NISTIR 6321

The NIST LEIDIR Prototype - Inserting Hypertext Links into the POMS Using Information Retrieval.

Installation Guide, User Guide and Software Documentation

John M. Tebbutt

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Information Technology Laboratory
Information Access and User Interfaces Division
Gaithersburg, MD 20899

NIST

NISTIR 6321

The NIST LEIDIR Prototype - Inserting Hypertext Links into the POMS Using Information Retrieval.

Installation Guide, User Guide and Software Documentation

John M. Tebbutt

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Information Technology Laboratory
Information Access and User
Interfaces Division
Gaithersburg, MD 20899

May 1999

The NIST LEIDIR Prototype

Inserting Hypertext Links into the POMS Using Information Retrieval

Installation Guide, User Guide, and software documentation

CONTENTS

Overview

This document provides comprehensive information about the LEIDIR prototype automatic hypertext generation system, developed by the National Institute of Standards and Technology (NIST) in collaboration with Social Security Administration (SSA). The system creates a hypertext version of the SSA's *Program Operations Manual System* (POMS) from the familiar CDROM distribution.

This document is organized as follows:

- Introduction to the LEIDIR system, how it works, and the advantages it offers;
- Description of resources required by the system, including other software;
- Installation Guide;
- User Guide;
- Description of the directory structure used by the system;
- Documentation of the purpose and operation of each system module.

NIST Disclaimer

The LEIDIR software produced by NIST, an agency of the U.S. government, is by statute not subject to copyright in the United States. Recipients of this software assume all responsibilities associated with its operation, modification and maintenance.

Any mention of specific products, services or software does not imply their endorsement by the Federal Government of the United States of America or any agencies thereof.

Introduction

The following is a brief description of the operation of NIST's prototype automatic hypertext generation system. We call the system LEIDIR (Link-Enhanced Information Discovery using Information Retrieval, pronounced "LAY-deer"). This description relates specifically to its use with the POMS manual.

The software is designed to create a web of documents from disjoint documents or document collections of arbitrary size and format. This is achieved by inserting into each document semantic links to the *n* most similar documents in the collection, where *n* is a user-configurable value. We refer to the inserted links as *semantic links* because they convey a relationship between the content or subject matter of the linked documents. The finished documents are formatted in HTML, making them accessible via any World Wide Web browser.

The system is designed with the goal of providing automatic maintenance of the semantic web, as necessary. At the moment this requires rebuilding the web and is thus limited to an overnight run.

LEIDIR has four principal software components:

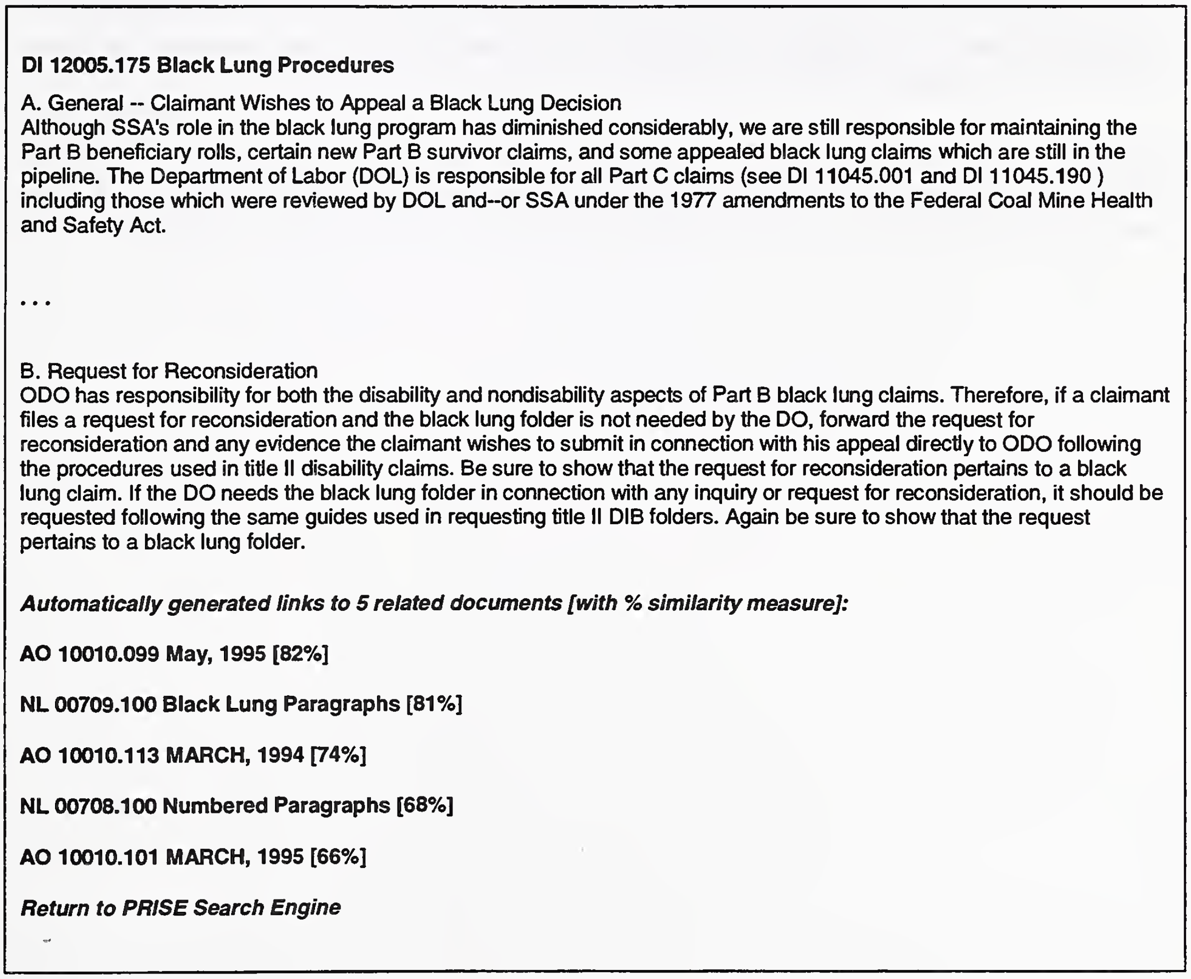
- a variable preprocessor module;
- an invariant main processor module;
- a search engine, and
- a browser capable of displaying HTML-formatted documents.

