Using TEX for a Publications Database

Mimi Burbank and Donna Burnette
Supercomputer Computations Research Institute
Florida State University
Tallahassee, FL 32306-4052
Internet: `mimi@scri.fsu.edu; donna@scri.fsu.edu`

Abstract

This article explains the impetus and chronology associated with the use of `plain` TEX at the Supercomputer Computations Research Institute (SCRI) to produce multiple and varied reports, and the evolutionary process of the SCRI "publications database" into its current state. Why would one use TEX for a database? Our reasons were simple: all of our publications were done in TEX, and the versatility of TEX as a programming language made it ideal. We have evolved from using a single file using an `\halign` to a series of macros which utilize one input file to produce a wide array of output formats according to preset information/design criteria. The reporting process is now quite complex though text entry remains the same — inputting information into only one file!

Introduction

The Supercomputer Computations Research Institute (SCRI) is an interdisciplinary program set up to do research in computational science, train researchers from various academic disciplines and provide them access to a supercomputer. It is a cooperative venture between Florida State University (FSU), the U.S. Department of Energy's Office of Energy Research, the State of Florida, and several computer manufacturers. The SCRI base consists of a 50-plus member group of application research scientists, technicians, software specialists and support personnel. However, this group is supplemented by various numbers of long-term visitors from other universities, collaborating faculty members from other FSU departments and graduate and undergraduate students.

During 1985 – 1986 approximately 94 publications were produced using TEX, which upon looking back, were reminiscent of medieval times (scissors, glue, copiers, black spots, white-out, etc.). At this time, there was no resident TEXpert — only a single person trying to learn and use TEX. During 1986, a computer science student was hired part-time to assist in manuscript preparation, and at this time we became officially responsible for the reporting of publications at monthly and annual intervals. There was no suitable database on-line, and fiscal records were maintained in a variety of Macintosh-related utilities. Since all of our publications were done in TEX, it seemed time- and cost-effective to utilize the programming capabilities of TEX.

We shall try to explain the underlying TEX concepts involved in making our database work. While reading through the details of our setup, it is important to realize that this concept could easily be applied to any collection of information that is reiterated periodically with the only difficulty being the design of the report document itself and the ability to turn on/off the appropriate commands to feed it accurate information.

Pre-Database Publications

With very little experience using TEX, we started out with a three-column design for our publications report, using a variety of fonts to accentuate the different areas required — preprint number, month/year, title, author, and where submitted or published. An example of the code used at that time is:

```
%
\halign{%
  \hbox{\vtop{\hsize=2in\raggedright #}}
   \hfil \quad
   &  {\hbox{\hsize=1in#}}\hfil\quad
   & \hbox{\vtop{\noindent\hsize=4in
    \raggedright #}}  \hfil\cr
{\bf FSU-SCRI-85-01\break}
Bhanot, Duke \& Salvador
& 4/85
&FRACTALS AND INTERPOLATING
DIMENSIONS\cr
\noalign{\vskip1.3ex}
&& {\it Published Phys. Lett. B.},
Vol. 165B, 12/26/85,
pp. 355--360.}}\cr
\noalign{\vskip3ex}
{\bf FSU-SCRI-85-02\break}
Hasenfratz, A. and  P. Hasenfratz
& 4/85
& LATTICE GAUGE THEORIES\cr
\noalign{\vskip1.3ex}
&& {\it Published in Ann. Rev. Nucl.
Part. Sci., Vol. 35, 1985, pp. 559
--604.}\cr
}}
```

Modification of this data was an exercise in torture — readability was almost zero and time lost debugging forgotten ampersands, `\cr`'s, etc. was commonplace and costly. Excerpts from this file were then copied into other files to represent partial listings that met certain criteria. Our facility's publication rate quickly made manually sorting through this list according to *any* criteria a cumbersome task. We decided to delve into TEX's programming abilities and see if there was a way to automate this process. In 1987, we hired a programmer and officially began maintaining a "real" database, from which we ran monthly reports, fiscal reports and one cumulative report from 1985 onward.

