
close;
end; {with Keywords}

```

{Send the Thankyou page }

```

send ('<P>The Keyword <B> ' + dkeyword + '</B> has been DELETED.</CENTER><BR>');
send('<P><FORM ACTION=" ../cgi-win/KeyDel.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="DELETE more
Keywords"></CENTER></FORM>');

```

end; {else Delete one keyword}

```

send('<P><FORM ACTION=" ../cgi-win/Keywords.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Edit more
Keywords"></CENTER></FORM>');

```

```

{Standard Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}

```

else begin {flag **incorrect**}

```

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');

```

```
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC="../../prototype/asil2b0.gif" align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');

    end; {flag **incorrect**}

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}

end; {Formcreate}

end. {Application}
```

Display Titles: **Invite.exe**

```
unit Invite_;
```

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type

TForm1 = class(TForm)
 DataSource1: TDataSource;
 Articles: TTable;
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 procedure FormCreate(Sender: TObject);
private
 { Private declarations }
public
 { Public declarations }
end;

var

Form1: TForm1;
dbtitle, dbPaper_Number: string;

dflag: string; {valid program call}

implementation

{\$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin

 with CGIEnvData1 do begin

 webSiteINIFilename:=paramstr(1);
 application.onException:=cgiErrorHandler;
 application.processMessages;
 createStdout;
 sendPrologue;

 send ('<HTML><HEAD>');


```
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
```

```
{Determine whether flag is valid}
```

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin  {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;
```

```
{Send Instructions}
```

```
send('<P>Articles must be marked as Invited so they will not be included in the Review
process. ');
```

```
send('<P><UL><LI>To mark Articles as Invited, select the checkbox next to the
Title.<BR>');
```

```
send('<LI>To change Articles to NOT Invited, change the action in the drop-down box to
"NOT Invited" and then ');
```

```
send('select the checkbox next to the Title.</UL>');
```

```
send('<P>Note: it is not necessary to mark an article NOT Invited unless it was previously
marked Invited ');
```

```
send('and it needs to be changed. When printing Article information for review, only Articles
marked Invited are ');
```

```
send('ignored. Articles that are not marked <B>or</B> are marked NOT Invited are printed
for review.');
```

```
sendhr;
```

```
{Build Form}
```

```
send('<FORM ACTION=" ../cgi-win/Invite2.exe" METHOD="GET">');
```

```
{Build Pull-Down menu for choices}
```

```
send('<P><center><SELECT NAME="Action">');
```

```
send('<OPTION> Invited');
```

```
send('<OPTION> NOT Invited');
```

```
send('</SELECT></center><P>');
```

```
{Build the check boxes of Titles}
```

```
with Articles do begin
```

```
open;
```

```
first;
```

```
While not Articles.EOF do begin
```

```
dbtitle:= fieldbyname('Title').asstring;
```

```
dbPaper_Number:= fieldbyname('Paper_Number').asstring;
```

```

        send ('<INPUT TYPE="checkbox" NAME="T" Value="" +dbPaper_Number+ "># '
+dbPaper_Number+ ' ' +dbtitle+ '<BR>');
        next;
        end; {while not EOF}
    close;
    end; {with articles do}

    send('<P><CENTER><INPUT TYPE="submit" Value=" Submit these articles "><P> ');
    send('<INPUT TYPE="reset" VALUE="Reset values"></CENTER></FORM>');

    sendhr;
    send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    end {flag is set}

else begin    {flag **incorrect**}

    send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
    sendhr;
    send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
    send ('<P>Please ensure that you are authorized to access this information. ');
    send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
    send ('<P>If you are not authorized to access this information, please note that a log is');
    send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
    send ('prosecution for your actions.<BR>');

    send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
    send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
    send ('Conference On-line Submission Page</A>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

        end; {flag **incorrect**}

    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );
    end; {with CGIEnvData1}

```

end; {FormCreate}

end.

Invite Titles: Invite2.exe

```
unit Invite_2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Articles: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  i, j, dTitle, MySelector, dbPaper_Number: integer;

  dAction : string;
  dTitlein: string;
  dString, AssignString: string;

  Titlelist : TStringList;

  function TitleToInt ( S: string) :integer;

implementation

{$R *.DFM}
{*****}
function TitleToInt ( S: string) :integer;

{This function is necessary because the string value of dTitle has several spaces in
it and the strtoint function cannot handle the conversion properly}
```

```

var
  temp: string;  {temporary variable to build new string into}
  j: integer;    {Counting Variable to keep track of how many characters are in the
number}

begin
  {set the counter to zero}
  j:= 0;

  {count the characters}
  repeat
    j:= j+1;
  until S[j+1] = '';

  {Copy the valid number characters over to temp}
  temp := Copy(S, 1, j);

  {Convert the temp string to an integer value}
  TitleToInt := strtoint (temp);

end; {NoSpaces}
{*****}

procedure TForm1.formcreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
    createStdout;
    sendPrologue;

    send ('<HTML><HEAD>');
    SendTitle('Asilomar Signals, Systems, and Computers');
    send ('</HEAD><BODY bgcolor=FFFFFF>');
    send('<center><H2>Asilomar Conference System Administration</H2></center>');
    sendhr;

    {Get query string from Assign.exe}
    Titlelist:= TStringlist.create;  {list of titles to assign to the session}

    AssignString:= CGIQueryString^;

```

j:= 0; {The counter for the individual strings within the QueryString}

for i := 7 to Length(AssignString) do {The counter for going through the AssignString - eight is the first valuable character (the beginning of the Session Neumonic) }

```

if AssignString[i] < '&' then begin    {the character is valuable}
    dString[j] := AssignString[i];
    j:= j+1;
end {if AssignString[i] < '&' then}
else begin    {The & signifies the end of the string}
    TitleList.add (dString);
    dString:= ' ';    {Blank out the old values}
    j:=0;    {reset the string counter}
    i:= i+1;    {Skip over the text "T="}
end; {else}

```

TitleList.add (dString); {This is to put the last string into the string list. It is necessary because there is no & at the end}

{Place the Session Neumonic into each Paper Number passed}

```
{The first item in the Title List is the Session Neumonic}
dAction:= TitleList[0];
```

```
if (dAction[1] = 'T') then MySelector := 1;  
if (dAction[1] = 'N') then MySelector := 2;
```

case MySelector of

1 : begin **{Invite only}**

{Send confirmation}
sendhdr ('3', 'P>The following Articles were INVITED:
<center>');

**{Invite the articles with the selected Paper_Numbers}
with Articles do begin**

```
open; {Prepare the Table for use}
for i:= 1 to TitleList.count-1 do begin {The rest of the items are Paper Numbers}
```

```
{Get the next title number}  
dTitleIn:= TitleList[i];
```

{Convert the Title string to an integer value}

```

dTitle:= TitleToInt (dTitlein);

{Move to proper record}
first; {go back to the first record before every new search}
while fieldbyname('Paper_Number').asinteger <> dTitle do
    next;

{Invite the Article}
edit;
Setfields ([nil, nil, 'Y']); {The remaining fields are ignored}
post;

{Send title name}
send('# ' + fieldbyname('Paper_Number').asstring + ' ' + fieldbyname('Title').asstring +
'<BR>');

end; {for}

close; { the table}

end; {with Articles}
send('</center>');
end; {Accept only}

2 : begin {NOT Invited}

{Send confirmation}
sendhdr ('3', '<P>The following Articles were marked as NOT
INVITED:<br><center>');

{Invite the articles with the selected Paper_Numbers}
with Articles do begin

    open; {Prepare the Table for use}

    for i:= 1 to TitleList.count-1 do begin {The rest of the items are Paper_Numbers}

        {Get the next title number}
        dTitlein:= TitleList[i];

        {Convert the Title string to an integer value}
        dTitle:= TitleToInt (dTitlein);

        {Move to proper record}

```

```

first; {go back to the first record before every new search}
while fieldbyname('Paper_Number').asinteger < dTitle do
    next;

    {Invite the Article}
    edit;
    Setfields ([nil, nil, 'N']); {The remaining fields are ignored}
    post;

    {Send title name}
    send('# ' + fieldbyname('Paper_Number').asstring + ' ' + fieldbyname('Title').asstring +
'<BR>');

    end; {for}

close; { the table}

end; {with Articles}

end; {Invite AND NOT Invite}

else begin {selector not working}

    send('Selector did not work<BR>');
    send('Action was ' + dAction + '.');
    end; {else}

end; {case}


{Send Footer}
send('<P><FORM ACTION="../../cgi-win/Invite.exe" METHOD="POST">');
send('<CENTER><INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><INPUT TYPE="submit" Value="Invite more Articles"></FORM><P> ');

sendhr;
send('<P><FORM ACTION="../../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send('</BODY></HTML>');
closeStdout;
closeApp( application );

```


end; {with cgiEnvData1 do}

end; {FormCreate}

end.

Select Articles: Choose.exe

```
unit Choose_;  
  
interface  
  
uses  
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;  
  
type  
  TForm1 = class(TForm)  
    CGIEnvData1: TCGIEnvData;  
    CGIDB1: TCGIDB;  
    DataSource1: TDataSource;  
    Query1: TQuery;  
    Articles: TTable;  
    procedure FormCreate(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
  
var  
  Form1: TForm1;  
  dbtitle, dbPaper_Number: string;  
  
  dflag: string;           {valid program call}  
  
implementation  
  
{$R *.DFM}  
  
procedure TForm1.FormCreate(Sender: TObject);  
begin  
  
  with CGIEnvData1 do begin  
  
    webSiteINIFilename:=paramstr(1);  
    application.onException:=cgiErrorHandler;  
    application.processMessages;  
    createStdout;  
    sendPrologue;
```

```

send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');

```

{Determine whether flag is valid}

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin  {flag is set}
```

```

send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

```

{Build Form}

```
send('<P><FORM ACTION=" ../cgi-win/Choose2.exe" METHOD="GET">');
```

{build Pull-Down menu for choices}

```

send('There are four methods of accepting or rejecting Articles:<BR><ul>');
send('<LI>Accept the selected articles only, do not modify any other Articles');
send('<LI>Accept the selected articles AND Reject all others');
send('<LI>Reject the selected articles only, do not modify any other Articles');
send('<LI>Reject the selected articles AND Accept all others</ul>');

```

```

send('<P><center><SELECT NAME="Action">');
send('<OPTION> ACCEPT');
send('<OPTION> ACCEPT_then_Reject');
send('<OPTION> REJECT');
send('<OPTION> REJECT_then_Accept');
send('</SELECT></center><P>');

```

{Build the check boxes of Titles}

```
with Articles do begin
```

```
open;
```

```
first;
```

```
While not Articles.EOF do begin
```

```
dbtitle:= fieldbyname('Title').asstring;
```

```
dbPaper_Number:= fieldbyname('Paper_Number').asstring;
```

```
send ('<INPUT TYPE="checkbox" NAME="T" Value="" +dbPaper_Number+ ""> '
```

```
+dbtitle+ ' <BR>');
```

```
next;
```

```
end; {while not EOF}
```

```
close;
```

```
end; {with articles do}
```

```

send('<P><CENTER><INPUT TYPE="submit" Value=" Accept / Reject these articles "><P>
');
send('<INPUT TYPE="reset" VALUE="Reset values"></CENTER></FORM>');

sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end b}

else begin    {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');

    end; {flag **incorrect**}

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}

end; {FormCreate}

end.

```

Accept / Reject Articles: Choose2.exe

```
unit Choose_2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Articles: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  i, j, dTitle, MySelector, dbPaper_Number: integer;

  dAction : string;
  dTitlein: string;
  dString, AssignString: string;

  Titlelist : TStringList;

  function TitleToInt ( S: string) :integer;

implementation

{$R *.DFM}
{*****}
function TitleToInt ( S: string) :integer;

{This function is necessary because the string value of dTitle has several spaces in
it and the strtoint function cannot handle the conversion properly}
```

```

var
  temp: string;   {temporary variable to build new string into}
  j: integer;     {Counting Variable to keep track of how many characters are in the
number}

begin
  {set the counter to zero}
  j:= 0;

  {count the characters}
  repeat
    j:= j+1;
  until S[j+1] = '';

  {Copy the valid number characters over to temp}
  temp := Copy(S, 1, j);

  {Convert the temp string to an integer value}
  TitleToInt := strtoint (temp);

end; {NoSpaces}
{*****}

procedure TForm1.formcreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
    createStdout;
    sendPrologue;

    send ('<HTML><HEAD>');
    SendTitle('Asilomar Signals, Systems, and Computers');
    send ('</HEAD><BODY bgcolor=FFFFFF>');
    send('<center><H2>Asilomar Conference System Administration</H2></center>');
    sendhr;

    {Get query string from Assign.exe}
    Titlelist:= TStringlist.create;  {list of titles to assign to the session}

    AssignString:= CGIQueryString^;

```

j:= 0; **{The counter for the individual strings within the QueryString}**

for i := 7 to Length(AssignString) do **{The counter for going through the AssignString - eight is the first valuable character (the beginning of the Session Neumonic) }**

```
if AssignString[i] <> '&' then begin  {the character is valuable}
  dString[j] := AssignString[i];
  j:= j+1;
end {if AssignString[i] <> '&' then}
else begin {The & signifies the end of the string}
  TitleList.add (dString);
  dString:= '                '; {Blank out the old values}
  j:=0;           {reset the string counter}
  i:= i+1;       {Skip over the text "T="}
end; {else}
```

TitleList.add (dString); **{This is to put the last string into the string list. It is necessary because there is no & at the end}**

{Place the Session Neumonic into each Paper_Number passed}

{The first item in the Title List is the Session Neumonic}

dAction:= TitleList[0];

```
if (dAction[1]='A') and (dAction[7] <> '_') then MySelector := 1;
if (dAction[1]='A') and (dAction[7] = '_') then MySelector := 2;
if (dAction[1]='R') and (dAction[7] <> '_') then MySelector := 3;
if (dAction[1]='R') and (dAction[7] = '_') then MySelector := 4;
```

case MySelector of

1 : begin **{Accept only}**

{Send confirmation}

 sendhdr ('3', '<P>The following Articles were ACCEPTED:
<center>');

{accept the articles with the selected Paper_Numbers}

 with Articles do begin

 open; **{Prepare the Table for use}**

 for i:= 1 to TitleList.count-1 do begin **{The rest of the items are Paper_Numbers}**

{Get the next title number}

 dTtitlein:= TitleList[i];

```

    {Convert the Title string to an integer value}
    dTitle:= TitleToInt (dTitlein);

    {Move to proper record}
    first; {go back to the first record before every new search}
    while fieldbyname('Paper_Number').asinteger < dTitle do
        next;

    {Insert the Session}
    edit;
    Setfields ([nil, nil, nil, 'Y']); {The remaining fields are ignored}
    post;

    {Send title name}
    send(" +fieldbyname('Title').asstring+ '<BR>');

    end; {for}

close; { the table}

    end; {with Articles}
    send('</center>');
    end; {Accept only}

2 : begin {Accept and Reject}

    {Send confirmation}
    sendhdr ('3', '<P>The following Articles were ACCEPTED:<br><center>');

    {accept the articles with the selected Paper_Numbers}
    with Articles do begin

        open; {Prepare the Table for use}

        {Set all *Not Invited* Articles to "rejected"}
        while not EOF do begin

            if fieldbyname('Invited').asstring < 'Y' then begin

                edit;
                Setfields ([nil, nil, nil, 'N']); {The remaining fields are ignored}
                post;
                end; {If not invited}

            next;

```


end; {While}

for i:= 1 to TitleList.count-1 do begin {The rest of the items are Paper_Numbers}

{Get the next title number}

dTitlein:= TitleList[i];

{Convert the Title string to an integer value}

dTitle:= TitleToInt (dTitlein);

{Move to proper record}

first; {go back to the first record before every new search}

while fieldbyname('Paper_Number').asinteger < dTitle do

next;

{Insert the Session}

edit;

Setfields ([nil, nil, nil, 'Y']); {The remaining fields are ignored}

post;

{Send title name}

send(" +fieldbyname('Title').asstring+ '
');

end; {for}

close; { the table}

end; {with Articles}

sendhdr('3', '<P>All OTHER Articles were Rejected!</center>
<P>');

end; {Accept and Reject}

3 : begin {'Reject only}

{Send confirmation}

sendhdr ('3', '<P>The following Articles were REJECTED:
<center>');

{The rest of the items are Paper_Numbers}

with Articles do begin

open; {Prepare the Table for use}

for i:= 1 to TitleList.count-1 do begin

{Get the next title number}

```

dTitlein:= TitleList[i];

{Convert the Title string to an integer value}
dTitle:= TitleToInt (dTitlein);

{Move to proper record}
first; {go back to the first record before every new search}
while fieldbyname('Paper_Number').asinteger < dTitle do
    next;

{Insert the Session}
edit;
Setfields ([nil, nil, nil, 'N']); {The remaining fields are ignored}
post;

{Send title name}
send(" +fieldbyname('Title').asstring+ '<BR>');

end; {for}

close; { the table}
send('</center>');

end; {with Articles}

end; {Reject only}

4: begin {'Reject and accept'}

{Send confirmation}
sendhdr ('3', '<P>The following Articles were REJECTED:<br><center>');

{accept the articles with the selected Paper_Numbers}
with Articles do begin

    open; {Prepare the Table for use}

    {Set all *Not Invited* Articles to "accepted"}
    while not EOF do begin

        if fieldbyname ('Invited').asstring < 'Y' then begin

            edit;
            Setfields ([nil, nil, nil, 'Y']); {The remaining fields are ignored}
            post;

```