The preprocessor module is used to break up the initial document collection (i.e. the text files comprising the POMS in the CDROM distribution) into a second generation collection of documents of a manageable size, and to format these documents into HTML. The notion of *manageable size* reflects several important factors:

- the documents should be of a size that is optimal for users – this depends on the user community in question;
- the documents should be of a size that is acceptable as a query to the search engine;
- the documents should be self-contained, i.e. the user should not need to refer to other documents in order to put a document in context.

The main processor module is the heart of LEIDIR, and yet, for the time being, its operation remains relatively straightforward: each second generation document is passed to the search engine as a query over the entire second generation collection. HTML links to the top *n*-ranked (see above) documents are then inserted at the end of the query document, thus creating a document collection in which each document is linked to *n* others. The inserted links are preceded by a short text label which identifies their origin and function. Along the way, a similarity calculation is employed to assign a *Percentage Similarity Measure* (PSM) to each inserted link: the idea behind this measure is to give a rough estimate of the similarity between the document containing the link and the target document of the link, thus aiding the user in making the decision as to whether to follow the link. Finally, a link to the search engine interface is inserted at the end of each document.

Figure 1 shows the general appearance of a completed document, with links.



DI 12005.175 Black Lung Procedures

A. General -- Claimant Wishes to Appeal a Black Lung Decision
Although SSA's role in the black lung program has diminished considerably, we are still responsible for maintaining the Part B beneficiary rolls, certain new Part B survivor claims, and some appealed black lung claims which are still in the pipeline. The Department of Labor (DOL) is responsible for all Part C claims (see DI 11045.001 and DI 11045.190) including those which were reviewed by DOL and--or SSA under the 1977 amendments to the Federal Coal Mine Health and Safety Act.

. . .

B. Request for Reconsideration
ODO has responsibility for both the disability and nondisability aspects of Part B black lung claims. Therefore, if a claimant files a request for reconsideration and the black lung folder is not needed by the DO, forward the request for reconsideration and any evidence the claimant wishes to submit in connection with his appeal directly to ODO following the procedures used in title II disability claims. Be sure to show that the request for reconsideration pertains to a black lung claim. If the DO needs the black lung folder in connection with any inquiry or request for reconsideration, it should be requested following the same guides used in requesting title II DIB folders. Again be sure to show that the request pertains to a black lung folder.

Automatically generated links to 5 related documents [with % similarity measure]:

AO 10010.099 May, 1995 [82%]

NL 00709.100 Black Lung Paragraphs [81%]

AO 10010.113 MARCH, 1994 [74%]

NL 00708.100 Numbered Paragraphs [68%]

AO 10010.101 MARCH, 1995 [66%]

Return to PRISE Search Engine

Figure 1. A typical POMS Section page augmented with semantic links (abridged).

The finished document comprises a title and associated text from the POMS, followed by a number of links to other documents dealing with similar subject matter, followed by a link to the text search page - the home page of the LEIDIR/POMS collection.