Primitive Database Design

Having learned how to use temporary storage boxes (i.e., `\setbox`) we decided that this was a place to start our research into TEX's abilities/inabilities. We categorized all of our preprint information and put each item (author, title, subject, date, journal, volume, etc.) into its own box. A sample preprint entry at this stage of our database follows.

```
\num{01}
\date 1/92
\author{D. Burnette}
\title{Using \TeX\ for a
        Publications Database}
\journal{Submitted to TUGboat}
\endref

\num{02}
\date 1/92
\author{Unknown}
\title{This is a Junky Title}
\journal{Published in the Journal of
        Unknown Works}
\volume{5}
\page{1200--1204}
\endref
```

With the addition of a command to re-initialize the contents of the boxes between preprint entries, we had a legible list of preprints which we could output in its entirety at any time. Though there should be a less painstaking method to accomplish this task, the code/command we use for initialization (`\resetvars`) is defined below. The additional box `\dummy` is used as a mechanism for discarding information that has already been processed or is otherwise unused. The need for this command arose because of the occasional omission of a field by one of the many people entering information into the `publist`*nn*`.dbf` file, which allowed information from one entry to show up in the output of a consecutive item, and we preferred to have an empty field rather than an erroneous field.

```
\newbox\dummy
\def\resetvars{%
%       reset the ALL variables.
%
\ifvoid\refnum{}
\else\global\setbox\dummy=%
   \hbox{\unhbox\refnum}
\fi%
.
.
.
\ifvoid\refextra{}
\else\global\setbox\dummy=%
   \hbox{\unhbox\refextra}
\fi%
}%
%
```

Each database entry contains some combination of the above commands, based on the type of preprint it is and the information we want to maintain, but they are all delimited by the command `\endref` which does all of the work.

```
\global\def\endref{%
\hsize=7truein\parindent=0pt
  \baselineskip=12.4pt \parskip=6pt%
\if\currentptype\p
 \vtop{\line{\okbreak%
  \vtop{\hsize=2in\noindent\raggedright%
    \ifvoid\refnum{}%
    \else\bf FSU-SCRI-\unhbox\refyear%
    \unhbox\refatype-%
    \unhbox\refnum\break%
    \fi%
   \unhbox\refauthor\par
   \unhbox\refsubject}\hfill%
   \hbox to .5truein{\hfill
    \unhbox\refdate\hfill}%
    \hfill\vtop{\parindent=0pt
   \hsize=4in\raggedright%
    \ifvoid\reftitle{}%
     \else\unhbox\reftitle\par%
     \fi%
    \ifvoid\refjournal{%
     \ifvoid\refbook{}%
     \else In: \unhbox\refbook%
     \fi%
    \ifvoid\refeditor{}%
     \else,\ \unhbox\refeditor, ed.%
     \fi%
    \ifvoid\refeditors{}%
     \else,\ \unhbox\refeditors, eds.%
     \fi%
    \ifvoid\refvolume{}%
     \else,\ \unhbox\refvolume%
     \fi%
    \ifvoid\refpage{}%
     \else,\ \unhbox\refpage%
     \fi%
    \ifvoid\refpublisher{}%
     \else,\ (\unhbox\refpublisher%
     \fi%
    \ifvoid\refpubyear{).}%
     \else,\ \unhbox\refpubyear).%
     \fi%
     }%
 \else\unhbox\refstatus\unhbox\refjournal%
  \ifvoid\refvolume{}%
   \else,\ \unhbox\refvolume%
   \fi%
  \ifvoid\refpage{}%
   \else,\ \unhbox\refpage%
   \fi%
  \ifvoid\refpubyear{.}%
   \else\ (\unhbox\refpubyear).%
   \fi%
  \fi%
  \ifvoid\refextra{}%
   \else\ \unhbox\refextra.%
   \fi%
   }}%
\bigskip
}\fi%
}
```

As you can see, we have replaced our `\halign` scheme with `\hboxes` and `\vboxes` with defined dimensions.