```

    end; {If not invited}

next;
end; {While}

for i:= 1 to TitleList.count-1 do begin {The rest of the items are Paper_Numbers}

    {Get the next title number}
    dTitlein:= TitleList[i];

    {Convert the Title string to an integer value}
    dTitle:= TitleToInt (dTitlein);

    {Move to proper record}
    first; {go back to the first record before every new search}
    while fieldbyname('Paper_Number').asinteger < dTitle do
        next;

    {Insert the Session}
    edit;
    Setfields ([nil, nil, nil, 'N']); {The remaining fields are ignored}
    post;

    {Send title name}
    send(" +fieldbyname('Title').asstring+ '<BR>');

    end; {for}

close; { the table}

end; {with Articles}

sendhdr('3', '<P>All OTHER Articles were Accepted!</center><BR><P>');

    end; {Reject and accept}
else begin {selector not working}

    send('Selector did not work<BR>');
    send('Action was ' +dAction+ '.');
    end; {else}

end; {case}

{Send Footer}

```

{Button to Accept/Reject more Articles}

```
send('<P><FORM ACTION=" ../cgi-win/Choose.exe" METHOD="POST">');
send('<CENTER><INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><INPUT TYPE="submit" Value="Accept/Reject more Articles"></FORM><P> ');
```

{Button to move to Assign Articles to Sessions}

```
send('<P><FORM ACTION=" ../cgi-win/Assign.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<P><INPUT TYPE="submit" Value="Now Assign the Articles to Sessions "
Size="30"></center></FORM>');
```

sendhr;

```
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM></BODY></HTML>');
closeStdout;
closeApp( application );
```

end; {with cgiEnvData1 do}

end; {FormCreate}

end.

Assign Articles: **D_Assign.exe**

```
unit D_assign;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Query1: TQuery;
    Sessions: TTable;
    Articles: TTable;
    SessionsNumber: TIntegerField;
    SessionsNeumonic: TStringField;
    Query1Number: TIntegerField;
    Query1Neumonic: TStringField;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  dbtitle, dbPaper_Number: string;

  dflag: string;           {valid program call}

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin

  with CGIEnvData1 do begin
```

```

webSiteINIFilename:=paramstr(1);
application.onException:=cgiErrorHandler;
application.processMessages;
createStdout;
sendPrologue;

```

```

send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');

```

{Determine whether flag is valid}

```

dflag:= getsmallfield ('flag');

```

```

if (dflag = '1') then begin   {flag is set}
send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

```

{Build Form}

```

send('<P><FORM ACTION=" ../cgi-win/Assign2.exe" METHOD="GET">');

```

{Pull Sessions from SESSION Table }

```

with Query1 do begin
close;
sql.clear;
sql.add('Select * FROM Session ');
open;

```

{Move records from query result to Session Pull-down menu}

```

Sessions.open;
Sessions.first;

```

```

send('<P><B>Assign the following Titles to Session: </B><SELECT
NAME="Session">');

```

```

while not Sessions.EOF do begin
send('<OPTION> ' +Sessions.fieldByName('Neumonic').AsString+ " ");
sessions.next;
end; {for all records in the query result}
send('</SELECT><P>');

```

```

end; {with query1}

```

```

with articles do begin

```

```

open;
first;
While not Articles.EOF do begin
    dbtitle:= fieldbyname('Title').asstring;
    dbPaper_Number:= fieldbyname('Paper_Number').asstring;
    send ('<INPUT TYPE="checkbox" NAME="Title" Value="" +dbPaper_Number+ ""> '
+dbtitle+ ' <BR>');
    next;
end; {while not EOF}
close;
end; {with articles do}

```

```

send('<P><CENTER><INPUT TYPE="submit" Value="Assign the selected articles to the
specified Session"><P> ');
send('<INPUT TYPE="reset" VALUE="Reset values"></CENTER></FORM>');
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

```

```

end {flag is set}

```

```

else begin    {flag **incorrect**}

```

```

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

```

```

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

```

```
end; {flag **incorrect**}  
end; {with cgiEnvData1 do}  
end; {FormCreate}  
end. {Application}
```


Order Articles: Assign2.exe

```
unit Assign_2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type
  TForm1 = class(TForm)
    Articles: TTable;
    DataSource1: TDataSource;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    procedure formcreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  i, j, dTitle: integer;

  dsession : string;
  dTitlein, dTitlestr: string;
  dString, AssignString: string;
  dnum: string;

  Titlelist : TStringList;

  function TitleToInt ( S: string) :integer;

implementation

{$R *.DFM}
{*****}
function TitleToInt ( S: string) :integer;

{This function is necessary because the string value of dTitle has several spaces in
it and the strtoint function cannot handle the conversion properly}
```

```

var
  temp: string;    {temporary variable to build new string into}
  j: integer;      {Counting Variable to keep track of how many characters are in the
number}

begin
  {set the counter to zero}
  j:= 0;

  {count the characters}
  repeat
    j:= j+1;
  until S[j+1] = '';

  {Copy the valid number characters over to temp}
  temp := Copy(S, 1, j);

  {Convert the temp string to an integer value}
  TitleToInt := strtoint (temp);

end; {TitleToInt}
{*****}

procedure TForm1.formcreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
    createStdout;
    sendPrologue;

    send ('<HTML><HEAD>');
    SendTitle('Asilomar Signals, Systems, and Computers');
    send ('</HEAD><BODY bgcolor=FFFFFF>');
    send('<center><H2>Asilomar Conference System Administration</H2></center>');
    sendhr;

    {Get query string from Assign.exe}
    Titlelist:= TStringlist.create;  {list of titles to assign to the session}

    AssignString:= CGIQueryString^;

```

j:= 0; {The counter for the individual strings within the QueryString}

for i := 8 to Length(AssignString) do {The counter for going through the AssignString -
eight is the first valuable character (the beginning of the Session Neumonic) }

```
if AssignString[i] <> '&' then begin {the character is valuable}
  dString[j] := AssignString[i];
  j:= j+1;
end {if AssignString[i] <> '&' then}
else begin {The & signifies the end of the string}
  TitleList.add (dString);
  dString:= '   '; {Blank out the old values}
  j:=0;           {reset the string counter}
  i:= i+5;        {Skip over the text "title="}
end; {else}
```

TitleList.add (dString); {This is to put the last string into the string list. It is
necessary because there is no & at the end}

{Place the Session Neumonic into each Paper_Number passed}

{The first item in the Title List is the Session Neumonic}

dSession:= TitleList[0];

{Send confirmation}

send ('<P>The following Articles were placed in Session ' +dSession+ '. ');

send ('Please enter the order number in the box next to the Title.
Note: Ensure
that a number is entered. The ');

send ('Order number will always be updated with this entry. ');

send ('If it is left blank, the order number will be blank.<P>');

{Start the FORM that will assign the order number within the session}

send('<P><FORM ACTION="..../cgi-win/Assign3.exe" METHOD="GET">');

send('<INPUT TYPE="hidden" Name="Sess" Value="" +dSession+ "">');

{The rest of the items are Paper_Numbers}

with Articles do begin

open; {Prepare the Table for use}

for i:= 1 to TitleList.count-1 do begin

{Get the next title number}

dTitlein:= TitleList[i];

```

{Convert the Title string to an integer value}
dTitle:= TitleToInt (dTitlein);

{Move to proper record}
first; {go back to the first record before every new search}
while fieldbyname('Paper_Number').asinteger < dTitle do
    next;

{Insert the Session}
edit;
Setfields ([nil, nil, nil, nil, nil, dSession]); {The remaining fields are ignored}
post;

{Convert counting variable "i" into string to send to HTML page}
dnum:= inttostr(i);

{Send title name}
send('<INPUT TYPE="text" Name="' +fieldbyname('Paper_Number').asstring+ "'
SIZE="2" VALUE="' +dnum+ "'> ');
send(' +fieldbyname('Title').asstring+ '<BR>');

end; {for}

close; { the table}

end; {with Articles}

send('<P><CENTER><INPUT TYPE="submit" Value="Assign the Articles in this
Order"></CENTER></FORM><P> ');

{Send Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with cgiEnvData1 do}
end; {FormCreate}

end. {Application}

```

Confirmation: Assign3.exe

```
unit Assign_3;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Query1: TQuery;
    Articles: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  i, j, dTitle: integer;

  dsession : string;
  dTitlein, dTitlestr: string;
  dString, AssignString: string;
  dnum: string;

  Titlelist : TStringList;

  function TitleToInt ( S: string) :integer;

implementation

{$R *.DFM}
{*****}
function TitleToInt ( S: string) :integer;

{This function is necessary because the string value of dTitle has several spaces in
```

it and the strtoint function cannot handle the conversion properly}

```
var
  temp: string;  {temporary variable to build new string into}
  j: integer;    {Counting Variable to keep track of how many characters are in the
number}

begin
  {set the counter to zero}
  j:= 0;

  {count the characters}
  repeat
    j:= j+1;
  until S[j+1] = '';

  {Copy the valid number characters over to temp}
  temp := Copy(S, 1, j);

  {Convert the temp string to an integer value}
  TitleToInt := strtoint (temp);

end; {TitleToInt}
{*****}

procedure TForm1.formcreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
    createStdout;
    sendPrologue;

    send ('<HTML><HEAD>');
    SendTitle('Asilomar Signals, Systems, and Computers');
    send ('</HEAD><BODY bgcolor=FFFFFF>');
    send('<center><H2>Asilomar Conference System Administration</H2></center>');
    sendhr;

    {Get query string from Assign.exe}
    Titlelist:= TStringlist.create;  {list of titles to assign to the session}
```

AssignString:= CGIQueryString^;

dString:= ' '; {Initialize as Blank }

j:= 1; {The counter for the individual strings within the QueryString}

for i := 6 to Length(AssignString) do {The counter for going through the AssignString - eight is the first valuable character (the beginning of the Session Neumonic) }

if (AssignString[i] <> '&') and (AssignString[i] <> '=') then begin {the character is valuable}

dString[j] := AssignString[i];

j:= j+1;

end {if AssignString[i] <> '&' then}

else begin {The & signifies the end of the string}

TitleList.add (dString);

dString:= ' '; {Blank out the old values}

j:=1; {reset the string counter}

end; {else}

TitleList.add (dString); {This is to put the last string into the string list. It is necessary because there is no & at the end}

{Place the Session Neumonic into each Paper_Number passed}

{Note: The string list consists of the following components -

[0] session #

[1] first paper-number

[2] first paper OrderInSession

[3] second paper-number

[4] second paper OrderInSession

. .

. .

. .

This is important to understand when interpreting the "for" loop!}

{The first item in the Title List is the Session Neumonic}

dSession:= TitleList[0];

{Send confirmation}

send ('<P>The following Articles have been placed in Session ' +dSession+ ' in the order indicated. The Session ');

send ('and the Orders may be changed at any time by repeating the process of Assigning an Article to a Session.<P>');

```

send('<pre>Order      Title<pre><BR>');

{The rest of the items are (Paper_Number and OrderInSession) pairs}
with Articles do begin

    open; {Prepare the Table for use}
    for i:= 1 to TitleList.count-1 do begin

        {Get the next Paper_Number}
        dTitlein:= TitleList[i];

        {Convert the Paper_Number string to an integer value}
        dTitle:= TitleToInt (dTitlein);

        {Move to proper record}
        first; {go back to the first record before every new search}
        while fieldbyname('Paper_Number').asinteger < dTitle do
            next;

        {***** Increment "i" to get to the OrderInSession number *****}
        i:=i+1;

        {Insert the Session}
        edit;
        Setfields ([nil, nil, nil, nil, nil, dSession, TitleList[i]]); {The remaining fields are
ignored}
        post;

        {Send title name}
        send(" +TitleList[i]+ ' ' +fieldbyname('Title').asstring+ '<BR>');

        end; {for}

    close; { the table}

    end; {with Articles}

send('<P><FORM ACTION=" ../cgi-win/Assign.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Assign more
Articles"></CENTER></FORM><P> ');

send('<P><FORM ACTION=" ../cgi-win/Times.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');

```



```
send('<P><CENTER><INPUT TYPE="submit" Value="Assign Presentation Times to
Articles"></CENTER></FORM><P> ');
```

{Send Footer}

```
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send('</BODY></HTML>');
closeStdout;
closeApp( application );
```

```
end; {with cgiEnvData1 do}
```

```
end; {FormCreate}
```

```
end. {Application}
```

Choose Session: **Time1.exe**

```
unit Time_1;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Sessions: TTable;
    Query1: TQuery;
    procedure FormCreate(Sender: TObject);
    procedure CGIDB1SendingHotField(currentRecord: TDataset;
      var s: OpenString);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  dflag: string;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do begin
    {Standard Header}
    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
```

```

sendPrologue;
send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');

```

{Determine whether flag is valid}

```
dflag:= getsmallfield ('flag');
```

```
if (dflag = '1') then begin  {flag is set}
```

```
send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;
```

```
{send instruction;}
```

```
send('<center>These are the current Session Neumonics.<br>');
```

```
send('To enter Presentation Times for Articles within a Session, simply click on the Session
Neumonic.<BR>');
```

{Retrieve current Session Information}

```
with Query1 do begin
```

```
close;
sql.clear;
sql.add('Select*FROM Session ');
open;
```

```
if recordcount > 0 then begin
```

```
Sessions.Open;
CGIDB1.DrawTable;
send('</center>');
Sessions.Close;
end {if recordcount > 0}
```

```
else
```

```
sendhdr ('2','There are no Sessions in the Database!');
```

```
end; {with Query1}
```

{Standard Footer}

```
sendhr;
```

```

    send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}

```

```

else begin {flag **incorrect**}

```

```

    send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
    sendhr;
    send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
    send ('<P>Please ensure that you are authorized to access this information. ');
    send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
    send ('<P>If you are not authorized to access this information, please note that a log is');
    send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
    send ('prosecution for your actions.<BR>');

```

```

    send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
    send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
    send ('Conference On-line Submission Page</A>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

```

```

    end; {flag **incorrect**}

```

```

    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );
    end; {with CGIEnvData1}

```

```

end; {FormCreate}

```

```

procedure TForm1.CGIDB1SendingHotField(currentRecord: TDataset;
var s: OpenString);
begin
    {Hotlink for single session edit}
    s:='<A HREF=" ../cgi-win/Times2.exe?'+ currentrecord.fieldbyName('Number').AsString + "'>
+ currentrecord.FieldbyName('Neumonic').AsString + '</A>';

```

```

end;

```

end.

Assign Times: Times2.exe

```
unit Times_2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Sessions: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    Articles: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  ArticlesInSessionList: TStringList;
  OrderInSessionList: TStringList;
  ArticlesInOrderList: TStringList;

  nextsession: string;
  SessionNumber,
  Session,
  dTitle,
  dbPaper_Number,
  dbtitle,
  dbOrderInSession,
  dbPresentationTime: string;

  i, j, k,
  order,
  dTitleint :integer;

implementation
```

{ \$R *.DFM }

procedure TForm1.FormCreate(Sender: TObject);

begin

with CGIEnvData1 do begin

webSiteINIFilename:=paramstr(1);
application.onException:=cgiErrorHandler;
application.processMessages;

createStdout;
sendPrologue;

{Standard Header}
send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

{Find Session to edit}

{ Get field from search page }
SessionNumber:= cgiQueryString^;

{Find Number in SESSIONS Table}
with Sessions do begin

{Prepare the table}
open;
first;

{find the correct Session}
while SessionNumber <> fieldbyname('Number').asstring do
next;

Session:= fieldbyname('Neumonic').asstring;

send('Here are the Articles in Session ' +Session+ ' for which you wanted to Assign
Presentation Times. ');
send('Fill in the Times and "Submit" them.
');

```

send('<P><FORM ACTION=" ../cgi-win/Times3.exe" METHOD="GET">');
send('<INPUT TYPE="hidden" Name="Sess" Value="" +Session+ ">');

```

```

end; {with Sessions}

```

{Find the Articles in this session}

```

{Create the stringlists}
ArticlesInSessionList:= TStringlist.create;
OrderInSessionList:= TStringlist.create;
ArticlesInOrderList:= TStringlist.create;

```

```

{Clear the stringlists}
ArticlesInSessionList.clear;
OrderInSessionList.clear;
ArticlesInOrderList.clear;

```

```

{Get the Article Record }
with Articles do begin

```

```

{Prepare Article Table}
Articles.open;
Articles.first;      {Start at the beginning of the ARTICLE Table}

```

```

{Check every article against the session neumonic. Matches go on InSession stringlist}
while not Articles.EOF do begin

```

```

    if Articles.fieldByName ('Session').asstring = Session then begin

```

```

        {Put the paper_number on a list and the OrderInSession on another - they
must be symmetrical}

```

```

        ArticlesInSessionList.add (fieldbyname('Paper_Number').asstring);
        OrderInSessionList.add (fieldbyname('OrderInSession').asstring);
        end; {if then}

```

```

        Articles.next;
        end; {while not articles EOF}

```



```

    end; {with Articles}

if ArticlesInSessionList.count > 0 then begin

    {Put Articles in order in the InOrder stringlist}

    {First initialize the list by placing a dummy value in the string for each value that
will be added}
    for j:= 0 to ArticlesInSessionList.count do {the [0] place will not be used but must be
included to get the order numbers to start at [1]}