The amount of POMS text in each document is decided at runtime (possibly by the system administrator), and depends upon the level at which the LEIDIR software is instructed to break up the POMS text, e.g. Chapter, Subchapter, Section, and so on. The higher the level, the more text per document.

The output of LEIDIR is marked up (or tagged) in a slightly augmented form of HTML. Three additional tag sets are employed: **<DOC>** and **</DOC>** to delimit individual documents, and **<DOCNO>** and **</DOCNO>** to encapsulate a unique identifier for each document (these tag sets are required by the PRISE search engine); finally **<NIST-HL>** and **</NIST-HL>** delimit the document library files in which documents are stored. These augmentations are not disruptive to browsers.

The number of links contained in each document is also decided at runtime, and can vary from 0 through 10.
The search engine we use is NIST's experimental PRISE system, a statistically-based, ranking information retrieval engine. The search engine serves two vital functions: it is essential in the creation of the semantically linked document web, as described above: the ranked output from the search engine determines which semantic links will be inserted between documents; it also serves as an entry point into the document web. The user initially submits a text query to the search engine, and is returned links to a number of documents in the collection, ranked in order of relevance. Following one of these links takes the user to a document, and the user is then free to browse within the web using the semantic links included in each document.

An HTML browser is essential for viewing the document web, and the choice of browser can markedly affect the utility of the system. Such features as a search utility, a good history mechanism, bookmarking facility, and display configuration all contribute significantly accessibility of the completed document web.

System and Software[1]

Hardware

The system was developed on a Sun Ultra 1 workstation with 128MB of RAM. Note that significant compute power is required for the text collection indexing and text linking phases of processing, as these are compute intensive and involve large amounts of data.

The size of the linked POMS collection is approximately 125MB, and at least double this amount is required during processing, so approximately 300MB of hard disk storage should be available for the data alone.

Software

The system was developed under Solaris 2.5.1, a Unix System V variant.

The bulk of the code is Perl 5.004, with other units relying on the Unix scripting languages *awk* and *csh.*

Perl compilers, interpreters and other information and support can be obtained at [2].

The system relies on NIST's PRISE ranking statistical search engine to index the POMS text collection, query the collection as a part of link formation, and to provide the user search interface. Thus PRISE should be installed prior installation of LEIDIR.

Information on obtaining and installing PRISE can be found at [1].

[1] Please see the section "NIST Disclaimer" on page 3 of this report.

Installation Guide

This section provides step by step instructions on how to install and configure the LEIDIR system from the distribution file provided.

Installing the System

1. Create a directory under which to install LEIDIR (the LEIDIR home directory), e.g. /home/LEIDIR. This directory will be referred to here as $LEIDIR;

2. Place the distribution archive file, *leidir.tar,* in the LEIDIR home directory, $LEIDIR;

3. In the LEIDIR home directory, extract the LEIDIR files and directories from the distribution archive, e.g.:

   tar xvf leidir.tar

4. The directory structure under the LEIDIR home directory should be as described below, under *Directory Structure;*

5. Edit the file *$LEIDIR/scripts/pextract* (this is a C-shell script):

   - Under the caption "#Initialize Directory names and other variables", complete the line

     set TOP =

     with the value of $LEIDIR, e.g.

     set TOP = /home/LEIDIR

     Next, complete the line

     set SRC =

     with the name of the directory containing the POMS source files (these are typically the files on the CDROM), e.g.

     set SRC = /cdrom/ssanew9_98/poms

   - Save the file and exit.

6. Edit the file *$LEIDIR/scripts/hlbuild* (this is a C-shell script):

   - Under the caption "#Initialize Directory names and other variables", complete the line

     set BASE =

     with the URL of the directory in which the linked files will reside, e.g.

     set BASE = http://www.ssa.gov/poms

     Next, complete the line

set LEIDIR =

with the value of $LEIDIR, e.g.

set LEIDIR = /home/LEIDIR

- Under the caption "#Aliases for program names", fill in the full pathnames of the PRISE indexer and batchprise programs the system will be using, e.g.

```
set INDEXER        = /home/pomsguy/bin/build.script.sh
set BATCHPRISE     = /home/pomsguy/bin/batchprise
```

- Save the file and exit.

7. Since LEIDIR is written entirely in scripting languages, it is essential that the location of the script interpreter, given on line 1 of every file in directory $LEIDIR/scripts, matches the actual location of the corresponding interpreter on the system on which LEIDIR is to be run. The interpreters in question are given as **#!/bin/csh** for C-shell scripts and **#!/usr/local/perl5.004/perl** for the Perl scripts. If the corresponding interpreters on your system are not in these locations, each file in the $LEIDIR/scripts directory should be edited to reflect this.