Defining Our Criteria

Our next task was to examine each record (`\num`...`\endref`), set up logical collections of records, and then mark them in some way. In 1989 our funding agency had expressed an interest in a regular listing of our published papers as well as a periodic report of the status of the publications during the reporting period. From that suggestion, we decided that obtaining a listing of papers in any one of the various steps toward publication might also be beneficial. We found that there were three steps in the publication process (submitted, accepted, and published). We also found that we had papers that were submissions to conference proceedings, and another set that were technical reports but otherwise unpublished. We assigned each record a `\papertype` according to its category and chose the letters `\s, \a, \p, \c,` and `\t` as their designations.

Beyond this collection, we also found it helpful to know how much publishing each of our authors have done (individually and as a group). Due to the multidisciplinary/international nature of the institute we have authors whose publications appear in more than one subject area.

We have a group of scientists that more or less make up the core of SCRI. These scientists invite scientists in their field to collaborate with them on projects. Quite often these visitors publish papers on their collaboration while they are here or after they have left. These papers (`\v`) are added to our database.

There are projects off-site that request allocations of supercomputer time which are granted by our funding agency provided they agree to supply us with the publications generated by their results. Once a year we solicit the external users of the supercomputer for their publications and also add them (`\e`) to our database.

We found that assigning an `\authortype` to each preprint would enable us to generate a report of work done by any particular type of author should the need arise and chose the letters 'e', 'v', and none (the default) as their designations. The default `\authortype` is used for members of the core group of SCRI scientists.

Assigning the Criteria Commands

We created a set of commands to hold the characters which would represent each paper category (`\s, \a, \p, \z, \t, \e, \v`) and a second set of commands to associate those conditions with the record 'type' in the database (`\papertype` corresponded to `\currentptype`, and `\authortype` corresponded to `\currentatype`) for comparison at run-time. `\z` was substituted for `\c` (to avoid conflict with TEX's cedilla) for conferences papers.

```
%
\global\def\a{A}  % to appear
\global\def\e{E}  % external
\global\def\p{P}  % published
\global\def\s{S}  % submitted
\global\def\t{T}  % technical
\global\def\v{V}  % visitor
\global\def\z{C}  % conf. proc.
%
```

Further refinement resulted in the following definitions, which include the introductory remarks regarding publication status. This was done to maintain uniformity of entries.

```
\global\def\published{\papertype{P}%
   \global\setbox\refstatus=\hbox{\it
    Published in: }}
\global\def\submitted{\papertype{S}
   \global\setbox\refstatus=\hbox{\it
    Submitted to: }}
\global\def\accepted{\papertype{A}
   \global\setbox\refstatus=\hbox{\it
    To appear in: }}
\global\def\technical{\papertype{T}
    \tech{SCRI Technical Report}}
\global\def\conf{\papertype{C}}
.
.
```

Using the above samples, database entries quickly changed to:

```
% SCRI author / submitted paper
\num{01}
\date 1/92
\submitted
\author{D. Burnette}
\title{Using \TeX\ for a
        Publications Database}
\journal{Submitted to the TUGboat}
\endref

% SCRI visitor / published paper
\num{02}
\date 1/92
\visitor
\published
\author{Unknown}
\title{This is a Junky Title}
\journal{Published in the Journal of
        Unknown Works}
\volume{5}
\page{1200--1204}
\pubyear{1992}
\endref
```

```
% External Author / accepted paper
\num{03}
\date 1/92
\external
\accepted
\author{Unknown}
\title{Another Junky Title}
\journal{Published in the Journal of
        Something Else}
\endref
```