        ArticlesInOrderList.add ('an empty field to hold the place');

    {Now go through the list of articles and put them in the "ArticlesInOrderList" in
order (as saved in the "OrderInSessionList")}
    for j:= 0 to ArticlesInSessionList.count-1 do begin

        {get the OrderInSessionList value and make it an integer}
        Order:= strtoint(OrderInSessionList[j]);

        {assign the article title to the InOrderList in its place}
        ArticlesInOrderList[Order]:= ArticlesInSessionList[j];
        end;

    {Go through the Articles in OrderList and retrieve information and print it out}
    for j:= 1 to ArticlesInOrderList.count-1 do begin

        { Move to proper record in ARTICLE Table}
        Articles.open;
        Articles.first;

        dTitle:= ArticlesInOrderList[j];

        {make ArticlesInOrderList[j] an integer for comparisom}
        dTitleint:= strtoint(dTitle);
        while Articles.fieldByName('Paper_Number').asinteger < dTitleint do
            Articles.next;

        { Retrieve record }
        with Articles do begin
            dbPaper_Number:= fieldByName ('Paper_Number').asstring;
            dbtitle:= fieldByName ('title').asstring;
            dbOrderInSession:= fieldByName ('OrderInSession').asstring;
            dbPresentationTime:= fieldByName ('PresentationTime').asstring;

```

```

{Send Session edit FORM to HTML page}
send(" + dbOrderInSession+ ' ');
send('<INPUT TYPE="text" Name="" +dbPaper_Number+ "" SIZE="10" Value=""
+dbPresentationTime+ "">');
send(' ' +dbtitle+ '<BR>');

end; {with Articles}

end; {for i= 1 to articleinOrderList}

send('<P><CENTER><INPUT TYPE="submit" Value="Submit Presentation Times">');
send('<INPUT TYPE="reset" VALUE="Reset to Original
Session"></CENTER></FORM>');

end; {if count > 0}

{Standard Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {with cgiEnvData1 do}

end; {FormCreate}
end.

```

Confirmation: **Times3.exe**

```
unit Times_3;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Articles: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  i, j, dTitle: integer;

  dsession : string;
  dTitlein, dTitlestr: string;
  dString, AssignString: string;
  dnum: string;

  Titlelist : TStringList;

  function TitleToInt ( S: string) :integer;

implementation

{$R *.DFM}
{*****}
function TitleToInt ( S: string) :integer;

{This function is necessary because the string value of dTitle has several spaces in
it and the strtoint function cannot handle the conversion properly}
```

```

var
  temp: string;    {temporary variable to build new string into}
  j: integer;      {Counting Variable to keep track of how many characters are in the
number}

begin
  {set the counter to zero}
  j:= 0;

  {count the characters}
  repeat
    j:= j+1;
  until S[j+1] = '';

  {Copy the valid number characters over to temp}
  temp := Copy(S, 1, j);

  {Convert the temp string to an integer value}
  TitleToInt := strtoint (temp);

end; {TitleToInt}
{*****}

procedure TForm1.formcreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
    createStdout;
    sendPrologue;

    send('<HTML><HEAD>');
    SendTitle('Asilomar Signals, Systems, and Computers');
    send('</HEAD><BODY bgcolor=FFFFFF>');
    send('<center><H2>Asilomar Conference System Administration</H2></center>');
    sendhr;

    {Get query string from Time2.exe}
    Titlelist:= TStringlist.create;  {list of titles to assign to the session}

    AssignString:= CGIQueryString^;

```

dString:= ' '; {Initialize as Blank }

j:= 1; {The counter for the individual strings within the QueryString}

for i := 6 to Length(AssignString) do {The counter for going through the AssignString - eight is the first valuable character (the beginning of the Presentation Time) }

if (AssignString[i] <> '&') and (AssignString[i] <> '=') then begin {the character is valuable}

dString[j] := AssignString[i];

j:= j+1;

end {if AssignString[i] <> '&' then}

else begin {The & signifies the end of the string}

TitleList.add (dString);

dString:= ' '; {Blank out the old values}

j:=1; {reset the string counter}

end; {else}

TitleList.add (dString); {This is to put the last string into the string list. It is necessary because there is no & at the end}

{Place the Presentation Time into each Paper_Number passed}

{Note: The string list consists of the following components -

[0] session #

[1] first paper-number

[2] first paper Presentation Time

[3] second paper-number

[4] second paper Presentation Time

. .

. .

. .

This is important to understand when interpreting the "for" loop!}

{The first item in the Title List is the Session Neumonic}

dSession:= TitleList[0];

{Send confirmation}

send ('<P>The following Articles have been placed in Session ' + dSession + ' in the order indicated. The Session ');

send ('and the Orders in the Session may be changed at any time by repeating the process of ');

send ('Assigning an Article to a Session.<P><pre>Order Title<pre>
');

{The rest of the items are (Paper_Number and Presentation Time) pairs}
with Articles do begin

open; **{Prepare the Table for use}**
for i:= 1 to TitleList.count-1 do begin

{Get the next Paper_Number}
dTitlein:= TitleList[i];

{Convert the Paper_Number string to an integer value}
dTitle:= TitleToInt (dTitlein);

{Move to proper record}
first; **{go back to the first record before every new search}**
while fieldbyname('Paper_Number').asinteger <> dTitle do
 next;

{*** Increment "i" to get to the Presentation Time *****}**
{This causes the "for" loop to skip every other element}
i:=i+1;

{Insert the Session}
edit;
Setfields ([nil, nil, nil, nil, nil, nil, TitleList[i]]); **{The remaining fields are ignored}**
post;

{Send title name}
send(" +TitleList[i]+ ' ' +fieldbyname('Title').asString+ '
');

end; **{for}**

close; **{ the table}**

end; **{with Articles}**

send('<P><FORM ACTION="../../cgi-win/Times.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Assign more Presentation Times"></CENTER></FORM><P> ');

{Send Footer}
sendhr;
send('<P><FORM ACTION="../../cgi-win/Adminpwd.exe" METHOD="POST">');

```
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    send('</BODY></HTML>');
    closeStdout;
    closeApp( application );

    end; {with cgiEnvData1 do}

end; {FormCreate}

end.
```

Selection options: **PassWds.exe**

```
unit Pass_wds;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgidb, Cgi, DB, DBTables;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  dUserID: string;
  dpwd: string;
  dbpassword: string;
  dflag: string;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do begin
    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
    sendPrologue;
```



```

send('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send('</HEAD><BODY bgcolor=FFFFFF>');

```

{Determine whether flag is valid}

```
dflag:= getsmallfield('flag');
```

```
if (dflag = '1') then begin  {flag is set}
```

```

send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

```

{Send Instructions}

```

sendhdr('3', 'System Admin user passwords. ');
send('To change a System Admin password, enter the information below and "Modify a
System Admin User Password." ');
send(' The new password must be entered twice to ensure validity. To ADD or DELETE a
User, select the appropriate ');
send('Buttons below. ');

```

```

send('<P><CENTER><ALIGN LEFT><FORM ACTION=" ../cgi-win/Pwdmod.exe"
METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<B>User name:      </B><INPUT NAME="userid" Size="20" TYPE="text"><BR>');
send('<B>Old Password:    </B><INPUT NAME="oldpwd" Size="20"
TYPE="password"><BR>');
send('<B>New Password:    </B><INPUT NAME="newpwd1" Size="20"
TYPE="password"><BR>');
send('<B>New Password Again:</B><INPUT NAME="newpwd2" Size="20"
TYPE="password"><BR></ALIGN left>');
send('<INPUT NAME="action" TYPE="hidden" Value="admin">');
send('<P><INPUT TYPE="submit" Value="Modify System Admin User Password">');
send('<INPUT TYPE="reset" Value="Clear Values"></FORM><BR>');
send('<br>*****');

```

```

send('<P><FORM ACTION=" ../cgi-win/PwdAdd.exe" METHOD="POST">');
send('<INPUT NAME="flag" Size="3" TYPE="hidden" Value="1">');
send('<INPUT NAME="action" TYPE="hidden" Value="admin">');
send('<P><INPUT TYPE="submit" Value="Add a new System Admin User"></FORM>');

```

```

send('<P><FORM ACTION=" ../cgi-win/PwdDel.exe" METHOD="POST">');
send('<INPUT NAME="flag" Size="3" TYPE="hidden" Value="1">');
send('<INPUT NAME="action" TYPE="hidden" Value="admin">');

```

```

send('<P><INPUT TYPE="submit" Value=" Delete a System Admin User
"></CENTER></FORM><BR>');

```

```

sendhr;
sendhdr('3', 'Reviewer passwords. ');
send('To change a reviewer password, enter the information below and "Modify a Reviewer
Password." ');
send(' The new password must be entered twice to ensure validity. To ADD or DELETE a
Reviewer, select the ');
send('appropriate Buttons below. ');

```

```

send('<P><CENTER><FORM ACTION=" ../cgi-win/Pwdmod.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<B>Reviewer name: </B><INPUT NAME="userid" Size="20"
TYPE="text"><BR>');
send('<B>Old Password: </B><INPUT NAME="oldpwd" Size="20"
TYPE="password"><BR>');
send('<B>New Password: </B><INPUT NAME="newpwd1" Size="20"
TYPE="password"><BR>');
send('<B>New Password Again:</B><INPUT NAME="newpwd2" Size="20"
TYPE="password"><BR>');
send('<INPUT NAME="action" TYPE="hidden" Value="reviewer">');
send('<P><INPUT TYPE="submit" Value="Modify Reviewer Password">');
send('<INPUT TYPE="reset" Value="Clear Values"></FORM><BR>');
send('<br>*****');

```

```

send('<P><FORM ACTION=" ../cgi-win/PwdAdd.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<INPUT NAME="action" TYPE="hidden" Value="reviewer">');
send('<P><INPUT TYPE="submit" Value="Add a new Reviewer"></FORM>');

```

```

send('<P><FORM ACTION=" ../cgi-win/PwdDel.exe" METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<INPUT NAME="action" TYPE="hidden" Value="reviewer">');
send('<P><INPUT TYPE="submit" Value="Delete a Reviewer"
Size="30"></FORM></CENTER><P>');

```

```

{Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
end {flag is set}

```

```
else begin {flag **incorrect**}
```

```
send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF="../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');
```

```
send ('<P><hr><P><IMG SRC="../prototype/asil2b0.gif" align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );
```

```
end; {flag **incorrect**}
```

```
send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}
```

```
end; {FormCreate}
```

```
end.
```

Modify Password: PwdMod.exe

```
unit Pwd_mod;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type
  TForm1 = class(TForm)
    Admin: TTable;
    Reviewer: TTable;
    DataSource1: TDataSource;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
var
  dflag: string;
  duserid: string;  {ID for either the Admin user or the reviewer}
  doldpwd: string;
  dnewpwd1, dnewpwd2: string;
  daction: string;

begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
```

```
createStdout;  
sendPrologue;
```

{Standard Header}

```
send ('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send ('</HEAD><BODY bgcolor=FFFFFF>');  
send ('<center><H2>Asilomar Conference System Administration</H2>');  
sendhr;
```

{ Get field from search page }

```
dflag:= getsmallfield ('flag');  
duserid:= getsmallfield ('userid');  { gets the ID for either the Admin user or the  
reviewer}  
doldpwd:= getsmallfield ('oldpwd');  
dnewpwd1:= getsmallfield ('newpwd1');  
dnewpwd2:= getsmallfield ('newpwd2');  
daction:= getsmallfield ('action');
```

{check flag}

```
if dflag = '1' then begin
```

```
    if dnewpwd1 = dnewpwd2 then begin
```

{If action is admin then modify admin table, else modify reviewer Table}

```
    if daction = 'admin' then begin
```

```
        with Admin do begin
```

{Update record }

{Move to record}

```
    open;  
    first;  
    while (fieldbyname('UserID').asstring <> duserid) and (not EOF) do  
        next;
```

```
    if (fieldbyname('UserID').asstring = duserid) then begin  
        edit;  
        setfields([nil, nil, dnewpwd1]);  
        post;  
        close;
```

```

    sendhdr('3', 'The System Admin Password has been changed!');
end

else begin

    send('<P>User not found, please try again. ');
    send('<P><CENTER><ALIGN LEFT><FORM ACTION=" ../cgi-win/Pwdmod.exe"
METHOD="POST">');
    send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
    send('<B>User name:      </B><INPUT NAME="userid" Size="20"
TYPE="text"><BR>');
    send('<B>Old Password:   </B><INPUT NAME="oldpwd" Size="20"
TYPE="password"><BR>');
    send('<B>New Password:   </B><INPUT NAME="newpwd1" Size="20"
TYPE="password"><BR>');
    send('<B>New Password Again:</B><INPUT NAME="newpwd2" Size="20"
TYPE="password"><BR></ALIGN left>');
    send('<INPUT NAME="action" TYPE="hidden" Value="admin">');
    send('<P><INPUT TYPE="submit" Value="Modify System Admin User Password">');
    send('<INPUT TYPE="reset" Value="Clear Values"></FORM>');
    end;

end;    {with admin}

end {if daction = admin}

else if daction = 'reviewer' then begin

    with Reviewer do begin

        {Update record }

        {Move to record}
        open;
        first;
        while (fieldname('ReviewerID').asString <> duserid) and (not EOF) do
            next;

        if (fieldname('ReviewerID').asString = duserid) then begin
            edit;
            setfields([nil, nil, dnewpwd1]);
            post;
            close;
            sendhdr('3', 'The Reviewer Password has been changed!');
            end

```

else begin

```
    send('<P>Reviewer not found, please try again.');
```

```
    send('<P><CENTER><ALIGN LEFT><FORM ACTION=" ../cgi-win/Pwdmod.exe" METHOD="POST">');
```

```
    send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
```

```
    send('<B>Reviewer name:      </B><INPUT NAME="userid" Size="20" TYPE="text"><BR>');
```

```
    send('<B>Old Password:      </B><INPUT NAME="oldpwd" Size="20" TYPE="password"><BR>');
```

```
    send('<B>New Password:      </B><INPUT NAME="newpwd1" Size="20" TYPE="password"><BR>');
```

```
    send('<B>New Password Again:</B><INPUT NAME="newpwd2" Size="20" TYPE="password"><BR></ALIGN left>');
```

```
    send('<INPUT NAME="action" TYPE="hidden" Value="reviewer">');
```

```
    send('<P><INPUT TYPE="submit" Value="Modify Reviewer Password">');
```

```
    send('<INPUT TYPE="reset" Value="Clear Values"></FORM>');
```

```
    end;
```

```
end;    {with reviewer}
```

end; {if daction = reviewer}

```
send('<P><FORM ACTION=" ../cgi-win/Passwds.exe" METHOD="POST">');
```

```
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
```

```
send('<P><CENTER><INPUT TYPE="submit" Value="Modify more Passwords"></CENTER></FORM>');
```

end {new passwords matched}

else begin

{new passwords don't match}

```
send('<P>The new passwords did not match, please try again.');
```

```
    send('<P><CENTER><ALIGN LEFT><FORM ACTION=" ../cgi-win/Pwdmod.exe" METHOD="POST">');
```

```
    send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
```

```
    send('<B>User name:      </B><INPUT NAME="userid" Size="20" TYPE="text"><BR>');
```

```
    send('<B>Old Password:      </B><INPUT NAME="oldpwd" Size="20" TYPE="password"><BR>');
```

```
    send('<B>New Password:      </B><INPUT NAME="newpwd1" Size="20" TYPE="password"><BR>');
```

```
    send('<B>New Password Again:</B><INPUT NAME="newpwd2" Size="20" TYPE="password"><BR></ALIGN left>');
```

```

send('<INPUT NAME="action" TYPE="hidden" Value="" +daction+ ">');

if daction = 'admin' then send('<P><INPUT TYPE="submit" Value="Modify System
Admin Password">')
else send('<P><INPUT TYPE="submit" Value="Modify Reviewer Password">');

send('<INPUT TYPE="reset" Value="Clear Values"></CENTER></FORM><BR>');
end; {if then else}

{Standard Footer}
sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');

end {flag is set}

else begin   {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {flag **incorrect**}

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}

```


end; {FormCreate}

end.

Add new user/reviewer: **PwdAdd.exe**

```
unit Pwd_add;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Admin: TTable;
    Reviewer: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
var
  dflag: string;
  daction: string;

begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
    sendPrologue;
```

{Standard Header}

```
send('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send('</HEAD><BODY bgcolor=FFFFFF>');
send('<center><H2>Asilomar Conference System Administration</H2>');
sendhr;
```

{ Get field from search page }

```
dflag:= getsmallfield ('flag');
daction:= getsmallfield ('action');
```

{check flag}

```
if dflag = '1' then begin
```

{If action is admin then modify admin table, else modify reviewer Table}

```
if daction = 'admin' then begin
```

```
    send('<P>To ADD a new System Admin User, please enter the following. The password is entered ');
```

```
    send('twice to ensure validity');
```

```
    send('<P><CENTER><ALIGN LEFT><FORM ACTION="../../cgi-win/PwdAdd2.exe" METHOD="POST">');
```

```
    send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
```

```
    send('<B>New User name:    </B><INPUT NAME="userid" Size="20" TYPE="text"><BR>');
```

```
    send('<B>Password:    </B><INPUT NAME="newpwd1" Size="20" TYPE="password"><BR>');
```