8. Installation is complete.

Configuration Files

The system has only one configuration file (though various preset configuration files are provided for the PRISE system, see below): *$LEIDIR/scripts/pomslist.* This file lists the names of the files containing the text of the POMS Parts, as they appear on the October 1998 edition of the CDROM (*/ssanew10_98/poms/*.txt*), minus the *.txt* suffix. *pextract* uses this file to determine which source files should be processed. The contents of this file should be checked to ensure that they match the names of the *.txt* files on the CDROM distribution in use and modified if they do not.

Over time, the contents of *pomslist* may need to be modified as the names of the *.txt* source files change. In addition, the contents of this file may be modified for testing purposes (see the User Guide under "Partial Execution").

Additional configuration files are provided for the PRISE indexer, in directory *$LEIDIR/index*, and should not be modified under normal circumstances:

$LEIDIR/index/PM.dtd	- describes the structure of documents in terms of SGML;
$LEIDIR/index/commonwords	- a list of common words to be ignored by the indexer;
$LEIDIR/index/options.spec	- configuration settings for the indexer;
$LEIDIR/index/sgmls.actions	- defines indexer behavior in response to SGML markup.
$LEIDIR/index/title_tags	- defines the markup used to identify document titles

Finally, the following HTML files are provided with sample content:

$LEIDIR/index/abstract.html
$LEIDIR/index/contact.html
$LEIDIR/index/title.html

The Web interface to the PRISE search engine (LEIDIR's search screen) uses these files to build the search screen interface:

- *abstract.html* is intended to hold usage instructions, background information, and so on - its content forms the body of the page;

- the content of *contact.html* appears at the bottom of the screen and is intended to hold contact information in case of problems, questions, comments, etc.;
- the content of *title.html* is placed at the top of the screen and gives a title to the search screen or the system (since the search screen is the entry point into the system).

User Guide

This section describes the steps necessary to create a hypertext version of the POMS using LEIDIR, as well as some pitfalls that may be encountered and methods to circumvent them.

Preliminary Steps

Prior to using LEIDIR, be sure that the following conditions are met:

1. The PRISE system is installed and operational (see [1]);
2. An up-to-date Perl interpreter is available (preferably version 5.004 or later - see [2]);
3. The LEIDIR system has been installed, as detailed in the Installation Guide;
4. The POMS source material is installed and readily accessible.

Creating the POMS Hypertext

There are two steps involved in creating the POMS hypertext:

1. The POMS source text is broken up into documents and converted to HTML using the command:

   $LEIDIR/scripts/pextract *<level>*

Where *level* denotes the structural level at which the text should be broken up, and may take any value from 3 through 6 (3 = Chapter, 4 = Subchapter, 5 = Section, 6 = Subsection).

While running, *pextract* outputs information about the various documents and files it is creating.

2. The documents output by *pextract* are processed into a hypertext using the command:

   $LEIDIR/scripts/hlbuid *<links>*

Where *links* denotes the number of hypertext links to be inserted at the end of each document, and may take any value from 0 (no links inserted) through 10 (ten links per document).

While running, *hlbuild* outputs information its progress through the various stages of processing. Part of this output comprises output from the PRISE indexer.

Execution Times

The POMS represents a sizeable text collection (approximately 125MB for the POMS of October 1998), and execution times for the LEIDIR prototype are correspondingly long. *pextract* can be expected to take from 1/2 - 1 hour, while *hlbuild* may require 18 hours or more of processing time. Execution time for *hlbuild* can vary considerably, and has a linear relationship to number of documents generated by *pextract*. This in turn depends on the size of the POMS for a given month, the internal text subdivisions within the POMS, and the level at which the POMS is parsed. The sample times above are for the October 1998 POMS parsed at the Section level.

The times given are for a moderately powerful desktop workstation (Sun UltraSparc 1) with no additional load and with all working directories served locally. The principal **html** directory, however, was remotely served across a 100MB/s PEPNet connection.

Partial Execution

It is possible (and desirable for testing purposes) to process a POMS Part or a collection of Parts without having to process the entire POMS. The file *$LEIDIR/scripts/pomslist* contains the list of POMS Parts to be processed, as identified by the names of the files containing these parts on the source distribution or CDROM.

To process a subset of the POMS, create a backup copy of the file *$LEIDIR/scripts/pomslist*, and then create a new version of the file containing only the names of those Parts which are to be processed.

Troubleshooting Tips

Though the LEIDIR system is a prototype, it has remained stable and reliable since February 1997. The main problems encountered so far have come as a result of changes to the formatting conventions or the presence of special characters in the input text (i.e. the files in the CDROM distribution).

