FreeHEP sections

The software packages within FreeHEP are divided into sections, with an editor (or editors) for each section. Choose one of the following sections to see a list of all the packages in that section. Note that some packages may appear in more than one section.

- Analysis and Data Reduction
- Artificial Intelligence
- CAD/CAE Systems
- Computer Hardware
- C++ (See also FreeHEP C++ reviews or C++ Virtual Library.)
- Data Acquisition
- Data bases and file systems
- Data Modeling and I/O
- Detector Simulation
- Event Generators
- General Libraries
- Graphics, visualization, GUIs
- HEP Theory
- Languages
- Lattice field theory (See also Lattice High Energy Physics overview.)
- Mathematics
- Networking/Electronic mail/News
- Operating Systems
- Parallelism and Distributed Computing
- Particle Properties
- Software Engineering

It is also possible to select packages from an alphabetical list of all packages, or to perform a database search to track down the package you are interested in.

FreeHEP Home Page

Submitting new software to FreeHEP

We are always looking for new software packages to add to the FreeHEP database. The only requirement for new items is that they be generally useful to the High Energy Physics community. If have, or know of, any software that you think meets this criterium please let us know. In the first instance you should contact the editor for the subject area in which you think the package belongs. If you are unsure what area is appropriate feel free to contact the managing editor for guidance.

All software packages in FreeHEP are described by a .dbase file which resides on the FreeHEP anonymnous FTP machine. A good way to submit new software is to create a .dbase file for the package and send it to the relevant editor. Once a new .dbase file is installed on the FreeHEP machine the spires and WWW databases will be automatically updated (after about a day).

If you would like to place source code or documentation on the FreeHEP machine to make it available by anonymous FTP to others then you should contact the FreeHEP managing editor directly.

Further Information about FreeHEP

The idea of setting up a library of useful and easily accessible HEP software was first proposed at the HEPLIB meeting at the SSC Lab in September 1991 and again at La Londe in January 1992. At that meeting it was generally recognized that this was a worthwhile idea that should be actively pursued.

Benefits

There are many potential benefits of FreeHEP both to the HEP user community and to software writers. Users benefit by gaining knowledge of existing software, by gaining easy access to the software they want, by gaining from the experience of other users and by having easy access to authors so that bugs and other problems can be fixed quickly. Authors benefit by gaining a mechanism for distributing their software, by avoiding duplication of efforts, by getting bug reports and suggestions from users and by making contacts with potential collaborators. Since FreeHEP is meant to be an inclusive service to authors as well as to the HEP user community, there is no requirement on the form of software distribution and we leave it up to the authors to distribute their packages in whatever form is most convenient.

Organization

FreeHEP currently consists of a database of useful software, accessible using WWW, Spires, or directly from the FreeHEP anonymous FTP site. The anonymous FTP site also contains areas for reviews of software packages, and in some cases the actual software itself. Software packages are organized into subject areas, with one or more editors for each section, as well as a managing editor.

We also plan to set up News Groups for different subject areas and to publish some form of (electronic?) newsletter listing new packages and other topical information.

We encourage anyone who has, or knows of, software that they believe should be included in the FreeHEP database to let us know.

HEPLIB

The first HEPLIB user's meeting was held at the SSC Laboratory, Dallas, Texas, September 19-20, 1991. Fifty-four scientists from thirty-one High Energy Physics research institutes and universities met for two full days to discuss the support and environments of High Energy Physics computing and to form and define the scope of a HEPLIB Users Group.

Initial Objectives

There was a general consensus for the following objectives:

HEPLIB should be world users group for enhancements, communications, and distribution of software in the HEP computing environments.

- HEPLIB will collect, maintain, document, and distribute shareable application software for HEP computing as well as non-HEP applications, including code management systems in heterogeneous environments, data base systems, and automatic installation and test procedures. (See FreeHEP).
- HEPLIB should promote and recommend industry, as well as HEP computing standards as appropriate, including operating systems, distributed computing environments, quality assurance, version control, and information exchange.

A steering committee was formed to plan subgroups, initiate exchange and communication, plan logistics, arrange for meetings, begin planning for a HEPLIB Newsletter, and look into questions of manpower and funding for the HEPLIB Users Group.

Progress

The current status of HEPLIB is summarized in three working documents:

Note 92-02
Summary - The HEPLIB'92/KEK International Users Meeting
Note 92-03
HEPLIB - Consensus and Objectives
Note 92-04
Standards for Certified Software in HEPLIB

Contacts

Initially, the following addresses may be used to contact the HEPLIB User Group :

Miguel Marquina
CN Division
CERN
CH-1211 Geneve 23
Switzerland

Tel (022) 767-4912
Fax (022) 767-7155

Marquina@CERNVM

Herald Johnstad
SSC Laboratory MS-2001
2550 Beckleyemeade Avenue
Dallas, Texas 75237
U.S.A.