```
    send('<B>Password Again:</B><INPUT NAME="newpwd2" Size="20" TYPE="password"><BR></ALIGN left>');
```

```
    send('<INPUT NAME="action" TYPE="hidden" Value="" +daction+ "">');
```

```
    send('<P><INPUT TYPE="submit" Value="ADD this New System Admin User">');
```

```
    send('<INPUT TYPE="reset" Value="Clear Values"></CENTER></FORM><BR>');
```

```
end {if daction then}
```

```
else begin
```

```
    send('<P>To ADD a new Reviewer, please enter the following. The password is entered ');
```

```
    send('twice to ensure validity');
```

```

        send('<P><CENTER><ALIGN LEFT><FORM ACTION="../../cgi-win/PwdAdd2.exe"
METHOD="POST">');
        send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
        send('<B>New Reviewer name:      </B><INPUT NAME="userid" Size="20"
TYPE="text"><BR>');
        send('<B>Password:      </B><INPUT NAME="newpwd1" Size="20"
TYPE="password"><BR>');
        send('<B>Password Again:</B><INPUT NAME="newpwd2" Size="20"
TYPE="password"><BR></ALIGN left>');
        send('<INPUT NAME="action" TYPE="hidden" Value="" +daction+ "">');
        send('<P><INPUT TYPE="submit" Value="ADD this New Reviewer">');
        send('<INPUT TYPE="reset" Value="Clear Values"></CENTER></FORM><BR>');
        end; {if daction then}

```

{Standard Footer}

```

        sendhr;
        send('<P><FORM ACTION="../../cgi-win/Adminpwd.exe" METHOD="POST">');
        send('<INPUT TYPE="hidden" Name="flag" Value="1">');
        send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');

```

end {flag is set}

else begin {flag **incorrect**}

```

        send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
        sendhr;
        send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
        send ('<P>Please ensure that you are authorized to access this information. ');
        send ('<P>If you made an error, then please <A HREF="../../prototype/admin20.htm">try to login
again.</A>');
        send ('<P>If you are not authorized to access this information, please note that a log is');
        send ('maintained that includes your URL.  You may be subject to civil and/or criminal ');
        send ('prosecution for your actions.<BR>');

```

```

        send ('<P><hr><P><IMG SRC="../../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
        send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
        send ('Conference On-line Submission Page</A>');
        send ('</BODY></HTML>');
        closeStdout;
        closeApp( application );

```

end; {flag **incorrect**}

```
send ('</BODY></HTML>');  
closeStdout;  
closeApp( application );  
end; {with CGIEnvData1}  
  
end; {FormCreate}  
  
end.
```

Addition confirmation: **PwdAdd2.exe**

```
unit Pwd_add2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    Admin: TTable;
    DataSource1: TDataSource;
    Reviewer: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
var
  dflag: string;
  duserid: string;  {ID for either the Admin user or the reviewer}
  dnewpwd1, dnewpwd2: string;
  daction: string;

begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.OnException:=cgiErrorHandler;
    application.ProcessMessages;
```

```
createStdout;  
sendPrologue;
```

{Standard Header}

```
send('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send('</HEAD><BODY bgcolor=FFFFFF>');  
send('<center><H2>Asilomar Conference System Administration</H2>');  
sendhr;
```

{ Get field from search page }

```
dflag:= getsmallfield('flag');  
duserid:= getsmallfield('userid');  { gets the ID for either the Admin user or the  
reviewer}  
dnewpwd1:= getsmallfield('newpwd1');  
dnewpwd2:= getsmallfield('newpwd2');  
daction:= getsmallfield('action');
```

{check flag}

```
if dflag = '1' then begin
```

```
    if (dnewpwd1 = dnewpwd2) and (dnewpwd1 <> "") then begin
```

{If action is admin then modify admin table, else modify reviewer Table}

```
    if daction = 'admin' then begin
```

```
        with Admin do begin
```

{Append record }

```
            open;  
            AppendRecord([nil, duserid, dnewpwd1]);  
            close;  
        end;  {with admin}
```

```
sendhdr('3', 'The System Admin User "<B>'+duserid+ '</B>' has been added!');  
end {if daction = admin}
```

```
else if daction = 'reviewer' then begin
```

```
    with Reviewer do begin
```

{Append record }

```
open;  
AppendRecord([nil, duserid, dnewpwd1]);  
close;  
end; {with reviewer}
```

```
sendhdr('3', 'The Reviewer "<B>' + duserid + '</B>' has been added!');  
end; {if daction = reviewer}
```

```
send('<P><FORM ACTION=" ../cgi-win/Passwds.exe" METHOD="POST">');  
send('<INPUT TYPE="hidden" Name="flag" Value="1">');  
send('<P><CENTER><INPUT TYPE="submit" Value="Modify more  
Passwords"></CENTER></FORM>');
```

end {new passwords matched}

else begin

{new passwords don't match}

```
send('<P>The new passwords did not match, please try again.');
```

```
send('<P><CENTER><ALIGN LEFT><FORM ACTION=" ../cgi-win/Pwdadd2.exe"  
METHOD="POST">');  
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');  
send('<B>User name:      </B><INPUT NAME="userid" Size="20"  
TYPE="text"><BR>');  
send('<B>New Password:    </B><INPUT NAME="newpwd1" Size="20"  
TYPE="password"><BR>');  
send('<B>New Password Again:</B><INPUT NAME="newpwd2" Size="20"  
TYPE="password"><BR></ALIGN left>');  
send('<INPUT NAME="action" TYPE="hidden" Value="" + daction + "">');  
if daction = 'admin' then send('<P><INPUT TYPE="submit" Value="ADD System Admin  
User">')  
else send('<P><INPUT TYPE="submit" Value="ADD New Reviewer">');  
send('<INPUT TYPE="reset" Value="Clear Values"></CENTER></FORM><BR>');  
end;
```

{Standard Footer}

```
sendhr;  
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');  
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
```



```
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin  
Options page"></CENTER></FORM>');
```

```
end {flag is set}
```

```
else begin {flag **incorrect**}
```

```
send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');  
sendhr;  
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');  
send ('<P>Please ensure that you are authorized to access this information.');
```

send ('<P>If you made an error, then please try to login
again.');

send ('<P>If you are not authorized to access this information, please note that a log is');

send ('maintained that includes your URL. You may be subject to civil and/or criminal ');

send ('prosecution for your actions.
');

```
send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');  
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');  
send ('Conference On-line Submission Page</A>');  
send ('</BODY></HTML>');  
closeStdout;  
closeApp( application );
```

```
end; {flag **incorrect**}
```

```
send ('</BODY></HTML>');  
closeStdout;  
closeApp( application );  
end; {with CGIEnvData1}
```

```
end; {FormCreate}
```

```
end.
```

Delete a user/reviewer: **PwdDel.exe**

```
unit Pwd_del;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Admin: TTable;
    Reviewer: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
var
  dflag: string;
  daction: string;

begin
  with CGIEnvData1 do begin
    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
    sendPrologue;
```

{Standard Header}

```
send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
send ('<center><H2>Asilomar Conference System Administration</H2>');
sendhr;
```

{ Get field from search page }

```
dflag:= getsmallfield ('flag');
daction:= getsmallfield ('action');
```

{check flag}

```
if dflag = '1' then begin
```

{If action is admin then modify admin table, else modify reviewer Table}

```
if daction = 'admin' then begin
```

```
send('<P>To DELETE a System Admin User, please enter the following.');
```

```
send('<P><CENTER><ALIGN LEFT><FORM ACTION=" ../cgi-win/PwdDel2.exe"
METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<B>User name:      </B><INPUT NAME="userid" Size="20"
TYPE="text"><BR>');
send('<B>Password:     </B><INPUT NAME="oldpwd" Size="20"
TYPE="password"><BR>');
send('<INPUT NAME="action" TYPE="hidden" Value="" +daction+ "">');
send('<P><INPUT TYPE="submit" Value="Delete this System Admin User">');
send('<INPUT TYPE="reset" Value="Clear Values"></CENTER></FORM><BR>');
end {if daction then}
```

```
else begin
```

```
send('<P>To DELETE a Reviewer, please enter the following.');
```

```
send('<P><CENTER><ALIGN LEFT><FORM ACTION=" ../cgi-win/PwdDel2.exe"
METHOD="POST">');
send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
send('<B>Reviewer name:    </B><INPUT NAME="userid" Size="20"
TYPE="text"><BR>');
```

```

        send('<B>Password:  </B><INPUT NAME="oldpwd" Size="20"
TYPE="password"><BR>');
        send('<INPUT NAME="action" TYPE="hidden" Value="" +daction+ "">');
        send('<P><INPUT TYPE="submit" Value="Delete this Reviewer">');
        send('<INPUT TYPE="reset" Value="Clear Values"></CENTER></FORM><BR>');
        end; {if daction then}

{Standard Footer}
sendhr;
send('<P><FORM ACTION="../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');

end {flag is set}

else begin    {flag **incorrect**}

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF="../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL.  You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

send ('<P><hr><P><IMG SRC="../prototype/asil2b0.gif" align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

        end; {flag **incorrect**}

send ('</BODY></HTML>');
closeStdout;
closeApp( application );
end; {with CGIEnvData1}

end; {FormCreate}

```

end.

Deletion confirmation: **PwdDel2.exe**

```
unit Pwd_del2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DB, DBTables;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Admin: TTable;
    Reviewer: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
var
  dflag: string;
  duserid: string;  {ID for either the Admin user or the reviewer}
  doldpwd: string;
  daction: string;

begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;
```

```
createStdout;  
sendPrologue;
```

{Standard Header}

```
send ('<HTML><HEAD>');  
SendTitle('Asilomar Signals, Systems, and Computers');  
send ('</HEAD><BODY bgcolor=FFFFFF>');  
send ('<center><H2>Asilomar Conference System Administration</H2>');  
sendhr;
```

{ Get field from search page }

```
dflag:= getsmallfield ('flag');  
duserid:= getsmallfield ('userid'); { gets the ID for either the Admin user or the  
reviewer}  
doldpwd:= getsmallfield ('oldpwd');  
daction:= getsmallfield ('action');
```

{check flag}

```
if dflag = '1' then begin
```

{If action is admin then modify admin table, else modify reviewer Table}

```
if daction = 'admin' then begin
```

```
with Admin do begin
```

{Update record }

{Move to record}

```
open;  
first;  
while (fieldname('UserID').asString <> duserid) and (not EOF) do  
    next;
```

```
if not EOF then begin
```

```
    edit;  
    delete;  
    close;
```

```
    sendhdr('3', 'The System Admin User "<i>'+duserid+ '</i>" has been Deleted!');  
end
```

```

else begin

    send('<P>The User "<B>' + duserid + '</B>' was not found. please try again. ');
    send('<P><CENTER><ALIGN LEFT><FORM ACTION=" ../cgi-win/Pwddel.exe"
METHOD="POST">');
    send('<INPUT NAME="flag" TYPE="hidden" Value="1">');
    send('<B>User name:      </B><INPUT NAME="userid" Size="20"
TYPE="text"><BR>');
    send('<B>Old Password:   </B><INPUT NAME="oldpwd" Size="20"
TYPE="password"><BR>');
    send('<INPUT NAME="action" TYPE="hidden" Value="" + daction + "">');
    send('<P><INPUT TYPE="submit" Value="Delete this System Admin User">');
    send('<INPUT TYPE="reset" Value="Clear Values"></CENTER></FORM>');
    end; {else}

end; {with admin}

end {if daction = admin}

else if daction = 'reviewer' then begin

    with Reviewer do begin

        {Delete record }

        {Move to record}
        open;
        first;
        while (fieldbyname('ReviewerID').asString <> duserid) and (not EOF) do
            next;

        if not EOF then begin
            edit;
            delete;
            close;

            sendhdr('3', 'The Reviewer "<i>' + duserid + '</i>' has been Deleted!');
            end

        else begin

            send('<P>The Reviewer "<B>' + duserid + '</B>' was not found. please try again. ');
            send('<P><CENTER><ALIGN LEFT><FORM ACTION=" ../cgi-win/Pwddel.exe"
METHOD="POST">');
            send('<INPUT NAME="flag" TYPE="hidden" Value="1">');

```



```

        send('<B>User name:      </B><INPUT NAME="userid" Size="20"
TYPE="text"><BR>');
        send('<B>Old Password:    </B><INPUT NAME="oldpwd" Size="20"
TYPE="password"><BR>');
        send('<INPUT NAME="action" TYPE="hidden" Value="" +daction+ ">');
        send('<P><INPUT TYPE="submit" Value="Delete this Reviewer">');
        send('<INPUT TYPE="reset" Value="Clear Values"></CENTER></FORM>');
        end; {else}

```

```

    end;    {with reviewer}

```

```

end; {if daction = reviewer}

```

```

send('<P><FORM ACTION=" ../cgi-win/Passwds.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Modify more
Passwords"></CENTER></FORM>');

```

{Standard Footer}

```

sendhr;

```

```

send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');

```

```

end {flag is set}

```

```

else begin    {flag **incorrect**}

```

```

send ('<center><H2>Asilomar Conference on Signals, Systems, & Computers</H2></center>');
sendhr;
send ('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
send ('<P>Please ensure that you are authorized to access this information. ');
send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to login
again.</A>');
send ('<P>If you are not authorized to access this information, please note that a log is');
send ('maintained that includes your URL.  You may be subject to civil and/or criminal ');
send ('prosecution for your actions.<BR>');

```

```

send ('<P><hr><P><IMG SRC=" ../prototype/asil2b0.gif " align=left alt="Asilomar facility">');
send ('Return to the <A HREF="http://131.120.20.70/Index.html">Asilomar');
send ('Conference On-line Submission Page</A>');
send ('</BODY></HTML>');
closeStdout;

```

```
closeApp( application );  
  
    end; {flag **incorrect**}  
  
    send ('</BODY></HTML>');  
    closeStdout;  
    closeApp( application );  
    end; {with CGIEnvData1}  
  
end; {FormCreate}  
  
end.
```

This single program: **P_Sess.exe**

```
unit Prsess;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DBTables, DB;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Query1: TQuery;
    Sessions: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin

  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
    sendPrologue;

    send ('<HTML><HEAD>');
```

```

SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');

end;

with cgiEnvData1 do begin

    send('<center><H2>Asilomar Conference System Administration</H2></center>');
    sendhr;

    send('<P><h3>These are the current Sessions Scheduled in the
database:</h3><P><center>');

    With Sessions do begin

        open;
        first;

        While not EOF do begin

            send (" +fieldbyname ('neumonic').asString+ ' <BR>');
            next;
        end; {while}

        close;
        send ('</center><P>');
        end; {with Sessions}

    sendhr;
    send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

    end; {with cgiEnvData1 do}

end; {FormCreate}

end.

```

Select Article or All: P_Artl.exe

unit Prartl;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type

TForm1 = class(TForm)
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 Query1: TQuery;
 DataSource1: TDataSource;
 Articles: TTable;
 Query1Paper_Number: TIntegerField;
 Query1Title: TStringField;
 ArticlesPaper_Number: TIntegerField;
 ArticlesTitle: TStringField;
 ArticlesInvited: TStringField;
 ArticlesAccepted: TStringField;
 ArticlesContactAuthorNumber: TIntegerField;
 ArticlesSession: TStringField;
 ArticlesKeyword1: TStringField;
 ArticlesKeyword2: TStringField;
 ArticlesKeyword3: TStringField;
 ArticlesAbstract: TMemoField;
 ArticlesContactOrder: TStringField;
 ArticlesFName2: TStringField;
 ArticlesLName2: TStringField;
 ArticlesInitial2: TStringField;
 ArticlesInstitution2: TStringField;
 ArticlesOrder2: TStringField;
 ArticlesFName3: TStringField;
 ArticlesLName3: TStringField;
 ArticlesInitial3: TStringField;
 ArticlesInstitution3: TStringField;
 ArticlesOrder3: TStringField;
 ArticlesFName4: TStringField;
 ArticlesLName4: TStringField;
 ArticlesInitial4: TStringField;
 ArticlesInstitution4: TStringField;
 ArticlesOrder4: TStringField;

```

ArticlesFName5: TStringField;
ArticlesLName5: TStringField;
ArticlesInitial5: TStringField;
ArticlesInstitution5: TStringField;
ArticlesOrder5: TStringField;
ArticlesFName6: TStringField;
ArticlesLName6: TStringField;
ArticlesInitial6: TStringField;
ArticlesInstitution6: TStringField;
ArticlesOrder6: TStringField;
procedure FormCreate(Sender: TObject);
procedure CGIDB1SendingHotField(currentRecord: TDataset;
  var s: OpenString);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin

  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
    sendPrologue;

    send ('<HTML><HEAD>');
    SendTitle('Asilomar Signals, Systems, and Computers');
    send ('</HEAD><BODY bgcolor=FFFFFF>');

    send('<center><H2>Asilomar Conference System Administration</H2></center>');
    sendhr;
  end;
end;

```

with Query1 do begin

```
prepare; { Optimizes query }
close;
sql.clear;
sql.add('Select * FROM Submissi');
sql.add( 'Order by Title ' );
open;
```

{Display the results of the Query}

```
send('<P>Choose the Article you would like to see below. The information will be
displayed to the screen ');
send('in a format that most Browsers will print directly. You may also <A HREF="..cgi-
win/P_artl2.exe">Print ');
send('Information for ALL Articles to a File</A>.<P><CENTER><BR>');
```