The Document Type Definition (DTD) file for use by the PRISE indexer, *$LEIDIR/index/PM.dtd,* has recently been modified such that all characters in the range 0 - 255 decimal (the characters of the ISO8859-Latin-1 alphabet) are accepted as valid. Thus it is very unlikely that new special characters will be added to POMS distributions in the foreseeable future which the system will be unable to handle.

Changes to formatting conventions, specifically the adoption of new text markup tags, are less predictable and will need to be dealt with as follows:

> Assume that the new markup consists of beginning and end tags, **<new_tag>** and **</new_tag>.**
>
> Edit the file *$LEIDIR/scripts/normalize* (this is part of the preprocessor software);
>
> Under the comment "#Remove the <phrase>, <tbl> and <tblhdr> markup" (line 41) insert the lines
>
> ```
> s/<new_tag>//g;
> s/<\/new_tag>//g;
> ```
>
> Note that the character "/" in the end tag, "</new_tag>" must be escaped with a backslash character ("\").

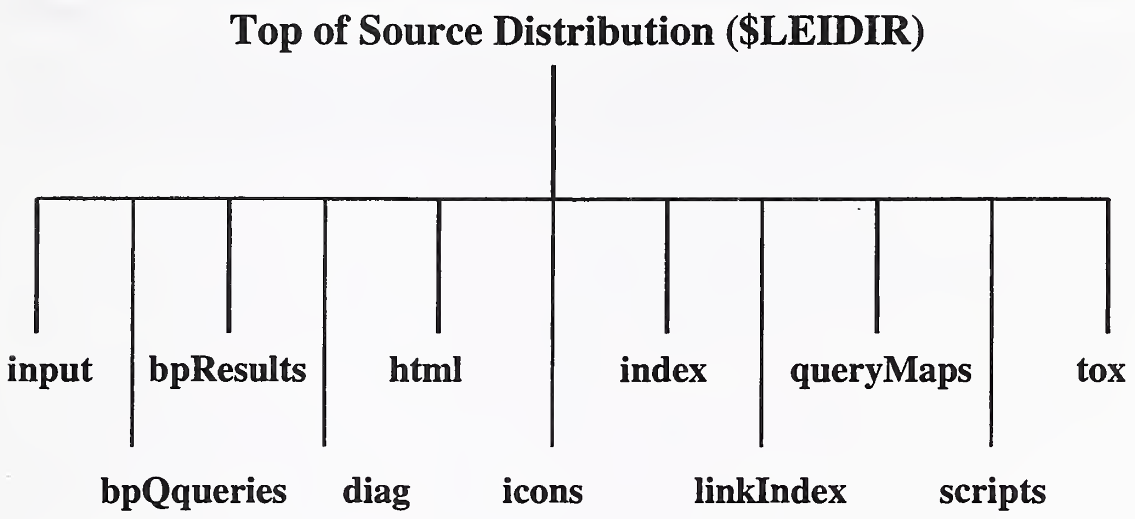
This ensures that the unfamiliar tags will be removed during processing.

Errors of this type will come to light during the SGML verification phase of the PRISE indexing process, and are output to the screen unless redirected. In order for indexing (and thus processing) to complete successfully, the errant tags should be identified and instructions for their removal inserted, as above. *pextract* should then be run before running *hlbuild* again.

Note: during SGML verification we found that a dozen or so errors regarding the **NAME** attribute of HTML are typically reported. These errors do not appear to affect the indexing process and are no cause for concern.

Directory Structure

The figure below shows a schematic layout of the directory structure of the LEIDIR system. Most of the directories shown hold temporary files, generated during processing and deleted prior to completion. The directories and their functions are described below. Note that the directory names used here are those the software expects to see. They can generally be changed, if necessary, by editing the module hlbuild, where all directory names are aliased.



Top of source distribution ($LEIDIR): This is the point in the directory structure at which the LEIDIR distribution is installed, e.g. */home/POMS/leidir.* All pathnames in the distribution are relative to this one.

input: This is the directory into which the preprocessor module, *pextract*, writes its output and where the main processor module, *hlbuild*, looks for its input. Once read, the files in this directory are discarded.

bpQueries: This directory holds the *batchprise* queries created by the module *makeBPQueries*.. They are used by *runBP* and discarded afterwards.

bpResults: The results files from *batchprise* are placed here, used by *hyperLinker* and then discarded.

diag: Diagnostic output messages are typically written to files in this directory. Most of the modules use a file named *hs.out.* Since the distribution was completed, most of the diagnostic output has been removed.

