

## Historic Pittsburgh Full-Text: Mounting Process (SSP-era Historic Pittsburgh Full-Text)

*author: Aaron Brenner*

*revision: 1.1 changed paths to reflect move to Atlantis  
date: 2003.01.07*

*revision: 1.0 original  
date: 2001.11.26*

The contents of a shipment should be loaded from the Windows server onto the UNIX production server (Atlantis) in the following directory:

```
/usr/local/images/hpbooks/incoming/
```

Change directories to be there. The contents of the directory should now have the following structure:

```
/usr/local/images/hpbooks/incoming/{uniqueid}/v0000/i000/
```

so an example might be:

```
/usr/local/images/hpbooks/incoming/00aga8934m/v0000/i000/
```

within this directory should be (at least):

1. A TIFF file for every page image
2. A text file (created from OCR) for the full-text of every page image
3. A document.txt file for the book
4. A document.txt.munged file for the book (created from earlier processing)
5. A metadata.toc file for the book
6. ..and other files left over from qc, ftp, etc.

Okay, back to where we were: `/usr/local/images/hpbooks/incoming/` . We want to do the following tasks:

- a) run `lintocr.pl` over all `.txt` files for every book)
- b) run `newmonoconcat.pl` over everything
- c) copy the SGML file for every book into a single place (which,

in this case is: `/usr/local/images/hpbooks/sgmfiles/`

One way to do this is by using a shell script (in tcsh):

**But first, make sure to change the “today’s date” variable in `newmonoconcat.pl`!**

```
% foreach i (*)
foreach? cd /usr/local/images/hpbooks/incoming/$i/v0000/i000
foreach? /usr/local/dlxs/legacy/sspbin/lintocr.pl *.txt
foreach? /usr/local/images/hpbooks/newmonoconcat.pl
foreach? cp $i.sgm /usr/local/images/hpbooks/sgmfiles/
foreach? echo $i
foreach? end
```

change directories to

```
/usr/local/images/hpbooks/sgmfiles
```

All of the SGML files for the shipment that we are adding should be here now. These will eventually have their “validity” (in a markup-language-manner-of-speaking) tested, but it’s easier to test them now when they are separate files. I’ve created a simple shell script that performs a batch validation on the SGML files, and creates a moderately-formatted report of any errors. The two pieces necessary are: 1) the shell script `batchvalidate.sh` and 2) a Perl script called `cleanreport.pl`.

Run the shell script now, from within the `sgmfiles` directory:

```
sh batchvalidate.sh
```

The error report is shown on screen, but is also saved as `report.txt` in the same directory.

Remember, it’s best (though frustrating) to correct any errors all the way up the chain—that is, correct spreadsheet errors in the `document.txt`, the `document.txt.munged`, and back to the Excel spreadsheets that live in `P:\Histpitt\hpbooksheets`.

When all the errors are fixed, it’s time to combine all the SGML files into one huge file that will be normalized and indexed. Run the `pittconcat.pl` script from within this directory.

```
./pittconcat.pl
```

The `pittconcat.pl` script concatenates all of the SGML files in its directory into a single, very large file that it names `testdata.unnorm.sgm2`. Copy and rename this file like so (all one line):

```
cp testdata.unnorm.sgm2
/usr/local/images/hpbooks/indextemp/pitttext.unnorm.sgm
```

In this indextemp directory is a makefile that will do the rest of the work to create the index. The makefile runs in two steps, initiated by the commands "make norm" and then "make index". Type:

```
make norm
```

"Make norm" will run `/usr/local/dlxs/legacy/sspbin/sgmlnorm`, a program that will normalize the SGML and check its validity against its DTD. If there are any problems with the SGML structure or validity, the makefile will exit with an error, and the SGML file cannot be indexed. (For a tool to find which SGML files are invalid and causing the problem, see the documentation titled "How to Batch Validate SGML Files") "Make norm" also runs `cleanlines.pl` over the SGML file.

When "make norm" works, type:

```
make index
```

...and the index will be made (this can take a while due to the size of the files involved). "Make index" does three things: 1) creates the data dictionary files; 2) creates the regions that are necessary; and 3) creates the index.

When the index is made, the idea is to eventually move the index file and all of the associated `.dd` and `.rgn` files into the "live" index directory that is accessed by the CGI script used for searching. This index directory is:

```
/usr/local/dlxs/idx/p/pitttext_ssp/
```

But don't move them yet... All of the book's image files also have to be moved to their proper place, which is up one directory from where they were first placed:

```
/usr/local/images/hpbooks/
```

This should be done first. When the new books' images are in the proper place, you can use a test version of the `pitttext-idx` script to access the new index files and check that everything's working properly. So, change directories to:

```
/usr/local/dlxs/dlxs9/cgi-bin/
```

Look at the script named `pitttext-idx.test`. The variable called 'dataDict' should be pointing to the files in the `/usr/local/images/hpbooks/indextemp/` directory, rather than the `/usr/local/dlxs/idx/` directory. Do a search in the

full-text collection, and then change the script name in the URL to 'pitttext-idx.test'. This should allow you to search, view, etc. the "new" stuff, as well as what's already there. As mentioned before, now's the time to look around and make sure that things are o.k. When everything checks out to your satisfaction, it's time to make the switch.

To be really polite, do this switch at a time when few users might be using the site. Say...3:30AM on a weeknight...what--too early?!-- more realistically, it should be o.k. to do this first thing in the morning on any particular day.

Here are the steps to follow:

1) change the "live" cgi script

(`/usr/local/dlxs/dlxs9/cgi-bin/pitttext.idx.pl`) to point to the new dataDict in `/usr/local/images/hpbooks/indextemp/`

2) copy the contents of `indextemp/` to `/usr/local/dlxs/idx/p/pitttext_ssp/`

3) go into the `pitttext.dd` and `pitttext.extra.dd` files that are now in `pitttext_ssp/` and fix all of the absolute paths they contain (pointing to

`/usr/local/images/hpbooks/indextemp/`) so that they point to

`/usr/local/dlxs/idx/p/pitttext_ssp/` instead (lots of search and replace).

4) change the test script (`pitttext-idx.test`) to point to the `pitttext.dd`

in the `pitttext_ssp/` directory--now we are testing the new index in it's new home, to make sure all those paths were changed correctly in the previous step

5) when all looks satisfactory, point the "live" script

(`pitttext-idx.pl`) to point back to the new dataDict in its rightful place: `pitttext_ssp/`

## FINAL CLEAN UP

The script to display the books has a feature that lets one look at "newbooks". This relies on date information hard-coded into the script; and there are two places where the month and year of the addition (for display) are hard-coded, too. These have to be changed. Open the file:

```
/usr/local/dlxs/dlxs9/cgi-bin/pitttext-idx.pl
```

in a text editor. Search for the string 'newbooks'. You should find a place in the script that will create a search for the attribute

DATE.CREATED. Change the value of this attribute to the DATE.CREATED value that is in your shipment's SGML headers.

Next, look at the current online full-text collection. Choosing the 'browse' view, you should see an option to view the latest additions, along with the month and year of the addition. Back in the text editor, do a search for the month that you saw. This should bring you to the first of two places to update the month and year, if necessary. The next occurrence is right below the first.

Also, you probably want to add a note to the HP 'news' page, and possibly the front page for the full-text collection.