```
CGIDB1.drawtable; {display all titles}
send ('</center><P>');
end; { with Query1 }
```

```
sendhr;
send('<P><FORM ACTION="..cgi-win/Adminpwd.exe" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );
```

end; {with cgiEnvData1 do}

end; {FormCreate}

```
procedure TForm1.CGIDB1SendingHotField(currentRecord: TDataset;
var s: OpenString);
begin
s:='<A HREF="..cgi-win/P_Artl2.exe?'+
currentrecord.fieldbyname('Paper_Number').AsString + "'>'+
+ currentrecord.Fieldbyname('Title').AsString + '</A>';
```

end;

end.

unit Prart12;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DBTables, DB, Cgidb, Cgi;

type

TForm1 = class(TForm)
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 Query1: TQuery;
 DataSource1: TDataSource;
 Articles: TTable;
 Authors: TTable;
 AuthorsANumber: TIntegerField;
 AuthorsFName: TStringField;
 AuthorsLName: TStringField;
 AuthorsInitial: TStringField;
 AuthorsHonorific: TStringField;
 AuthorsInstitution: TStringField;
 AuthorsDepartment: TStringField;
 AuthorsMailstop: TStringField;
 AuthorsAddress_Street: TStringField;
 AuthorsAddress_City: TStringField;
 AuthorsAddress_State: TStringField;
 AuthorsAddress_Zip: TStringField;
 AuthorsCountry: TStringField;
 AuthorsPhone_LocalNumber: TStringField;
 AuthorsPhone_AreaCode: TStringField;
 AuthorsPhone_FaxNumber: TStringField;
 AuthorsEmail: TStringField;
 ArticlesPaper_Number: TIntegerField;
 ArticlesTitle: TStringField;
 ArticlesInvited: TStringField;
 ArticlesAccepted: TStringField;
 ArticlesContactAuthorNumber: TIntegerField;
 ArticlesSession: TStringField;
 ArticlesKeyword1: TStringField;
 ArticlesKeyword2: TStringField;
 ArticlesKeyword3: TStringField;
 ArticlesAbstract: TMemoField;


```

ArticlesContactOrder: TStringField;
ArticlesFName2: TStringField;
ArticlesLName2: TStringField;
ArticlesInitial2: TStringField;
ArticlesInstitution2: TStringField;
ArticlesOrder2: TStringField;
ArticlesFName3: TStringField;
ArticlesLName3: TStringField;
ArticlesInitial3: TStringField;
ArticlesInstitution3: TStringField;
ArticlesOrder3: TStringField;
ArticlesFName4: TStringField;
ArticlesLName4: TStringField;
ArticlesInitial4: TStringField;
ArticlesInstitution4: TStringField;
ArticlesOrder4: TStringField;
ArticlesFName5: TStringField;
ArticlesLName5: TStringField;
ArticlesInitial5: TStringField;
ArticlesInstitution5: TStringField;
ArticlesOrder5: TStringField;
ArticlesFName6: TStringField;
ArticlesLName6: TStringField;
ArticlesInitial6: TStringField;
ArticlesInstitution6: TStringField;
ArticlesOrder6: TStringField;
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

theabstract : TStringList;
theAuthors : TStringList;

i: integer;

dbContactOrderInt: integer;
dbOrder2int: integer;
dbOrder3int: integer;
dbOrder4int: integer;

```

dbOrder5int: integer;
dbOrder6int: integer;

dPaper_Number: string;

dbtitle: string;
dbinvited: string;
dbaccepted: string;
dbANumber: string;
dbSession: string;

dbkeyword1: string;
dbkeyword2: string;
dbkeyword3: string;
dbabstract: string;
dbcontactOrder: string;

dbfname2: string;
dblname2: string;
dbinitial2: string;
dbinstitution2: string;
dborder2: string;

dbfname3: string;
dblname3: string;
dbinitial3: string;
dbinstitution3: string;
dborder3: string;

dbfname4: string;
dblname4: string;
dbinitial4: string;
dbinstitution4: string;
dborder4: string;

dbfname5: string;
dblname5: string;
dbinitial5: string;
dbinstitution5: string;
dborder5: string;

dbfname6: string;
dblname6: string;
dbinitial6: string;
dbinstitution6: string;

dborder6: string;

dbFName : string;

dbLName : string;

dbInitial: string;

dbHonorific : string;

dbInstitution : string;

implementation

{ \$R *.DFM }

procedure TForm1.FormCreate(Sender: TObject);

begin

 with CGIEnvData1 do

 begin

 webSiteINIFilename:=paramstr(1);

 application.onException:=cgiErrorHandler;

 application.processMessages;

 createStdout;

 sendPrologue;

 { Get the Paper Number Selected }

 try

 dPaper_Number:= CGIQueryString^;

 finally

 { Send the html page }

 send ('<HTML><HEAD>');

 SendTitle('Asilomar Signals, Systems, and Computers System Admin: Print an Article');

 send ('</HEAD><BODY bgcolor=FFFFFF>');

 end; {try}

 {Determine if one Article or ALL Articles were requested}

 if dPaper_Number = cginotfound then begin

 send('<center><H2>Asilomar Conference System Administration</H2></center>');

 sendhdr;

 sendhdr ('3', 'Print to a file.');

```

    send('Please enter the name of the file into which you would like the Article Information
placed. ');
    send('The file extension is not required, the file will be given the extension .Art after it is
created. ');
    send('If an existing file name is used, the old file extension will be changed to .old and the
new ');
    send('file"s extension will be .Art');
    send('<P><FORM ACTION=" ../cgi-win/P_Artfil.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<INPUT TYPE="hidden" Name="Paper_Number" Value="" +dPaper_Number+ "">');
    send('<CENTER><B>File Name: </B><INPUT TYPE="text" Name="filename" Size="25"
Value="Title.Art"><BR>');
    send('<P><INPUT TYPE="submit" Value="Print the Articles to this
file"></CENTER></FORM><BR>');

```

```

    sendhr;
    send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');

```

end {Print ALL Articles}

else begin

{Print ONE Article}

{ Get the Paper Record from the SUBMISSION Table }
with Articles do begin

{ Move to proper record}

```

open;
first;
while fieldByName('Paper_Number').asString <> dPaper_Number do
    next;

```

{ Retrieve record }

```

dbtitle:= fieldByName ('title').asString;
dbinvited:= fieldByName ('invited').asString;
dbaccepted:= fieldByName ('accepted').asString;
dbANumber:= fieldByName ('ContactAuthorNumber').asString;
dbSession:= fieldByName ('Session').asString;
dbkeyword1:= fieldByName ('keyword1').asString;
dbkeyword2:= fieldByName ('keyword2').asString;
dbkeyword3:= fieldByName ('keyword3').asString;

```

```
CGIDB1.memoToStringList (fieldbyname ('Abstract'), theabstract);
```

```
dbcontactOrder:= fieldByName ('contactOrder').asstring;
```

```
dborder2:= fieldByName ('order2').asstring;  
dbfname2:= fieldByName ('fname2').asstring;  
dbinitial2:= fieldByName ('initial2').asstring;  
dblname2:= fieldByName ('lname2').asstring;  
dbinstitution2:= fieldByName ('institution2').asstring;
```

```
dborder3:= fieldByName ('order3').asstring;  
dbfname3:= fieldByName ('fname3').asstring;  
dbinitial3:= fieldByName ('initial3').asstring;  
dblname3:= fieldByName ('lname3').asstring;  
dbinstitution3:= fieldByName ('institution3').asstring;
```

```
dborder4:= fieldByName ('order4').asstring;  
dbfname4:= fieldByName ('fname4').asstring;  
dbinitial4:= fieldByName ('initial4').asstring;  
dblname4:= fieldByName ('lname4').asstring;  
dbinstitution4:= fieldByName ('institution4').asstring;
```

```
dborder5:= fieldByName ('order5').asstring;  
dbfname5:= fieldByName ('fname5').asstring;  
dbinitial5:= fieldByName ('initial5').asstring;  
dblname5:= fieldByName ('lname5').asstring;  
dbinstitution5:= fieldByName ('institution5').asstring;
```

```
dborder6:= fieldByName ('order6').asstring;  
dbfname6:= fieldByName ('fname6').asstring;  
dbinitial6:= fieldByName ('initial6').asstring;  
dblname6:= fieldByName ('lname6').asstring;  
dbinstitution6:= fieldByName ('institution6').asstring;  
end; {With Articles}
```

```
{ Get Author Name information from the AUTHOR Table }
```

```
with Authors do begin
```

```
  { Move to proper record }
```

```
  open;
```

```
  first;
```

```
  while fieldbyname('ANumber').asstring <> dbANumber do  
    next;
```

{ Retrieve Author Name information }

```
dbFName := fieldByName('FName').AsString;  
dbLName := fieldByName('LName').AsString;  
dbInitial := fieldByName('Initial').AsString;  
dbHonorific := fieldByName('Honorific').AsString;  
dbInstitution := fieldByName('Institution').AsString;  
close;  
end;
```

{Print Article}

```
sendhdr ('3', '<P><BR><P><CENTER>' + dbtitle + '</CENTER>');
```

```
send('<P><BR><B>Registration Number: </B>' + dPaper_Number + ");  
send ('<P><B>Keyword #1: </B>' + dbkeyword1 + ");  
if (dbKeyword2 <> 'None') then send ('<P><B>Keyword #2: </B>' + dbkeyword2 + ");  
if dbKeyword3 <> 'None' then send ('<P><B>Keyword #3: </B>' + dbkeyword3 + ");
```

{Determine Order of Authors }

```
theAuthors:= TStringList.create;
```

{Send the Authors}

```
send('<P><B>Authors:</B><p>');
```

```
if dbContactOrder = " then send(" + dbFName + ' ' + dbInitial + ' ' + dbLName + ', '  
+ dbInstitution + ")  
else begin
```

{initialize the stringlist}

```
for i:= 0 to 6 do  
  theAuthors.add("This string is not used");
```

```
dbContactOrderInt:= strtoint(dbContactOrder);  
theAuthors[dbContactOrderInt]:= " + dbFName + ' ' + dbInitial + ' ' + dbLName + ', '  
+ dbInstitution + ";
```

```
if dbOrder2 <> " then begin  
  dbOrder2Int:= strtoint(dbOrder2);  
  theAuthors[dbOrder2Int]:= " + dbFName2 + ' ' + dbInitial2 + ' ' + dbLName2 + ', '  
+ dbInstitution2 + ";  
end;
```

```
if dbOrder3 <> " then begin  
  dbOrder3Int:= strtoint(dbOrder3);
```

```

        theAuthors[dbOrder3Int]:= "+dbFName3+ ' '+dbInitial3+ ' '+dbLName3+ ', '
+dbInstitution3+ ";
        end;

        if dbOrder4 <> " then begin
            dbOrder4Int:= strtoint(dbOrder4);
            theAuthors[dbOrder4Int]:= "+dbFName4+ ' '+dbInitial4+ ' '+dbLName4+ ', '
+dbInstitution4+ ";
            end;

        if dbOrder5 <> " then begin
            dbOrder5Int:= strtoint(dbOrder5);
            theAuthors[dbOrder5Int]:= "+dbFName5+ ' '+dbInitial5+ ' '+dbLName5+ ', '
+dbInstitution5+ ";
            end;

        if dbOrder6 <> " then begin
            dbOrder6Int:= strtoint(dbOrder6);
            theAuthors[dbOrder6Int]:= "+dbFName6+ ' '+dbInitial6+ ' '+dbLName6+ ', '
+dbInstitution6+ ";
            end;

        for i:= 1 to 6 do
            if theAuthors[i] <> 'This string is not used' then send(" +theAuthors[i]+ '<BR>');

        end; {else}

        {Send the Abstract}
        send ('<P><BR><B>Abstract:  </B><br>');
        for i := 0 to theabstract.count - 1 do
            send( theabstract.strings[i] );

        theabstract.free; {release the memory held by 'theabstract'}

        end; {IF One Article}

        send ('</BODY></HTML>');
        closeStdout;
        closeApp( application ); { don't leave form around }

        end; {with CGIEnvData1}

end; {with FormCreate}

```

end.

Print Articles to file: **P_Artfile.exe**

unit Prartfil;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type

TForm1 = class(TForm)
 DataSource1: TDataSource;
 Articles: TTable;
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 ArticlesPaper_Number: TIntegerField;
 ArticlesTitle: TStringField;
 ArticlesInvited: TStringField;
 ArticlesAccepted: TStringField;
 ArticlesContactAuthorNumber: TIntegerField;
 ArticlesSession: TStringField;
 ArticlesKeyword1: TStringField;
 ArticlesKeyword2: TStringField;
 ArticlesKeyword3: TStringField;
 ArticlesAbstract: TMemoField;
 ArticlesContactOrder: TStringField;
 ArticlesFName2: TStringField;
 ArticlesLName2: TStringField;
 ArticlesInitial2: TStringField;
 ArticlesInstitution2: TStringField;
 ArticlesOrder2: TStringField;
 ArticlesFName3: TStringField;
 ArticlesLName3: TStringField;
 ArticlesInitial3: TStringField;
 ArticlesInstitution3: TStringField;
 ArticlesOrder3: TStringField;
 ArticlesFName4: TStringField;
 ArticlesLName4: TStringField;
 ArticlesInitial4: TStringField;
 ArticlesInstitution4: TStringField;
 ArticlesOrder4: TStringField;
 ArticlesFName5: TStringField;
 ArticlesLName5: TStringField;
 ArticlesInitial5: TStringField;

```

ArticlesInstitution5: TStringField;
ArticlesOrder5: TStringField;
ArticlesFName6: TStringField;
ArticlesLName6: TStringField;
ArticlesInitial6: TStringField;
ArticlesInstitution6: TStringField;
ArticlesOrder6: TStringField;
Authors: TTable;
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

```
var
```

```
Form1: TForm1;
```

```
MyFileHandle: THandle;
```

```
F: Textfile;
```

```
theabstract : TStringList;
```

```
theAuthors : TStringList;
```

```
i: integer;
```

```
dbContactOrderInt: integer;
```

```
dbOrder2int: integer;
```

```
dbOrder3int: integer;
```

```
dbOrder4int: integer;
```

```
dbOrder5int: integer;
```

```
dbOrder6int: integer;
```

```
dFileName, dFileNamein, dflag: string;
```

```
PathFile, OldFile: string;
```

```
dbPaper_Number: string;
```

```
dbtitle: string;
```

```
dbinvited: string;
```

```
dbaccepted: string;
```

```
dbANumber: string;
```

```
dbSession: string;
```

```
dbkeyword1: string;
```

```
dbkeyword2: string;
```

dbkeyword3: string;
dbabstract: string;
dbcontactOrder: string;

dbfname2: string;
dblname2: string;
dbinitial2: string;
dbinstitution2: string;
dborder2: string;

dbfname3: string;
dblname3: string;
dbinitial3: string;
dbinstitution3: string;
dborder3: string;

dbfname4: string;
dblname4: string;
dbinitial4: string;
dbinstitution4: string;
dborder4: string;

dbfname5: string;
dblname5: string;
dbinitial5: string;
dbinstitution5: string;
dborder5: string;

dbfname6: string;
dblname6: string;
dbinitial6: string;
dbinstitution6: string;
dborder6: string;

dbFName : string;
dbLName : string;
dbInitial: string;
dbHonorific : string;
dbInstitution : string;

implementation

{SR *.DFM}

```

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do
    begin

      webSiteINIFilename:=paramstr(1);
      application.onException:=cgiErrorHandler;
      application.processMessages;

      createStdout;
      sendPrologue;

      { Send the html page }
      send ('<HTML><HEAD>');
      SendTitle('Asilomar Signals, Systems, and Computers System Admin: Print an Article');
      send ('</HEAD><BODY bgcolor=FFFFFF>');
      send('<center><H2>Asilomar Conference System Administration</H2></center>');
      sendhr;

      { Get the FORM Data}
      dflag:= getsmallfield ('flag');
      dFileNamein:= getsmallfield ('filename');

      if dflag = '1' then begin

        {Print Articles}

        {Prepare File}
        try
          if ExtractFileExt(dFileNamein) = 'Art' then dFileName:= dFileNamein
            else dFileName:= ChangeFileExt("+dFileNamein+ ", '.Art');
          PathFile:= 'c:\website\htdocs\Download\' +dFileName+";
          if FileExists(PathFile) then begin
            {rename old file}
            OldFile := ChangeFileExt("+PathFile+ ", '.OLD');
            {open New file}
            fileopen (dFileName, 0);
            {send notice}
            send('There was an existing file named "' +PathFile+ '" which was renamed to "'
+OldFile+ "'');
            end
            else MyFileHandle := filecreate(PathFile);

```

finally

{Prepare File for Writing}

 AssignFile (F, PathFile);

 Append(F);

end; {try}

{Prepare Table}

Articles.open;

Articles.first;

While not Articles.EOF do begin

 with Articles do begin

{ Retrieve record }

 dbPaper_Number:= fieldName ('Paper_Number').asString;

 dbtitle:= fieldName ('title').asString;

 dbinvited:= fieldName ('invited').asString;

 dbaccepted:= fieldName ('accepted').asString;

 dbANumber:= fieldName ('ContactAuthorNumber').asString;

 dbSession:= fieldName ('Session').asString;

 dbkeyword1:= fieldName ('keyword1').asString;

 dbkeyword2:= fieldName ('keyword2').asString;

 dbkeyword3:= fieldName ('keyword3').asString;