html: This is where the system's final output files are deposited by module *hyperLinker*. This directory is also used for the processing of HTML files along the way: the initial HTML document library files are written here by the module *linkindex*, and the files in this directory are also used by the modules *insertTitles, makeIndexerList, build.script.sh* and *makeBPQueries*. The files in this directory are not deleted after processing, since they embody the POMS hypertext.

icons: This directory is designed to hold any icons or image files for use by the system. At present it is the home of a small GIF file, *red_bull.gif*, which holds a small red bullet used to accentuate the link to the search engine which LEIDIR inserts at the end of each document.

index: This directory holds all of the information required by and produced by the PRISE indexer (see [1]). Several files are supplied with the distribution, the names and functions of which are described below. The files in this directory are administered by the PRISE software and should be considered permanent.

linkIndex: This directory holds temporary link indexing files generated by module *linkIndex* and used by module *hyperLinker*. The files are deleted after processing.

queryMaps: This directory holds temporary document-query mapping files generated by module *makeBPQMaps* and used by module *hyperLinker*. The files are deleted after processing.

scripts: This is where the source modules of the distribution reside. These files should be considered permanent!

tox: The various table of contents files generated during processing are stored here. They are not typically used, because of their large size, but may be of interest. Each file holds a list of links to the various sections of the POMS Part it represents in the form of a table of contents.

Software Modules

Module Name:	**pextract** ***<level>***
Called By:	Command line
Calls:	normalize, sep_poms.awk
Files:	$LEIDIR/scripts/pomslist - list of input files to be processed; $LEIDIR/input - directory into which output files are written; $LEIDIR/tox - directory into which top level (i.e. POMS Part) tables of contents are written.
Language:	csh
Description:	This is the top level preprocessor module for the POMS. For each of the POMS parts listed in $LEIDIR/scripts/pomslist, the POMS input is parsed into smaller *texts*, the boundaries of which are determined by the command line parameter *level*. The texts are formatted into HTML and written into document library files in directory $LEIDIR/input. An HTML table of contents is created for each POMS part, listing the titles and identifiers of the documents, e.g. **GN 05.001 Application Forms**. The table of contents files are written to directory $LEIDIR/tox. *level* may take any value from 3 through 6, which indicates the text boundary class at which to split the input into text for processing (3 = Chapter; 4 = Subchapter, 5 = Section, 6 = Subsection).

Module Name:	**normalize** ***<filename>***
Called By:	pextract
Calls:	-
Files:	-
Language:	Perl 5.004
Description:	This file contains instructions for the removal or replacement in file *<filename>* of proprietary markup tags, escape characters and other characters or character sequences as part of the reformatting of the POMS input text into HTML.

Module Name:	**sep_poms.awk** ***<work directory> <source filename> <toc name> <level> <filename>***
Called By:	pextract
Calls:	-
Files:	-
Language:	awk
Description:	This module takes the output of module *normalize* (*<filename>*), breaks it up into documents based on *level*, the text boundary class supplied on the command line to *pextract*, and writes the documents out to library files about 1,000 lines in length. The library files are of variable length because documents are intentionally not split across library files. Thus, once 1,000 lines have been written to a library file, the last document is written out in full before a new library file is created. As each new document is written, it is formatted into HTML. The documents are delimited using <DOC> and </DOC> markup to denote the beginning and end of each document, and are uniquely identified using a <DOCNO>...</DOCNO> element. The library files are delimited using the augmenting tags <NIST-HL> and </NIST-HL>. *work directory* and *source filename* are the name of the directory in which the input file resides and the local filename of the input file, respectively: these are used in the creation of the output filenames. Note that the output files are written to *work directory*.

toc name is the name of the table of contents file which the script creates for the output files. The table of contents files are written to directory $LEIDIR/tox.

Module Name:	**hlbuild** ***<no. of links>***
Called By:	Command line
Calls:	linkIndex, insertTitles, makeIndexerList, build.script.sh, makeBPQueries, runBP, hyperLinker, postProcess
Files:	$LEIDIR/input/*, $LEIDIR /linkIndex/*, $LEIDIR/bpQueries/*, $LEIDIR/bpResults/*, $LEIDIR/queryMaps/*, $LEIDIR/diag/hs.out, $LEIDIR/icons/red_bull.gif, $LEIDIR/index/list
Language:	csh
Description:	This is the top level hypertext creation module. It operates on the document library files output by the preprocessor module.

The command line parameter *no. of links* is checked. It should be an integer with a value of 0 through 10 and denotes the number of hypertext links to be appended to each document.