Finally, we incorporated conditional statements into our definition of `\endref` to test for various types of entries. It is important to note the conditional statement `\if\currentptype\p` takes the current value of `\currentptype` and compares it to the current value of `\p`. From the previous definitions, we know that `\p` should always be "`P`". The value of `\currentptype` is set in the record entry by `\papertype{<value>}`. If `<value>` = "`P`" also, then this conditional equates to `TRUE` and the various details about the preprint are unboxed according to the macro. Otherwise, the information in this record is not needed and no output is generated by this entry. It is fairly easy to see that if you include a conditional statement for each type of record that you need to search for and vary the boxes that are utilized according to the type of entry involved, overall control of the output generated is fairly simple to tailor to your individual needs. We have constantly rehashed our use of conditional statements to remove unnecessary comparisons and to cut down on the confusion caused by nesting too many conditional statements, each of which has to be evaluated anyway (tabbing can only do so much!).

Current Status

The refinement of the file has progressed to its present state in which the following information is maintained:

```
   \num{<sequential = {1,2,3,...,N}>}
   \date mo/yr
   \author{<required>}
   \external {or  \visitor}
   \submitted {or \accepted,
       \conf, \published, or \technical}
   \title{<required>}
   \volume{<if published>}
   \page{<if published>}
   \pubyear{<if published>}
   \extra{<optional comment>}
   \reprinttrue
   \journal-code <or \proc or \tech>
   \SUBJECT-AREA-CODE
   \AUTHOR1-CODE\AUTHOR2-CODE
\endref
```

Our database files are called `publist`*nn*`.dbf` with *nn* being the fiscal year — i.e., `publist92.dbf`. The `publist`*nn*`.dbf` file for the current fiscal year contains a list of `\<author code>`s and `\<subject code>`s for reference by the various people who are allowed to enter information into this file. The `\authorname` usually is the username of the author, and we classify publications into fourteen subject areas.

When a paper is published, `\submitted` is changed to `\published` and the `\volume{}`,`\page{}` and `\pubyear{}` information is added, and `\reprinttrue` is appended before the `\endref` to indicate we have received a published 'reprint' which corresponds to the reference information. If no reprint has been received, then `\reprintfalse` may be appended (this is also the default).

A list of `<journal code>`s and the journals they represent is maintained on 183 different journals. Some of the abbreviations were adopted from the Science Citation Index (1990) to ensure the uniqueness of our commands. Journal abbreviations that weren't found in this issue were created — there were quite a few in this category.

With the above "counters" we can generate reports by month, year, author, research area, and by individual journal, plus we can report how many have been published, how many were submitted, how many are 'to appear', how many technical reports (unpublished work), and published conference proceedings have been done by our researchers.

Lists. `allscrinames.list` is a list of all SCRI authors (i.e., those included in the annual report and proposals to date). This file notes whether each author is still here [`\localfalse` or `\localtrue` (which is the default)], and whether they are a member of the Lattice Gauge Theory Group [`\latticetrue` or `\latticefalse` (which is the default)]. Other helpful notations can be added as they arise. These conditions are noted above the author's name in the input file. Some examples would be:

```
\global\latticetrue
\name{Berg, Bernd A.}{berg.prep}
  (LGT Member, Local)
.
\global\localfalse
\name{Berger, Mordechai}{berger.prep}
 (Not LGT, Not Local)
.
\name{Burnette, Donna E.}{burnette.prep}
(Not LGT, Local)
```

The output file name is the same as the author code. Directing the output to the appropriate fiscal/super file is done during execution according to the user's interactive command.

`jnlabbrv.tex` and `emptyjnls.tex` contain the definitions necessary to create uniform output for each journal name. `jnlabbrv.tex` contains the actual meaning of the abbreviations.

`subjectabbrv.tex` contains the definitions necessary to create uniform output for each subject/discipline area.