 CGIDB1.memoToStringList (fieldname ('Abstract'), theabstract);

 dbcontactOrder:= fieldName ('contactOrder').asString;

 dborder2:= fieldName ('order2').asString;

 dbfname2:= fieldName ('fname2').asString;

 dbinitial2:= fieldName ('initial2').asString;

 dblname2:= fieldName ('lname2').asString;

 dbinstitution2:= fieldName ('institution2').asString;

 dborder3:= fieldName ('order3').asString;

 dbfname3:= fieldName ('fname3').asString;

 dbinitial3:= fieldName ('initial3').asString;

 dblname3:= fieldName ('lname3').asString;

 dbinstitution3:= fieldName ('institution3').asString;

 dborder4:= fieldName ('order4').asString;

 dbfname4:= fieldName ('fname4').asString;

```
dbinitial4:= fieldByName ('initial4').asString;
dblname4:= fieldByName ('lname4').asString;
dbinstitution4:= fieldByName ('institution4').asString;
```

```
dborder5:= fieldByName ('order5').asString;
dbfname5:= fieldByName ('fname5').asString;
dbinitial5:= fieldByName ('initial5').asString;
dblname5:= fieldByName ('lname5').asString;
dbinstitution5:= fieldByName ('institution5').asString;
```

```
dborder6:= fieldByName ('order6').asString;
dbfname6:= fieldByName ('fname6').asString;
dbinitial6:= fieldByName ('initial6').asString;
dblname6:= fieldByName ('lname6').asString;
dbinstitution6:= fieldByName ('institution6').asString;
end; {With Articles}
```

{ Get Author Name information from the AUTHOR Table }

with Authors do begin

{ Move to proper record }

```
open;
first;
while fieldByName('ANumber').asString <> dbANumber do
    next;
```

{ Retrieve Author Name information }

```
dbFName := fieldByName('FName').AsString;
dbLName := fieldByName('LName').AsString;
dbInitial := fieldByName('Initial').AsString;
dbHonorific := fieldByName('Honorific').AsString;
dbInstitution := fieldByName('Institution').AsString;
close;
end; {with Authors}
```

{Print Article data to file}

```
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ' + dbtitle + ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
```

```

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, 'Registration Number: ' + dbPaper_Number + ");
Writeln(F, ' ');
Writeln(F, 'Keyword #1: ' + dbkeyword1 + ");
Writeln(F, ' ');
if dbKeyword2 <> 'None' then Writeln(F, 'Keyword #2: ' + dbkeyword2 + ");
Writeln(F, ' ');
if dbKeyword3 <> 'None' then Writeln(F, 'Keyword #3: ' + dbkeyword3 + ");
Writeln(F, ' ');
Writeln(F, ' ');

```

{Determine Order of Authors }

```
theAuthors:= TStringList.create;
```

{Send the Authors}

```

Writeln(F, 'Authors:');
Writeln(F, ' ');

```

```

if dbContactOrder = " then Writeln (F, " + dbFName+ ' ' + dbInitial+ ' ' + dbLName+ ', '
+ dbInstitution+ ")
else begin

```

{initialize the stringlist}

```
for i:= 0 to 6 do
```

```
theAuthors.add("This string is not used");
```

```
dbContactOrderInt:= strtoint(dbContactOrder);
```

```
theAuthors[dbContactOrderInt]:= " + dbFName+ ' ' + dbInitial+ ' ' + dbLName+ ', '
+ dbInstitution+ ";
```

```
if dbOrder2 <> " then begin
```

```
dbOrder2Int:= strtoint(dbOrder2);
```

```
theAuthors[dbOrder2Int]:= " + dbFName2+ ' ' + dbInitial2+ ' ' + dbLName2+ ', '
+ dbInstitution2+ ";
end;
```

```
if dbOrder3 <> " then begin
```

```
dbOrder3Int:= strtoint(dbOrder3);
```

```
theAuthors[dbOrder3Int]:= " + dbFName3+ ' ' + dbInitial3+ ' ' + dbLName3+ ', '
+ dbInstitution3+ ";
end;
```

```
if dbOrder4 <> " then begin
```

```
dbOrder4Int:= strtoint(dbOrder4);
```

```

        theAuthors[dbOrder4Int]:= "+dbFName4+ ' ' +dbInitial4+ ' ' +dbLName4+ ', '
+dbInstitution4+ ";
        end;

        if dbOrder5 <> " then begin
            dbOrder5Int:= strtoint(dbOrder5);
            theAuthors[dbOrder5Int]:= "+dbFName5+ ' ' +dbInitial5+ ' ' +dbLName5+ ', '
+dbInstitution5+ ";
            end;

        if dbOrder6 <> " then begin
            dbOrder6Int:= strtoint(dbOrder6);
            theAuthors[dbOrder6Int]:= "+dbFName6+ ' ' +dbInitial6+ ' ' +dbLName6+ ', '
+dbInstitution6+ ";
            end;

        for i:= 1 to 6 do
            if theAuthors[i] <> 'This string is not used' then Writeln(F, " +theAuthors[i]+ '<BR>');

        end; {else}

{Send the Abstract}
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, 'Abstract: ');
Writeln(F, ' ');
for i := 0 to theabstract.count - 1 do
    Writeln(F, " +theabstract.strings[i]+ ");

theabstract.free; {release the memory held by 'theabstract'}

{Fill out the rest of the page}
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');

```



```

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, '----- ');

```

```

Articles.next;
end; {while}

```

```

closeFile(F);

```

```

end {If flag}

```

```

else begin {bad flag}

```

```

    send('<center><H1>Asilomar Conference on Signals, Systems, &
Computers</H1></center>');
    sendhdr;
    send('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
    send('<P>Please ensure that you are authorized to access this information. ');
    send('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to
login again.</A>');
    send('<P>If you are not authorized to access this information, please note that a log is');
    send('maintained that includes your URL. You may be subject to civil and/or criminal ');
    send('prosecution for your actions.<BR>');
    end; {if dflag else}

```

```

sendhdr('3', 'The file has been generated!');

```

```

send('The file <B>' + dFileName + '</B> is ready to be Downloaded. ');
send('You may <A HREF=" ../download/' + dFileName + '>Download the file by FTP
now!</A><BR>');
sendhdr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');

```

```
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin  
Options page"></CENTER></FORM>');  
    send('</BODY></HTML>');  
    closeStdout;  
    closeApp( application ); { don't leave form around }  
  
    end; {with CGIEnvData1}  
  
end; {with FormCreate}  
  
end.
```

Instructions / Options: **P_Accept.exe**

```
unit Praccept;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  dflag: string;
  daction: string;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
    sendPrologue;

    { Get fields from search page }
```

```

dflag:= getsmallfield ('flag');
daction:= getsmallfield ('action');

send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');

send('<A NAME=#top"><center><H2>Asilomar Conference System
Administration</H2></center></A>');
sendhr;

send('<P>After review, Articles will have been accepted or rejected. This information is
entered ');
send('into the database in the "Database Functions" portion of the System Admin Options
Page. ');
send('If this information has not yet been entered, please go to the <A
HREF="#bottom">Bottom of this Page</A> ');
send('for a link to the System Admin Options Page.');
```

send('<P>There are three options in printing out an Article's accept/reject letter:');

```

send('<P><A HREF="#one to screen">Print a letter for ONE specific Article to the
SCREEN.</A><BR>');
send('<P><A HREF="#one to file">Print a letter for ONE specific Article to a
FILE.</A><BR>');
send('<P><A HREF="#all to file">Print a letter for EVERY Article to a
FILE.</A><P><BR><P>');
```

{print to screen option}

```

send ('<A NAME="one to screen"><HR></A><P>');
sendhdr ('3', 'Printing a letter for ONE specific Article to the screen');
send ('Selecting this option will display the letter as an HTML page that can be ');
send ('printed directly from most Browsers.<BR>');
```

```

send('<P><FORM ACTION=" ../cgi-win/P_accep2.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><BR><CENTER><INPUT TYPE="submit" Value="Print ONE
Article"></CENTER></FORM>');
```

```

send('<P><BR><A HREF=#top">Go back to the top of the page for more
options</A><P><BR>');
```

{print to file option}

```

send ('<A NAME="one to file"><HR></A><P>');
sendhdr ('3', 'Printing a letter for ONE specific Article to a file');
```

send ('Selecting this option will print a letter to a file. You will provide a Name for the file and then be able ');

send ('to FTP the file to your location. The file will contain ASCII text that can be manipulated by any ');

send ('word-processor.
');

send('<P><CENTER><FORM ACTION=" ../cgi-win/P_accep2.exe" METHOD="POST">');

send('<INPUT TYPE="hidden" Name="flag" Value="1">');

send('<P><INPUT TYPE="submit" Value="Print ONE Article"></CENTER></FORM>');

send('<P>
Go back to the top of the page for more options<P>
');

{print ALL to file option}

send ('<HR><P>');

sendhdr ('3', 'Printing a letter for EVERY Article to a file');

send ('Selecting this option will print a letter for EVERY Article in the database to a single file. ');

send ('Enter the file Name below. The file will contain ASCII text and can be retrieved by FTP.
');

send ('The file extension is not required as it will be .Let after the operation is complete. ');

send ('If an existing file name is used, the old file extension will be changed to .old and the new ');

send ('file"s extension will be .Let');

send ('<P><FORM ACTION=" ../cgi-win/P_accAll.exe" METHOD="POST">');

send ('<INPUT TYPE="hidden" Name="flag" Value="1">');

send ('<center>File Name: <INPUT TYPE="text" Name="filename" Size="25" Value="AllAuths.Let">
');

send ('<P><INPUT TYPE="submit" Value="Print the letters to this file"></CENTER></FORM>');

send ('<P>
Go back to the top of the page for more options<P>
');

{send Footer}

sendhr;

send ('<P>
');

send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');

send('<INPUT TYPE="hidden" Name="flag" Value="1">');

send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin Options page"></CENTER></FORM>');

send('<P>
<P>
<P>
');

send ('</BODY></HTML>');

closeStdout;

```
closeApp( application );  
  
end; {with cgiEnvData1 do}  
  
end; {FormCreate}  
  
end.
```

Print all to a file: **P_AccAll.exe**

unit Praccall;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DB, DBTables, Cgidb, Cgi;

type

TForm1 = class(TForm)
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 DataSource1: TDataSource;
 Articles: TTable;
 Authors: TTable;
 procedure FormCreate(Sender: TObject);
private
 { Private declarations }
public
 { Public declarations }
end;

var

Form1: TForm1;

dFileName, dFileNamein: string;
dflag: string;
dPaper_Number: string;

dbANumber, dbTitle: string;

dbFName : string;
dbLName : string;
dbInitial : string;
dbHonorific : string;
dbInstitution : string;
dbDepartment : string;
dbMailstop : string;
dbAddress_Street : string;
dbAddress_City : string;
dbAddress_State : string;
dbAddress_Zip : string;
dbCountry : string;

```
dbPhone_LocalNumber : string;  
dbPhone_FaxNumber : string;  
dbEmail : string;
```

```
PathFile, OldFile: string;  
MyFileHandle: THandle;  
F: TextFile;
```

implementation

```
{ $R *.DFM }
```

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
  with CGIEnvData1 do begin
```

```
    webSiteINIFilename:=paramstr(1);  
    application.onException:=cgiErrorHandler;  
    application.processMessages;
```

```
    createStdout;  
    sendPrologue;
```

```
    { Get fields from search page }  
    dflag:= getsmallfield ('flag');  
    dPaper_Number:= getsmallfield ('Paper_Number');  
    dFilenamein:= getsmallfield ('filename');
```

```
    {HTML Header}
```

```
    send ('<HTML><HEAD>');  
    SendTitle('Asilomar Signals, Systems, and Computers');  
    send ('</HEAD><BODY bgcolor=FFFFFF>');  
    send('<center><H2>Asilomar Conference System Administration</H2></center>');  
    sendhr;
```

```
{Check the flag}  
if dflag = '1' then begin
```

```
  {Prepare File}
```

```
  try  
    if ExtractFileExt(dFilenamein) = '.let' then dFileName:= dFilenamein  
    else dFileName:= ChangeFileExt("+dFilenamein+", '.let');  
    PathFile:= 'c:\website\Download\' + dFileName+";
```



```

if FileExists(PathFile) then begin
    {rename old file}
    OldFile := ChangeFileExt("+PathFile+ ", '.OLD');
    {open New file}
    fileopen (dFileName, 0);
    {send notice}
    send('There was an existing file named "' + PathFile + '" which was renamed to "'
+ OldFile + '"');
    end
else MyFileHandle := filecreate(PathFile);

```

```

finally
    {Prepare File for Writing}
    AssignFile (F, PathFile);
    Append(F);

```

```

end; {try}

```

```

{Prepare Table}
Articles.open;
Articles.first;

```

```

While not Articles.EOF do begin

```

```

    {Retrieve Data}
    dbANumber := Articles.fieldbyname ('ContactAuthorNumber').asString;
    dbTitle := Articles.fieldbyname ('Title').asString;

```

```

    {Get Contact Author Data}
    With Authors do begin

```

```

        {Move to Record}
        open;
        first;
        while fieldbyname('ANumber').asString <> dbANumber do
            next;

```

```

        dbFName := fieldByName('FName').AsString;
        dbLName := fieldByName('LName').AsString;
        dbInitial := fieldByName('Initial').AsString;
        dbHonorific := fieldByName('Honorific').AsString;
        dbInstitution := fieldByName('Institution').AsString;
        dbDepartment := fieldByName('Department').AsString;
        dbMailstop := fieldByName('Mailstop').AsString;
        dbAddress_Street := fieldByName('Address_Street').AsString;

```

```

dbAddress_City := fieldByName('Address_City').AsString;
dbAddress_State := fieldByName('Address_State').AsString;
dbAddress_Zip := fieldByName('Address_Zip').AsString;
dbCountry := fieldByName('Country').AsString;
dbPhone_LocalNumber := fieldByName('Phone_LocalNumber').AsString;
dbPhone_FaxNumber := fieldByName('Phone_FaxNumber').AsString;
dbEmail := fieldByName('Email').AsString;
end; {with Authors}

{Print Article data to file}
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, '                                     '+webdate(now)+' ');

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ''+dbHonorific+ ''+dbFName+ ''+dbInitial+ ''+dbLName+ '<BR>');
Writeln(F, ''+dbAddress_Street+ ');
if dbCountry = 'USA' then Writeln(F, ''+dbAddress_City+ ', '+dbAddress_State+ ''
+dbAddress_Zip+ ")
    else Writeln(F, ''+dbAddress_City+ ', '+dbCountry+ '<BR>');
Writeln(F, ' ');
Writeln(F, ' ');

Writeln(F, 'Dear '+dbHonorific+ ''+dbLName+ ');
Writeln(F, ' ');
Writeln(F, 'Thank you for your interest in submitting a Paper for presentation at the Thirtieth
Annual ');
Writeln(F, 'Asilomar Conference on Signals, Systems, & Computers.');
```

if Articles.fieldbyname ('accepted').asstring = 'Y' then begin {Print accept}

```

    Writeln(F, 'We are pleased to inform you that your article, "'+dbTitle+ '" has been
selected ');
    Writeln(F, 'for the conference. Please ensure you have a copy of the final article to give
to the ');
    Writeln(F, 'conference for publication in the Proceedings.<BR>');
    Writeln(F, ' ');
    Writeln(F, 'Thank you again, we"ll see you at the conference.');
```

Writeln(F, ' ');

Writeln(F, ' ');

Writeln(F, ' ');

Writeln(F, ' ');

Writeln(F, ' ');

Writeln(F, ' ');

```

Writeln(F, ' ');
Writeln(F, '                Technical Program Chair');

```

{Fill out the rest of the page}

```

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, '----- ');
end {Print accept}

```

else if Articles.fieldbyname ('accepted').asString = 'N' then begin {Print Reject}

Writeln(F, 'We are sorry to inform you that your article, "' + dbTitle + '" was not selected
for this year"s ');

```

Writeln(F, 'conference.<BR>');
Writeln(F, ' ');
Writeln(F, 'Thank you again for your interest. We hope to see you at the conference. ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, '                Technical Program Chair');

```

{Fill out the rest of the page}

```

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');

```

```

        Writeln(F, ' ');
        Writeln(F, ' ');
        Writeln(F, ' ');
        Writeln(F, ' ');
        Writeln(F, ' ----- ');
    end {else Print Reject}

else Begin {send no accept/reject data}

    send('There is no Accept/Reject information on the article "' + dbTitle + '"');
    end; {bad accept/reject data}

Articles.next;
end; {while}

end {If flag}

else begin {bad flag}
    send('<center><H1>Asilomar Conference on Signals, Systems, &
Computers</H1></center>');
    sendhr;
    send('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
    send ('<P>Please ensure that you are authorized to access this information. ');
    send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to
login again.</A>');
    send ('<P>If you are not authorized to access this information, please note that a log is');
    send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
    send ('prosecution for your actions.<BR>');
    end; {if dflag else}

    sendhdr('3','The file has been generated!');

    send('The file <B>' + dFileName + '</B> is ready to be Downloaded. ');
    send('You may <A HREF=" ../download/" + dFileName + ">Download the file by FTP
now!</A><BR>');
    sendhr;
    send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

```

end; {with cgiEnvData1 do}

end; {FormCreate}

end.

Select Article: **P_Accep2.exe**

unit Praccep2;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, Cgldb, Cgi, DBTables, DB;

type

TForm1 = class(TForm)
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 DataSource1: TDataSource;
 Articles: TTable;
 Query1: TQuery;
 Query1Title: TStringField;
 Query1Paper_Number: TIntegerField;
 procedure FormCreate(Sender: TObject);
 procedure CGIDB1SendingHotField(currentRecord: TDataset;
 var s: OpenString);
private
 { Private declarations }
public
 { Public declarations }
end;

var

Form1: TForm1;

implementation

{ \$R *.DFM }

procedure TForm1.FormCreate(Sender: TObject);
begin

with CGIEnvData1 do begin

webSiteINIFilename:=paramstr(1);
 application.onException:=cgiErrorHandler;
 application.processMessages;