Module Name:	**linkIndex** ***<source dir> <dest dir> <linkIndex dir> <final dir> <diag>***
Called By:	hlbuild
Calls:	-
Files:	-
Language:	Perl 5.004
Description:	This module takes the document library files created by the preprocessor and stored in *<source dir>* as input. These files are processed in three ways and written out to *dest dir:*

1. Within each input library file, the <DOCNO> identifier of each document is modified to reflect the name of the library file in which it is stored. This is done so that the link index file for the document (see below) can be rapidly identified during the linking process,. ensuring a linear relationship between the document collection size and the processing time required for linking;

2. Within each input library file, the highest-level header in each document is formatted into an HTML anchor - this is the anchor which will be used to retrieve the document whenever a link to it is activated during subsequent use;

3. For each input library file, a link index file is created in directory *<linkIndex dir>* which consists of HTML links to all of the document headers processed in step 2. The link index files can also be used in the production of tables of contents.

<final dir> is the name of the directory where the library files of linked documents will eventually reside, used in the construction of HTML <BASE> elements.

<diag> is the name of the diagnostic output file.

Module Name:	**makeIndexerList** ***<HTML dir> <DTD> <list> <indexer dir> <diag>***
Called By:	hlbuild
Calls:	-
Files:	-
Language:	Perl 5.004
Description:	This module creates a list, *<list>*, for use by the PRISE indexer. Each line consists of the full pathname of a document library file to be indexed, followed by a single space, followed by the local name of the DTD file for the collection, *<DTD>*, within the indexer

directory, *<indexer dir>*. For example,

```
...
/home/ssa/poms/html/file74 POMS.dtd
...
```

The DTD file holds a description of the structural elements comprising a document for use by SGML-based applications (of which PRISE is one).

<HTML dir> is the directory in which the document libraries are located (/home/ssa/poms/html in our example), and *<diag>* is the diagnostic output file.

Module Name:	**build.script.sh** ***<index dir> <list>***
Called By:	hlbuild
Calls:	-
Files:	-
Language:	sh
Description:	This is the PRISE indexer and is not included in the LEIDIR distribution. It collects statistics about the occurrences of words in the document collection for later use in searching. See [1] for details.

<index dir> is the name of the directory in which index information for the collection is to be stored. *<list>*is the list produced by *makeIndexerList.*

Module Name:	**makeBPQueries** ***<HTML dir> <script dir> <query dir> <query map dir>***
Called By:	hlbuild
Calls:	makeBPQMap, makeBPQueries.prl
Files:	-
Language:	Perl 5.004
Description:	This module creates library files of queries suitable as input to the PRISE utility *batchprise,* from the HTML document library files in *<HTML dir>* and writes them to *<query dir>*. At the same time, mapping files of the relationship between query files, document files and their associated filenames are created in *<query map dir>.*

Queries must have the form:

<unique query ID><SPACE>"--"<SPACE><special text>

Module Name:	**makeBPQMap** ***<seqno> <bpq dir> <map dir> <file>***
Called By:	makeBPQueries
Calls:	-
Files:	-
Language:	Perl 5.004
Description:	This module takes a copy of the input document library file, *<file>*, and inserts an integer suitable as a *batchprise* query number at the start of each document. The query number consists of a prefix, the sequence number, *<seqno>*, of the document library file being processed, and a suffix, which is the ordinal number of the document within the library file. The modified library file is then copied out to the directory *<bpq dir>* for further processing, with the suffix "-qry" appended to the original library file name, *<file>*.

A query mapping file is created for each document library in the directory *<map dir>*. This file contains entries, one per line, associating the query number of each query with the document identifier, <DOCNO>, of the document from which it was created. This is for use later during linking.

Module Name:	**makeBPQueries.prl** ***<file>***
Called By:	makeBPQueries
Calls:	-
Files:	-
Language:	Perl 5.004
Description:	This module takes the output of makeBPQMap, *<file>*, and converts the contents into the "special text" format required by batchprise. This text must contain only words and spaces, thus this module simply strips all punctuation, markup and newlines/carriage returns from *<file>*.

Module Name:	**runBP** ***<links> <program> <query dir> <output dir> <index dir> <diag>***
Called By:	hlbuild
Calls:	batchprise
Files:	-
Language:	Perl 5.004
Description:	This is a harness to run the PRISE batch search program, *batchprise*. Each file in directory *<query dir>* is processed as a batch of queries to the search engine, and the results written out to a similarly named file in the *<output dir>* directory. The *batchprise* output file has *<links>* results per query, with one line per result. Each line contains the query number, and within each group of results bearing the same query number, the results are listed in order of relevance, or similarity to the query document. Each result line contains the <DOCNO> identifier of the document it represents. Note: One of the results returned by *batchprise* will almost certainly point to the query document, as it is the most similar document to itself in the collection. Provisions are made to account for this and ignore this result. For more details on *batchprise*, see [1]. *<program>* holds the name of the *batchprise* program. *<index dir>* holds the name of the index directory. *<diag>* holds the name of the diagnostic output file.