Divider files. `publist.tex` is run interactively to separate the preprints by author or to generate a publications list since the beginning of SCRI, depending upon the option you enter at the beginning prompt. (See Appendix A for a definitive portion of the macro for this file, and Appendix B for an example of the interactive commands.) We simply write out to 13 files at a time, close them and then open another 13 files until the end of the run. At the termination of a run, you have either `\input` all of the database files and written out 178 author files, **or** you have `\input` 178 files and written out one file. Additional subsets can be created if the need arises and requires little effort (a change in the `\input` file) to produce the same information for a different period of time.

`subject.tex` does the actual division of files into the fourteen discipline areas.

`journals.tex` separates preprints by journal, regardless of what state they are in (published, accepted, submitted). `jnlspub.tex` separates only those preprints that have reached a published status. An entry identical to that used in `publist.tex` is required to accomplish the separation (substituting the journal code for the author code.

Report files. `monthly.tex` is used to run the various reports we do. An example of the interactive commands to generate reports is shown in Appendix C.

The author files all perform the same essential function, the only difference being in the usage of the conditions talked about earlier (`\localfalse` and `\latticetrue`). `authors-lattice.tex` extracts the preprints for the authors that are members of the LGT Group; `authors-local.tex` extracts the preprints for the authors that are still at SCRI; `authors-scriline.tex` extracts the preprints for the SCRI-funded faculty line authors; and `authors-superpub.tex` extracts the preprints for everybody since the beginning of SCRI — it also includes the preprints by external authors.

`subject-superpub.tex` reports the preprints by subject area since SCRI began.

Running `journals-superpub.tex` creates a report based on the files generated by `journals.tex` (everything since the beginning of SCRI regardless of its status).

Designing a User Friendly Interactive Environment for Generating Reports

One of our goals was to make using this setup as simple as possible. We wanted this information to be public and easy to access for those who might take an interest. We also didn't want everyone at SCRI to have to be a TEXpert to be able to benefit from our development. To that end, we have created an interactive menu which reads input from the screen (see Appendices A and B), initializes another series of conditionals, and generates the requested report based on user input.

We have evidently been successful, for some of our scientists are now using their `authorname.super` files to update their own *vitae*, as well as finding preprints in various subject areas. We recently began posting our lattice field theory publications to an on-line mail server.

Summary

We began using TEX to generate reports first in 1986, one year after first being introduced TEX. At that time, our scientists used `plain` TEX, or submitted handwritten copy; LATEX was not used much at the time, or we might have tried to use BIBTEX. We have learned much about the programming capabilities of `plain` TEX over the years, and though the present program might be much more sophisticated, we have been satisfied with its simplicity. This simplicity has allowed us to easily modify the programs, often on short notice; and it brings to mind a quote from *The TEXbook* (page 373):

> ... Always remember, however, that there's usually a simpler and better way to do something than the first way that pops into your head. You may not have to resort to any subterfuge at all, since TEX is able to do lots of things in a straightforward way. Try for simple solutions first.

One of the ever-present problems which confronts those of us in "production" is that we are not often given time to learn more, or simply to "play", and unless we are lucky enough to have programming support we are confined to "getting the job done", rather than exploring avenues of getting the job done easier, or prettier, or faster. For some reason, our scientists almost universally wait until the last minute and then send us a file that has a deadline of "a week ago".

In all, our programs total several thousand lines and represent some seven report-generating files, which input five definition or macro files, to generate up to 13 reports. (No easy process to explain!)

It takes a little over two hours of cpu time to run `publist.tex`, approximately 40 minutes to run `authors-superpub.tex`, approximately five minutes to run `subject.tex`, and approximately two minutes to run the monthly report files. In the world of today's workstations, this overhead is negligible.

Bibliography

Institute for Scientific Information. "Lists of Source Publications." pages 66-85 (arranged by ISI abbreviations), *Science Citation Index*, vol. 1, Philadelphia, PA, 19104: University Science Center, 1990. (Also pages 86 – 106, arranged by full title.)

Knuth, Donald E. *The TEXbook*. Reading, Mass.: Addison-Wesley, 1984.