```

createStdout;
sendPrologue;

send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');

send('<center><H2>Asilomar Conference System Administration</H2></center>');
sendhr;

send('<P>Click on the Title of the article for which the letter should be created.<BR>');

with Query1 do begin

    prepare;    { Optimizes query }
    close;
    sql.clear;
    sql.add('Select * FROM Submissi');
    sql.add( 'Order by Title ' );
    open;
    send('<center><P>');
    CGIDB1.drawtable; {display all titles}
    send ('</center>');
    end; { with Query1}

sendhr;
send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
send('<INPUT TYPE="hidden" Name="flag" Value="1">');
send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
send ('</BODY></HTML>');
closeStdout;
closeApp( application );

end; {with cgiEnvData1 do}

end; {FormCreate}

procedure TForm1.CGIDB1SendingHotField(currentRecord: TDataset;
var s: OpenString);
begin
    s:='<A HREF=" ../cgi-win/P_Accep3.exe?"'+
currentrecord.fieldByName('Paper_Number').AsString + "'>'
+ currentrecord.FieldByName('Title').AsString + '</A>';

```

end;

end.

Choose screen or file: **P_Accep3.exe**

```
unit Praccep3;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DBTables, DB;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Articles: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    Authors: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  dPaper_Number: string;

  dTitle: string;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do begin

    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
```

```

sendPrologue;

{Get Title of Article from search page }
dPaper_Number:= cgiQueryString^;

send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');
send('<center><H2>Asilomar Conference on Signals, Systems, &
Computers</H2></center>');
sendhr;

{Retrieve the Title}
with Articles do begin

    {Prepare the Table}
    open;
    first;

    {Move to proper record}
    While fieldbyname('Paper_Number').asstring <> dPaper_Number do
        next;

    {Retrieve data}
    dTitle:= fieldbyname('Title').asstring;
    end; {With Articles}

    {Send the Options}
    send('The accept/reject letter for the Article "' + dTitle + '" may be printed to the screen or to a
file. ');

    {Send the "To the screen option"}
    sendhr;
    sendhdr ('3', 'Print to the screen. ');
    send('<P><FORM ACTION="./cgi-win/P_accscr.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<INPUT TYPE="hidden" Name="Paper_Number" Value="' + dPaper_Number + '">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Print the Article Accept/Reject letter
to the Screen"></CENTER></FORM>');

    {Send the "To a file option"}
    sendhr;
    sendhdr ('3', 'Print to a file. ');
    send('Please enter the name of the file into which you would like the letter placed. ');
    send('The file extension is not required as it will be .Let after the operation is complete. ');

```

```

    send('If an existing file name is used, the old file extension will be changed to .old and the
new ');
    send('file's extension will be .Let');
    send('<P><FORM ACTION=" ../cgi-win/P_accfil.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<INPUT TYPE="hidden" Name="Paper_Number" Value="" +dPaper_Number+ ">');
    send('<CENTER><B>File Name: </B><INPUT TYPE="text" Name="filename" Size="25"
Value="Authors.Let"><BR>');
    send('<P><INPUT TYPE="submit" Value="Print the Article Accept/Reject letter to this
File"></CENTER></FORM>');

```

{Send Footer}

```

    sendhr;
    send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    send('</BODY></HTML>');
    closeStdout;
    closeApp( application );

```

```

    end; {with cgiEnvData1 do}

```

```

end; {FormCreate}

```

```

end.

```

Print to screen: P_AccScr.exe

```
unit Praccscr;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Cgldb, Cgi, DBTables, DB;

type
  TForm1 = class(TForm)
    DataSource1: TDataSource;
    Articles: TTable;
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    Authors: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
i:integer;
dflag: string;
dPaper_Number: string;
dbANumber, dbTitle: string;
dbFName, dbLName, dbInitial, dbHonorific: string;
dbInstitution: string;
dbDepartment: string;
dbMailstop: string;
dbAddress_Street: string;
dbAddress_City: string;
dbAddress_State: string;
dbAddress_Zip: string;
dbCountry: string;
dbPhone_LocalNumber: string;
dbPhone_FaxNumber: string;
dbEmail: string;

dDate: string;
```

```

function GetDate (s:string) : string;

implementation

{$R *.DFM}

{*****}
function GetDate (s:string) : string;

const
    BlanksAllowed = 3;

var
    i, blanks:integer;
    temp: string;

begin
    i:= -1;    {variable to step through strings}
    blanks := 0; {variable to count the blank spaces in the string}

    repeat

        i:= i + 1;
        if s[i] = ' ' then blanks := blanks + 1; {increment blank count}
        temp[i]:= s[i];

    until (blanks = blanksAllowed) or (i=12);

    GetDate := temp;

end;
{*****}

procedure TForm1.FormCreate(Sender: TObject);
begin
    with CGIEnvData1 do begin

        webSiteINIFilename:=paramstr(1);
        application.onException:=cgiErrorHandler;
        application.processMessages;

        createStdout;
        sendPrologue;
    end;
end;

```

```

send ('<HTML><HEAD>');
SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');

```

{Get fields from search page }

```

dflag:= getsmallfield ('flag');
dPaper_Number:= getsmallfield ('Paper_Number');

```

{Get data from the Record in the Articles Table}

with Articles do begin

{Move to Record}

```

open;
first;
while fieldbyname('Paper_Number').asString <> dPaper_Number do
    next;

```

{Retrieve Data}

```

dbANumber:= fieldbyname ('ContactAuthorNumber').asString;
dbTitle:= fieldbyname ('Title').asString;

```

{Get Contact Author Data}

With Authors do begin

{Move to Record}

```

open;
first;
while fieldbyname('ANumber').asString <> dbANumber do
    next;

```

```

dbFName := fieldByName('FName').AsString;
dbLName := fieldByName('LName').AsString;
dbInitial := fieldByName('Initial').AsString;
dbHonorific := fieldByName('Honorific').AsString;
dbInstitution := fieldByName('Institution').AsString;
dbDepartment := fieldByName('Department').AsString;
dbMailstop := fieldByName('Mailstop').AsString;
dbAddress_Street := fieldByName('Address_Street').AsString;
dbAddress_City := fieldByName('Address_City').AsString;
dbAddress_State := fieldByName('Address_State').AsString;
dbAddress_Zip := fieldByName('Address_Zip').AsString;
dbCountry := fieldByName('Country').AsString;
dbPhone_LocalNumber := fieldByName('Phone_LocalNumber').AsString;
dbPhone_FaxNumber := fieldByName('Phone_FaxNumber').AsString;
dbEmail := fieldByName('Email').AsString;

```

end; {with Authors}

{Print out the letter}

if fieldbyname ('accepted').asstring = 'Y' then begin {Print accept}

```
dDate:= GetDate (webdate(now));
send('<P><BR><P ALIGN=right>' + dDate + " );
send('<P>' +dbHonorific+ '' +dbFName+ '' +dbInitial+ '' +dbLName+ '<BR>');
send(" +dbAddress_Street+ '<BR>');
if dbCountry = 'USA' then send(" +dbAddress_City+ ', ' +dbAddress_State+ ''
+dbAddress_Zip+ ")
    else send(" +dbAddress_City+ ', ' +dbCountry+ '<BR>');

send('<P>Dear ' +dbHonorific+ '' +dbLName+ ',<BR>');

send('Thank you for your interest in submitting a Paper for presentation at the Thirtieth
Annual ');
send('Asilomar Conference on Signals, Systems, & Computers.');
```

send('<P>We are pleased to inform you that your article, "' +dbTitle+ '" has been
selected ');
send('for the conference. Please ensure you have a copy of the final article to give to the
send('conference for publication in the Proceedings.
');

send('<P>Thank you again, we'll see you at the conference.');

send('<P>
<P>
<center>Technical Program Chair</center>');

end {Print accept}

else if fieldbyname ('accepted').asstring = 'N' then begin {Print Reject}

```
dDate:= GetDate (webdate(now));
send('<P><BR><P ALIGN=right>' + dDate + " );
send('<P>' +dbHonorific+ '' +dbFName+ '' +dbInitial+ '' +dbLName+ '<BR>');
send(" +dbAddress_Street+ '<BR>');
if dbCountry = 'USA' then send(" +dbAddress_City+ ', ' +dbAddress_State+ ''
+dbAddress_Zip+ ")
    else send(" +dbAddress_City+ ', ' +dbCountry+ '<BR>');

send('<P>Dear ' +dbHonorific+ '' +dbLName+ ',<BR>');

send('Thank you for your interest in submitting a Paper for presentation at the Thirtieth
Annual ');
send('Asilomar Conference on Signals, Systems, & Computers.');
```

```

    send('<P>We are sorry to inform you that your article, "' + dbTitle + '" was not selected
for this year"s ');
    send('conference.<BR>');

```

```

    send('<P>Thank you again for your interest. We hope to see you at the conference.');
```

```

    send('<P><BR><P><BR><center>Technical Program Chair</center>');
    end {else Print Reject}

```

```

else Begin {send no accept/reject data}

```

```

    {HTML Header}

```

```

    send('<center><H2>Asilomar Conference System Administration</H2></center>');
    sendhr;

```

```

    sendhdr('2', 'There is no Accept/Reject information on this article');
    send('<P><FORM ACTION=" ../cgi-win/P_Accep2.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Print a letter for another
article"></CENTER></FORM>');

```

```

    { HTML Footer}

```

```

    sendhr;
    send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    end; {bad accept/reject data}

```

```

end; {with Articles}

```

```

send ('</BODY></HTML>');
closeStdout;
closeApp( application );

```

```

end; {with cgiEnvData1 do}
end; {FormCreate}
end.

```


Print to file: **P_AccFil.exe**

unit Praccfil;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, Cgldb, Cgi, DBTables, DB;

type

TForm1 = class(TForm)
 DataSource1: TDataSource;
 Articles: TTable;
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 Authors: TTable;
 procedure FormCreate(Sender: TObject);
private
 { Private declarations }
public
 { Public declarations }
end;

var

Form1: TForm1;

dPath: string;
dFileName: string;
dflag: string;
dPaper_Number: string;

dbANumber, dbTitle: string;

dbFName : string;
dbLName : string;
dbInitial : string;
dbHonorific : string;
dbInstitution : string;
dbDepartment : string;
dbMailstop : string;
dbAddress_Street : string;
dbAddress_City : string;
dbAddress_State : string;
dbAddress_Zip : string;

```
dbCountry : string;  
dbPhone_LocalNumber : string;  
dbPhone_FaxNumber : string;  
dbEmail : string;
```

```
PathFile, dFileNamein, OldFile: string;  
MyFileHandle: THandle;  
F: TextFile;
```

implementation

```
{ $R *.DFM }
```

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
  with CGIEnvData1 do begin
```

```
    webSiteINIFilename:=paramstr(1);  
    application.onException:=cgiErrorHandler;  
    application.processMessages;
```

```
    createStdout;  
    sendPrologue;
```

```
    { Get fields from search page }  
    dflag:= getsmallfield ('flag');  
    dFileNamein:= getsmallfield ('filename');
```

```
    {HTML Header}  
    send ('<HTML><HEAD>');  
    SendTitle('Asilomar Signals, Systems, and Computers');  
    send ('</HEAD><BODY bgcolor=FFFFFF>');  
    send('<center><H2>Asilomar Conference System Administration</H2></center>');  
    sendhr;
```

```
{Check the flag}  
if dflag = '1' then begin
```

```
  {Prepare File}  
  try  
    if ExtractFileExt(dFileNamein) = 'Let' then dFileName:= dFileNamein  
    else dFileName:= ChangeFileExt("+dFileNamein+ ", 'Let');  
    PathFile:= 'c:\website\Download\' +dFileName+";
```

```

if FileExists(PathFile) then begin
    {rename old file}
    OldFile := ChangeFileExt("+PathFile+ ", '.OLD');
    {open New file}
    fileopen (dFileName, 0);
    {send notice}
    send('There was an existing file named "' + PathFile + '" which was renamed to "'
+ OldFile + '".');
    end
    else MyFileHandle := filecreate(PathFile);

```

```

finally
    {Prepare File for writing}
    AssignFile (F, PathFile);
    Append(F);

```

```

end; {try}

```

```

{Print every Article}
with Articles do begin

```

```

    {Prepare Table}
    open;
    first;

```

```

    {Move to Record}
    while fieldbyname('Paper_Number').asString <> dPaper_Number do
        next;

```

```

    {Retrieve Data}
    dbANumber:= fieldbyname ('ContactAuthorNumber').asString;
    dbTitle:= fieldbyname ('Title').asString;
    end; {with Articles}

```

```

    {Get Contact Author Data}
    With Authors do begin

```

```

        {Move to Record}
        open;
        first;
        while fieldbyname('ANumber').asString <> dbANumber do
            next;

```

```

        dbFName := fieldByName('FName').AsString;
        dbLName := fieldByName('LName').AsString;

```

```

dbInitial := fieldByName('Initial').AsString;
dbHonorific := fieldByName('Honorific').AsString;
dbInstitution := fieldByName('Institution').AsString;
dbDepartment := fieldByName('Department').AsString;
dbMailstop := fieldByName('Mailstop').AsString;
dbAddress_Street := fieldByName('Address_Street').AsString;
dbAddress_City := fieldByName('Address_City').AsString;
dbAddress_State := fieldByName('Address_State').AsString;
dbAddress_Zip := fieldByName('Address_Zip').AsString;
dbCountry := fieldByName('Country').AsString;
dbPhone_LocalNumber := fieldByName('Phone_LocalNumber').AsString;
dbPhone_FaxNumber := fieldByName('Phone_FaxNumber').AsString;
dbEmail := fieldByName('Email').AsString;
end; {with Authors}

```

{Print Article data to file}

```

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, '                                     '+webdate(now)+' ');

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, '' +dbHonorific+ '' +dbFName+ '' +dbInitial+ '' +dbLName+ '<BR>');
Writeln(F, '' +dbAddress_Street+ ');
if dbCountry = 'USA' then Writeln(F, '' +dbAddress_City+ ', ' +dbAddress_State+ ''
+dbAddress_Zip+ ')
    else Writeln(F, '' +dbAddress_City+ ', ' +dbCountry+ '<BR>');
Writeln(F, ' ');
Writeln(F, ' ');

Writeln(F, 'Dear ' +dbHonorific+ '' +dbLName+ ',');
Writeln(F, ' ');
Writeln(F, 'Thank you for your interest in submitting a Paper for presentation at the Thirtieth
Annual ');
Writeln(F, 'Asilomar Conference on Signals, Systems, & Computers.');
```

if Articles.fieldbyname ('accepted').asString = 'Y' then begin {Print accept}

```

    Writeln(F, 'We are pleased to inform you that your article, "' +dbTitle+ '" has been
selected ');
    Writeln(F, 'for the conference. Please ensure you have a copy of the final article to give
to the ');
    Writeln(F, 'conference for publication in the Proceedings.<BR>');
    Writeln(F, ' ');

```

```

Writeln(F, 'Thank you again, we'll see you at the conference. ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, '
                                Technical Program Chair');

```

{Fill out the rest of the page}

```

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, '----- ');
end {Print accept}

```

else if Articles.fieldbyname ('accepted').asString = 'N' then begin {Print Reject}

Writeln(F, 'We are sorry to inform you that your article, "' + dbTitle + '" was not selected for this year's ');

```

Writeln(F, 'conference.<BR>');
Writeln(F, ' ');
Writeln(F, 'Thank you again for your interest. We hope to see you at the conference. ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, '
                                Technical Program Chair');

```

{Fill out the rest of the page}

```

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');

```

```

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, ' ----- ');
end {else Print Reject}

```

```

else Begin {send no accept/reject data}

```

```

    send('There is no Accept/Reject information on the article "' + dbTitle + '");
    end; {bad accept/reject data}

```

```

end {If flag}

```

```

else begin {bad flag}

```

```

    send('<center><H2>Your password was <strong>not accepted!!</strong></H2></center>');
    send ('<P>Please ensure that you are authorized to access this information. ');
    send ('<P>If you made an error, then please <A HREF=" ../prototype/admin20.htm">try to
login again.</A>');
    send ('<P>If you are not authorized to access this information, please note that a log is');
    send ('maintained that includes your URL. You may be subject to civil and/or criminal ');
    send ('prosecution for your actions.<BR>');
    end; {if dflag else}

```

```

    sendhdr('3','The file has been generated!');

```

```

    send('The file <B>' + dFileName + '</B> is ready to be Downloaded. ');
    send('You may <A HREF=" ../download/' + dFileName + '">Download the file by FTP
now!</A><BR>');
    sendhr;
    send('<P><FORM ACTION=" ../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

```

```
end, {with cgiEnvData1 do}
end; {FormCreate}
end.
```

Print Initiation: P_AbsCat.exe

```
unit Prabscat;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    CGIDB1: TCGIDB;
    DataSource1: TDataSource;
    Articles: TTable;
    Authors: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  with CGIEnvData1 do begin
    webSiteINIFilename:=paramstr(1);
    application.onException:=cgiErrorHandler;
    application.processMessages;

    createStdout;
    sendPrologue;

    send('<HTML><HEAD>');
```



```

SendTitle('Asilomar Signals, Systems, and Computers');
send ('</HEAD><BODY bgcolor=FFFFFF>');

end;

with cgiEnvData1 do begin

    send('<center><H2>Asilomar Conference System Administration</H2></center>');
    sendhr;

    send('<P>This function Prints the Abstract Catalog, organized by sessions to the file named
below. ');
    send('Changes to File Name can be made below. If the file does not exist, it will be
created. ');

    send('<P>Running this routine may take a long time.<BR>');

    send('<center><FORM ACTION="../../cgi-win/P_absca2.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<B>File Name: </B><INPUT TYPE="text" Name="filename"
Value="Abstract.Cat"><BR>');
    send('<P><CENTER><INPUT TYPE="submit" Value="Print the Abstract Catalog
now"></CENTER></FORM><BR>');

    sendhr;
    send('<FORM ACTION="../../cgi-win/Adminpwd.exe" METHOD="POST">');
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
    send('<P><CENTER><INPUT TYPE="submit" Value="Return to the System Admin
Options page"></CENTER></FORM>');
    send ('</BODY></HTML>');
    closeStdout;
    closeApp( application );

end; {with cgiEnvData1 do}

end; {FormCreate}

end.