Module Name:	**hyperLinker** ***<links> <query map dir> <bp result dir> <level> <diag> <link index dir> <HTML dir>***
Called By:	hlbuild
Calls:	-
Files:	red_bull.gif
Language:	Perl 5.004
Description:	This module performs the final linking stage of the processing: it integrates the information produced by *batchprise*, *linkIndex* and *makeBPQueries* to append *<links>* hypertext links to each document in the POMS collection. For each document in each document library file in *<HTML dir>*, the <DOCNO> identifier of the document is read, and this is looked up in the query map files residing in *<query map dir>*. The *batchprise* query ID corresponding to the document is retrieved from the query map file, and the corresponding *batchprise* results are retrieved from the *batchprise* result file in directory *<bp result dir>*. The <DOCNO> identifier is extracted from each result line and is used to locate the pre-existing hypertext link to that document in the link index file in directory *<link index dir>*. The link index file to search is determined from the suffix of the <DOCNO> identifier, which was set by *linkIndex* (see above) for precisely this purpose. The HTML linkline is inserted into the document in a font size determined by *<level>*. Finally, a link to the PRISE search screen is appended.

Module Name: **makeBPQueries.prl *<file>***
Called By: makeBPQueries
Calls: -
Files: -
Language: Perl 5.004
Description: This module takes the output of makeBPQMap, *<file>*, and converts the contents into the "special text" format required by batchprise. This text must contain only words and spaces, thus this module simply strips all punctuation, markup and newlines/carriage returns from *<file>*.

Module Name: **runBP *<links> <program> <query dir> <output dir> <index dir> <diag>***
Called By: hlbuild
Calls: batchprise
Files: -
Language: Perl 5.004
Description: This is a harness to run the PRISE batch search program, *batchprise*. Each file in directory *<query dir>* is processed as a batch of queries to the search engine, and the results written out to a similarly named file in the *<output dir>* directory. The *batchprise* output file has *<links>* results per query, with one line per result. Each line contains the query number, and within each group of results bearing the same query number, the results are listed in order of relevance, or similarity to the query document. Each result line contains the <DOCNO> identifier of the document it represents.

Note: One of the results returned by *batchprise* will almost certainly point to the query document, as it is the most similar document to itself in the collection. Provisions are made to account for this and ignore this result. For more details on *batchprise*, see [1].

<program> holds the name of the *batchprise* program. *<index dir>* holds the name of the index directory. *<diag>* holds the name of the diagnostic output file.

Module Name: **hyperLinker *<links> <query map dir> <bp result dir> <level> <diag> <link index dir> <HTML dir>***
Called By: hlbuild
Calls: -
Files: red_bull.gif
Language: Perl 5.004
Description: This module performs the final linking stage of the processing: it integrates the information produced by *batchprise*, *linkIndex* and *makeBPQueries* to append *<links>* hypertext links to each document in the POMS collection.

For each document in each document library file in *<HTML dir>*, the <DOCNO> identifier of the document is read, and this is looked up in the query map files residing in *<query map dir>*. The *batchprise* query ID corresponding to the document is retrieved from the query map file, and the corresponding *batchprise* results are retrieved from the *batchprise* result file in directory *<bp result dir>*. The <DOCNO> identifier is extracted from each result line and is used to locate the pre-existing hypertext link to that document in the link index file in directory *<link index dir>*. The link index file to search is determined from the suffix of the <DOCNO> identifier, which was set by *linkIndex* (see above) for precisely this purpose. The HTML linkline is inserted into the document in a font size determined by *<level>*.

Finally, a link to the PRISE search screen is appended.

<diag> holds the name of the diagnostic output file.

Module Name:	**postProcess *<HTML dir> <script dir> <index dir> <diag> <icon dir>***
Called By:	hlbuild
Calls:	docmap
Files:	red_bull.gif
Language:	Perl 5.004
Description:	This module performs final administration operations. The file *red_bull.gif* is copied from the *<icon dir>* to the *<HTML dir>* directory. This file contains an image of a small red bullet, used to highlight the link at the end of each document to the PRISE search screen.

The document mapping function of the PRISE indexer, *docmap*, is run again on the completed document library files. This needs to be done in order to allow the search engine to locate the HTML documents comprising the POMS, which have changed in size through being augmented with hypertext links. For more information on *docmap*, see [1].

References

[1] Guide to Z39.50/PRISE 2.0: http://www.nist.gov/itl/div894/894.02/works/papers/zp2/zp2.html

[2] PERL.COM, the Perl homepage: http://www.perl.com/