A. publist.tex Example

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\input publist.macros
\input emptydefs
\input emptyjnls
\input jnlabbrv
\input subjectabbrv
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\immediate\write16{}
\immediate\write16{IS A PRINTOUT OF THE ENTIRE PUBLICATIONS LIST REQUIRED (Y/N)?}
\immediate\write16{}
\message{--> }}
.
.
.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\immediate\write16{}
\immediate\write16{Enter S if you want the SUPERPUB files separated}
\immediate\write16{Enter P if you want the file for the current fiscal year separated}
\immediate\write16{}
\message{--> }}
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\global\read-1 to \datain
.
.
.
%
\if\runtype\s%
\def\fdir{preprints:[superpub]}%
% \def\fdir{[.test]}
\def\fextension{.super}
\fi%
\if\runtype\p%
\def\fdir{preprints:[fiscal]}%
% \def\fdir{[.test]}
\def\fextension{.fiscal}
\fi%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\if\printout\y{\global\def\endref{%
\hsize=7truein\parindent=0pt\baselineskip=12.4pt \parskip=6pt%
%
\if\currentptype\a\vtop{\line{\okbreak%
    \vtop{\hsize=2in\noindent\raggedright%
    \ifvoid\refnum{}%
    \else\bf FSU-SCRI-\unhbox\refyear\unhbox\refatype-%
\unhbox\refnum\break%
    \fi%
    \unhbox\refauthor\par\unhbox\refsubject}\hfill%
\hbox to .5truein{\hfill\unhbox\refdate\hfill}%
    \hfill\vtop{\parindent=0pt\hsize=4in\raggedright%
\ifvoid\reftitle{}%
\else\unhbox\reftitle\par%
\fi%
    \ifvoid\refjournal{%
\ifvoid\refbook{}%
\else In: \unhbox\refbook%
\fi%
```

```
\ifvoid\refeditor{}%
\else,\ \unhbox\refeditor, ed.%
\fi%
\ifvoid\refeditors{}%
\else,\ \unhbox\refeditors, eds.%
\fi%
\ifvoid\refvolume{}%
\else,\ \unhbox\refvolume%
\fi%
\ifvoid\refpage{}%
\else,\ \unhbox\refpage%
\fi%
\ifvoid\refpublisher{}%
\else,\ (\unhbox\refpublisher%
\fi%
\ifvoid\refpubyear{).}%
\else,\ \unhbox\refpubyear).%
\fi%
}%
    \else\unhbox\refstatus\unhbox\refjournal%
    \ifvoid\refvolume{}%
    \else,\ \unhbox\refvolume%
    \fi%
    \ifvoid\refpage{}%
    \else,\ \unhbox\refpage%
    \fi%
    \ifvoid\refpubyear{.}%
    \else\ (\unhbox\refpubyear).%
    \fi%
    \fi%
    \ifvoid\refextra{}%
    \else\ \unhbox\refextra.%
    \fi%
    }}%
\bigskip
}\fi%
%
\if\currentptype\p\vtop{\line{\okbreak%
.
\if\currentptype\s\vtop{\line{\okbreak%
.
\if\currentptype\t\vtop{\line{\okbreak%
.
\if\currentptype\z\vtop{\line{\okbreak%
.
.
.
}\fi%
\resetvars}%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
.
.
.
\newcount\filezero \filezero=0
\newcount\fileone \fileone=1
\newcount\filetwo \filetwo=2
\newcount\filethree \filethree=3
\newcount\filefour \filefour=4
\newcount\filefive \filefive=5
```

```
\newcount\filesix \filesix=6
\newcount\fileseven \fileseven=7
\newcount\fileeight \fileeight=8
\newcount\filenine \filenine=9
\newcount\fileten \fileten=10
\newcount\fileeleven \fileeleven=11
\newcount\filetwelve \filetwelve=12
\newcount\filethirteen   \filethirteen=13