```

Print Confirmation: **Prabsca3.exe**

unit Prabsca3;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, DBTables, DB, Cgldb, Cgi;

type

TForm1 = class(TForm)
 DataSource1: TDataSource;
 Articles: TTable;
 CGIEnvData1: TCGIEnvData;
 CGIDB1: TCGIDB;
 Authors: TTable;
 AuthorsANumber: TIntegerField;
 AuthorsLName: TStringField;
 AuthorsFName: TStringField;
 AuthorsInitial: TStringField;
 AuthorsHonorific: TStringField;
 AuthorsInstitution: TStringField;
 AuthorsDepartment: TStringField;
 AuthorsMailstop: TStringField;
 AuthorsAddress_Street: TStringField;
 AuthorsAddress_City: TStringField;
 AuthorsAddress_State: TStringField;
 AuthorsAddress_Zip: TStringField;
 AuthorsCountry: TStringField;
 AuthorsPhone_LocalNumber: TStringField;
 AuthorsPhone_FaxNumber: TStringField;
 AuthorsEmail: TStringField;
 Query1: TQuery;
 ArticlesPaper_Number: TIntegerField;
 ArticlesTitle: TStringField;
 ArticlesInvited: TStringField;
 ArticlesAccepted: TStringField;
 ArticlesContactAuthorNumber: TIntegerField;
 ArticlesSession: TStringField;
 ArticlesOrderInSession: TIntegerField;
 ArticlesPresentationTime: TStringField;
 ArticlesKeyword1: TStringField;
 ArticlesKeyword2: TStringField;
 ArticlesKeyword3: TStringField;

ArticlesAbstract: TMemoField;
ArticlesContactOrder: TStringField;
ArticlesFName2: TStringField;
ArticlesLName2: TStringField;
ArticlesInitial2: TStringField;
ArticlesInstitution2: TStringField;
ArticlesOrder2: TStringField;
ArticlesFName3: TStringField;
ArticlesLName3: TStringField;
ArticlesInitial3: TStringField;
ArticlesInstitution3: TStringField;
ArticlesOrder3: TStringField;
ArticlesFName4: TStringField;
ArticlesLName4: TStringField;
ArticlesInitial4: TStringField;
ArticlesInstitution4: TStringField;
ArticlesOrder4: TStringField;
ArticlesFName5: TStringField;
ArticlesLName5: TStringField;
ArticlesInitial5: TStringField;
ArticlesInstitution5: TStringField;
ArticlesOrder5: TStringField;
ArticlesFName6: TStringField;
ArticlesLName6: TStringField;
ArticlesInitial6: TStringField;
ArticlesInstitution6: TStringField;
ArticlesOrder6: TStringField;
Query1Paper_Number: TIntegerField;
Query1Title: TStringField;
Query1Invited: TStringField;
Query1Accepted: TStringField;
Query1ContactAuthorNumber: TIntegerField;
Query1Session: TStringField;
Query1OrderInSession: TIntegerField;
Query1PresentationTime: TStringField;
Query1Keyword1: TStringField;
Query1Keyword2: TStringField;
Query1Keyword3: TStringField;
Query1Abstract: TMemoField;
Query1ContactOrder: TStringField;
Query1FName2: TStringField;
Query1LName2: TStringField;
Query1Initial2: TStringField;
Query1Institution2: TStringField;
Query1Order2: TStringField;

```

Query1FName3: TStringField;
Query1LName3: TStringField;
Query1Initial3: TStringField;
Query1Institution3: TStringField;
Query1Order3: TStringField;
Query1FName4: TStringField;
Query1LName4: TStringField;
Query1Initial4: TStringField;
Query1Institution4: TStringField;
Query1Order4: TStringField;
Query1FName5: TStringField;
Query1LName5: TStringField;
Query1Initial5: TStringField;
Query1Institution5: TStringField;
Query1Order5: TStringField;
Query1FName6: TStringField;
Query1LName6: TStringField;
Query1Initial6: TStringField;
Query1Institution6: TStringField;
Query1Order6: TStringField;
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

  MyFileHandle: THandle;
  F: Textfile;

  theabstract : TStringList;
  theAuthors : TStringList;

  i: integer;

  dbContactOrderInt: integer;
  dbOrder2int: integer;
  dbOrder3int: integer;
  dbOrder4int: integer;
  dbOrder5int: integer;
  dbOrder6int: integer;

```

dFileName, dFileNamein, dflag: string;
PathFile, OldFile: string;

dbPaper_Number: string;
dbtitle: string;
dbinvited: string;
dbaccepted: string;
dbANumber: string;
dbSession: string;
dbSessOrder: string;
dbkeyword1: string;
dbkeyword2: string;
dbkeyword3: string;
dbabstract: string;
dbcontactOrder: string;

dbfname2: string;
dblname2: string;
dbinitial2: string;
dbinstitution2: string;
dborder2: string;

dbfname3: string;
dblname3: string;
dbinitial3: string;
dbinstitution3: string;
dborder3: string;

dbfname4: string;
dblname4: string;
dbinitial4: string;
dbinstitution4: string;
dborder4: string;

dbfname5: string;
dblname5: string;
dbinitial5: string;
dbinstitution5: string;
dborder5: string;

dbfname6: string;
dblname6: string;
dbinitial6: string;
dbinstitution6: string;
dborder6: string;

```
dbFName : string;  
dbLName : string;  
dbInitial: string;  
dbHonorific : string;  
dbInstitution : string;
```

implementation

```
{ $R *.DFM }
```

```
procedure TForm1.FormCreate(Sender: TObject);  
begin
```

```
  with CGIEnvData1 do  
    begin
```

```
      webSiteINIFilename:=paramstr(1);  
      application.onException:=cgiErrorHandler;  
      application.processMessages;
```

```
      createStdout;  
      sendPrologue;
```

```
      { Send the html page }
```

```
      send ('<HEAD>');  
      SendTitle('Asilomar Signals, Systems, and Computers System Admin: Print an Article');  
      send ('</HEAD>');  
      send('<center>
```

Asilomar Conference System Administration

```
</center>');  
      sendhr;
```

```
      { Get the FORM Data }
```

```
      dflag:= getsmallfield ('flag');  
      dFileName:= getsmallfield ('filename');
```

```
if dflag = '1' then begin
```

```
  try
```

```
    if ExtractFileExt(dFileName) = '.Cat' then dFileName:= dFileNamein
```

```

        else dFileName:= ChangeFileExt("+dFileNamein+ ", '.Cat');
PathFile:= 'c:\website\Download\' +dFileName+";
if FileExists(PathFile) then begin
    {rename old file}
    OldFile := ChangeFileExt("+PathFile+ ", '.OLD');
    renamefile(PathFile, OldFile);

    {send notice}
    send('There was an existing file named "' +PathFile+ '" which was renamed to "'
+OldFile+ "'');
    end;
    MyFileHandle := filecreate(PathFile);

finally
    {Prepare File for Writing}
    AssignFile (F, PathFile);
    Append(F);

end; {try}

```

```

With Query1 do begin
    {prepare;}
    close;
    sql.clear;
    sql.add('Select * FROM Submissi WHERE Accepted = "Y" Order by Session,
OrderInSession');
    open;

    first;

```

```

While not Query1.EOF do begin

```

```

    { Retrieve record }
    dbPaper_Number:= fieldByName ('Paper_Number').asstring;
    dbtitle:= fieldByName ('title').asstring;
    {dbinvited:= fieldByName ('invited').asstring;}
    dbaccepted:= fieldByName ('Accepted').asstring;
    dbANumber:= fieldByName ('ContactAuthorNumber').asstring;
    dbSession:= fieldByName ('Session').asstring;
    dbSessOrder:=fieldbyname ('OrderInSession').asstring;
    dbkeyword1:= fieldByName ('keyword1').asstring;

```

```

dbkeyword2:= fieldByName ('keyword2').asString;
dbkeyword3:= fieldByName ('keyword3').asString;

CGIDB1.memoToStringList (fieldbyname ('Abstract'), theabstract);

dbcontactOrder:= fieldByName ('contactOrder').asString;

dborder2:= fieldByName ('order2').asString;
dbfname2:= fieldByName ('fname2').asString;
dbinitial2:= fieldByName ('initial2').asString;
dblname2:= fieldByName ('lname2').asString;
{dbinstitution2:= fieldByName ('institution2').asString;}

dborder3:= fieldByName ('order3').asString;
dbfname3:= fieldByName ('fname3').asString;
dbinitial3:= fieldByName ('initial3').asString;
dblname3:= fieldByName ('lname3').asString;
{dbinstitution3:= fieldByName ('institution3').asString;}

dborder4:= fieldByName ('order4').asString;
dbfname4:= fieldByName ('fname4').asString;
dbinitial4:= fieldByName ('initial4').asString;
dblname4:= fieldByName ('lname4').asString;
{dbinstitution4:= fieldByName ('institution4').asString;}

dborder5:= fieldByName ('order5').asString;
dbfname5:= fieldByName ('fname5').asString;
dbinitial5:= fieldByName ('initial5').asString;
dblname5:= fieldByName ('lname5').asString;
{dbinstitution5:= fieldByName ('institution5').asString;}

dborder6:= fieldByName ('order6').asString;
dbfname6:= fieldByName ('fname6').asString;
dbinitial6:= fieldByName ('initial6').asString;
dblname6:= fieldByName ('lname6').asString;
{dbinstitution6:= fieldByName ('institution6').asString;}

{ Get Author Name information from the AUTHOR Table }

with Authors do begin

    { Move to proper record }
    open;
    first;
    while fieldByName('ANumber').asString <> dbANumber do

```



```

    next;

    { Retrieve Author Name information }
    dbFName := fieldByName('FName').AsString;
    dbLName := fieldByName('LName').AsString;
    dbInitial := fieldByName('Initial').AsString;
    dbHonorific := fieldByName('Honorific').AsString;
    dbInstitution := fieldByName('Institution').AsString;
    close;
    end; {with Authors}

    {Print Article data to file}
    Writeln(F, ' ');
    Writeln(F, ' ');
    Writeln(F, '          '+dbtitle+ ' ');
    Writeln(F, ' ');
    Writeln(F, ' ');
    Writeln(F, ' ');
    Writeln(F, ' ');
    Writeln(F, ' ');
    Writeln(F, ' ');
    Writeln(F, 'Registration Number: '+dbPaper_Number+ " ");
    Writeln(F, ' ');
    Writeln(F, 'Keyword #1: '+dbkeyword1+ " ");
    Writeln(F, ' ');
    if dbKeyword2 <> 'None' then Writeln(F, 'Keyword #2: '+dbkeyword2+ " ");
    Writeln(F, ' ');
    if dbKeyword3 <> 'None' then Writeln(F, 'Keyword #3: '+dbkeyword3+ " ");
    Writeln(F, ' ');
    Writeln(F, ' ');

    {Determine Order of Authors }
    theAuthors:= TStringList.create;

    {Send the Authors}
    Writeln(F, 'Authors:');
    Writeln(F, ' ');

    if dbContactOrder = " then Writeln (F, "+dbFName+ '' +dbInitial+ '' +dbLName+ ', '
+dbInstitution+ ")
    else begin

        {initialize the stringlist}

```

```

for i:= 0 to 6 do
  theAuthors.add('This string is not used');

  dbContactOrderInt:= strtoint(dbContactOrder);
  theAuthors[dbContactOrderInt]:= "+dbFName+ ' ' +dbInitial+ ' ' +dbLName+ ' ',
+dbInstitution+ ";

  if dbOrder2 <> " then begin
    dbOrder2Int:= strtoint(dbOrder2);
    theAuthors[dbOrder2Int]:= "+dbFName2+ ' ' +dbInitial2+ ' ' +dbLName2+ ";
    end;

  if dbOrder3 <> " then begin
    dbOrder3Int:= strtoint(dbOrder3);
    theAuthors[dbOrder3Int]:= "+dbFName3+ ' ' +dbInitial3+ ' ' +dbLName3+ ";
    end;

  if dbOrder4 <> " then begin
    dbOrder4Int:= strtoint(dbOrder4);
    theAuthors[dbOrder4Int]:= "+dbFName4+ ' ' +dbInitial4+ ' ' +dbLName4+ ";
    end;

  if dbOrder5 <> " then begin
    dbOrder5Int:= strtoint(dbOrder5);
    theAuthors[dbOrder5Int]:= "+dbFName5+ ' ' +dbInitial5+ ' ' +dbLName5+ ";
    end;

  if dbOrder6 <> " then begin
    dbOrder6Int:= strtoint(dbOrder6);
    theAuthors[dbOrder6Int]:= "+dbFName6+ ' ' +dbInitial6+ ' ' +dbLName6+ ";
    end;

  for i:= 1 to 6 do
    if theAuthors[i] <> 'This string is not used' then Writeln(F, " +theAuthors[i]+ '
');

  end; {else}

```

{Send the Abstract}

```

Writeln(F, ' ');
Writeln(F, ' ');
Writeln(F, 'Abstract: ');
Writeln(F, ' ');
for i := 0 to theabstract.count - 1 do
  Write(F, " +theabstract[i]+ ");

```


Asilomar Conference on Signals, Systems, & Computers

```
</center>');  
    sendhr;  
    send('<center>
```

Your password was not accepted!!

```
</center>');  
    send ('  
Please ensure that you are authorized to access this information.');
```

If you made an error, then please try to login again.');

```
    send ('  
If you are not authorized to access this information, please note that a log is');
```

send ('maintained that includes your URL. You may be subject to civil and/or criminal ');

send ('prosecution for your actions.

```
);  
    end; {if dflag else}
```

```
    sendhdr('3','The file has been generated!');
```

```
    send('The file "' + dFileName + '" is ready to be Downloaded.');
```

send('You may Download the file by FTP now!

```
);  
    sendhr;  
    send('
```

<FORM ACTION="..../cgi-win/Adminpwd.exe" METHOD="POST">');

```
    send('<INPUT TYPE="hidden" Name="flag" Value="1">');
```

send('

<CENTER><INPUT TYPE="submit" Value="Return to the System Admin Options

page"></CENTER></FORM>');

```
    send ('</BODY>');
```

closeStdout;

closeApp(application); { don't leave form around }

```
  
    end; {with CGIEnvData1}
```

end; {with FormCreate}

end.

LIST OF REFERENCES

1. <http://www.allaire.com>
2. <http://www.borland.com>
3. <http://www.relcom.ru/gnu/msdos/perl>
4. <http://www.href.com>
5. <http://home.city.net/win-httpd/>
6. <http://software.ora.com>
7. <http://learning.lib.vt.edu/wintcpip/wftpd.html>
8. <http://home.netscape.com>
9. <http://www.interlog.com/~argosoft>
10. <http://salsa.walldata.com/center/sal.htm>
11. <http://www.zbserver.com>
12. <http://pemclab.cn.nctu.edu.tw/cdrom/cdpc.htm>
13. <http://www.sausage.com>

References were accurate as of 11 September 1996

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
8725 John J. Kingman Road., Ste 0944
Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
Naval Postgraduate School
411 Dyer Rd.
Monterey, CA 93943-5101

3. Professor Monique P. Fargues 2
Department of Electrical and Computer Engineering, Code EC/FA
Naval Postgraduate School
Monterey, CA, 93943-5121

4. Professor Kishore Sengupta 1
Department of Systems Management, Code SM/SE
Naval Postgraduate School
Monterey, CA, 93943-5121

5. Lt Kevin M Coats 1
Commander, Cruiser-Destroyer Group EIGHT
Unit 60009
FPO AE 09506-4704

6. Embry-Riddle Aeronautical University 1
attn: Lucille Babcock
NAS Cecil Field Resident Center
Building 199, Box 157
NAS Cecil Field, FL 32215

7. Lt. Michael Chalfant 1
1260 6th St. Apt. #6
Monterey, CA, 93940