.
.
.
```

The above 13 counters are used, emptied and reused until all publications have been written out to individual files.

B. Interactive Commands to Divide Files

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This preamble appears at the beginning and
% explain how to input data interactively.
{\obeyspaces\immediate\write16{}
\immediate\write16{}
\immediate\write16{}
\immediate\write16{                       FLORIDA STATE UNIVERSITY}
\immediate\write16{             SUPERCOMPUTER COMPUTATIONS RESEARCH INSTITUTE}
\immediate\write16{}
\immediate\write16{            SEPARATOR OF INDIVIDUAL SCRI AUTHOR'S PUBLICATIONS
\immediate\write16{                     (USING PUBLIST85.DBF -- present)}
\immediate\write16{}
\immediate\write16{IS A PRINTOUT OF THE ENTIRE PUBLICATIONS LIST REQUIRED (Y/N)
\immediate\write16{}
\message{--> }}
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\global\read-1 to \datain
%
\def\stripspace#1 \next{#1}
\def\stripzero0#1\next{#1}
\edef\datain{\expandafter\stripspace\datain\next}% strip \datain's space
%
\def\next#1\endname{\uppercase{\global\def\printout{#1}}}
%
\expandafter\next\datain\endname   %make\datain uppercase
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
{\obeyspaces\immediate\write16{}
\immediate\write16{Enter S if you want the SUPERPUB files separated}
\immediate\write16{Enter P if you want the file for the current fiscal year sep
\immediate\write16{}
\message{--> }}
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\global\read-1 to \datain
%
\def\stripspace#1 \next{#1}
\def\stripzero0#1\next{#1}
\edef\datain{\expandafter\stripspace\datain\next}% strip \datain's space
%
```

```
\def\next#1\endname{\uppercase{\global\def\runtype{#1}}}
%
\expandafter\next\datain\endname   %make\datain uppercase
%
\if\runtype\s%
        \def\fdir{preprints:[superpub]}%
%       \def\fdir{[.test]}
        \def\fextension{.super}
\fi%
\if\runtype\p%
        \def\fdir{preprints:[fiscal]}%
%       \def\fdir{[.test]}
        \def\fextension{.fiscal}
\fi%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

C. Interactive Commands to Generate Reports

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This preamble appears at the beginning and
% explain how to input data interactively.
{\obeyspaces\immediate\write16{}
\immediate\write16{}
\immediate\write16{}
\immediate\write16{                        FLORIDA STATE UNIVERSITY}
\immediate\write16{              SUPERCOMPUTER COMPUTATIONS RESEARCH INSTITUTE}
\immediate\write16{}
\immediate\write16{                 INTERACTIVE PUBLICATIONS LIST GENERATOR}
\immediate\write16{}
\immediate\write16{}
\immediate\write16{ENTER "Q" FOR A MONTHLY REPORT}
\immediate\write16{ENTER "R" FOR A COMPLETE PUBLICATIONS LIST}
\immediate\write16{ENTER "Y" FOR A COMPLETE LISTING OF ABSTRACTS}
\immediate\write16{}
\immediate\write16{ENTER "A" FOR A LIST OF PUBLICATIONS WHICH ARE "TO APPEAR"}
\immediate\write16{ENTER "C" FOR A LIST OF CONFERENCE PROCEEDINGS}
\immediate\write16{ENTER "E" FOR A LIST OF EXTERNAL PUBLICATIONS}
\immediate\write16{ENTER "M" FOR A LIST OF MISCELLANEOUS PUBLICATIONS}
\immediate\write16{ENTER "P" FOR A LIST OF PUBLISHED PUBLICATIONS}
\immediate\write16{ENTER "S" FOR A LIST OF SUBMITTED PUBLICATIONS}
\immediate\write16{ENTER "T" FOR A LIST OF TECHNICAL REPORTS}
\immediate\write16{ENTER "V" FOR A LIST OF VISITORS PUBLICATIONS}
\immediate\write16{}
\message{--> }}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```