Tel (214) 708-6000
Tel (214) 708-0006

Johnstad@SSCVX1.SSC.GOV

Using Spires to Search FreeHEP

Enter SPIRES Search Command:

You can use Spires to search for specific packages in FreeHEP. You can issue search commands by typing commands of the following form as "keywords".

- AUTHOR Youssef, Saul
- AUTHOR Rene Brun
- AUTHOR T. Burnett
- SECTION Analysis
- DATE 1992
- DATE March 1992
- TITLE motif
- ABSTRACT radiative

Note that when searching for titles or abstracts any package that contains the specified word in the title or abstract will match. You can also form compound searches using the word "AND", for example:

- AUTHOR Youssef, Saul AND DATE 1992
- AUTHOR Johnson AND TITLE Motif

Commands are not case sensitive. You can also get a complete list of spires keywords(AUTHOR, SECTION, DATE etc.) available for use with FreeHEP, browse an alphabetical list of all packages or search for packages by subject area.

FreeHEP Home Page

Further Information about FreeHEP

The idea of setting up a library of useful and easily accessible HEP software was first proposed at the HEPLIB meeting at the SSC Lab in September 1991 and again at La Londe in January 1992. At that meeting it was generally recognized that this was a worthwhile idea that should be actively pursued.

Benefits

There are many potential benefits of FreeHEP both to the HEP user community and to software writers. Users benefit by gaining knowledge of existing software, by gaining easy access to the software they want, by gaining from the experience of other users and by having easy access to authors so that bugs and other problems can be fixed quickly. Authors benefit by gaining a mechanism for distributing their software, by avoiding duplication of efforts, by getting bug reports and suggestions from users and by making contacts with potential collaborators. Since FreeHEP is meant to be an inclusive service to authors as well as to the HEP user community, there is no requirement on the form of software distribution and we leave it up to the authors to distribute their packages in whatever form is most convenient.

Organization

FreeHEP currently consists of a database of useful software, accessible using WWW, Spires, or directly from the FreeHEP anonymous FTP site. The anonymous FTP site also contains areas for reviews of software packages, and in some cases the actual software itself. Software packages are organized into subject areas, with one or more editors for each section, as well as a managing editor.

We also plan to set up News Groups for different subject areas and to publish some form of (electronic?) newsletter listing new packages and other topical information.

We encourage anyone who has, or knows of, software that they believe should be included in the FreeHEP database to let us know.

Example FreeHEP .dbase file

The following is an example of a FreeHEP .dbase file. These simple text files are kept on the FreeHEP anonymous FTP machine and provide information on each of tha packages in FreeHEP. These files are also imported into the Spires and WWW databases daily.

```
                Name: HippoPlotamus
             Version: 1.10
                Date: May, 1992
               Title: A package for 'tuple viewing and manipulation
          Authors(s): Mike Gravina(SLAC,mfg@ebnextk.slac.stanford.edu)
                      Paul Kunz(SLAC,pfkeb@kaon.slac.stanford.edu)
                      Paul Rensing(SLAC,rensing@unixhub.slac.stanford.edu)
             Contact: pfkeb@kaon.slac.stanford.edu
     Subject Area(s): graphics_vis_gui, analysis
 News Group or Email: hippo_comment@ebnextk.slac.stanford.edu (e-mail)
      Bug reports to: hippo_bug@ebnextk.slac.stanford.edu (e-mail)
     Software Needed: XDR ANSI-C
     Hardware Needed: A computer running VM, VMS or UNIX
              Access: anonymous ftp from heplib.slac.stanford.edu
           User Base:
       Documentation: Included in TAR file
Published References: Proceedings of L'Agelonde workshop
            See Also: HippoDraw
            Abstract: Hippoplotamous is a n-tuple management and display
                      package written in ANSI C with an object orientation.
                      The management part is designed to be user friendly and
                      also has a FORTRAN binding.  Binary files use the XDR
                      format so binary ftp can be done between machines of
                      different architectures.   Files can also converted
                      from or to a plain text format and from HBOOK4 format
                      with supplied utilities.

                      The display package can produce histograms, scatter
                      plots, grey or color density plots, and x-y plots.  It
                      is designed to be friendly to one who implements an
                      interactive application for visualizing the n-tuple
                      data.   Drivers for Display Postscript, X11,
                      InterViews, UNIXPlot, line printer and PostScript
                      printer are supplied.

                      Hippoplotamus has been tested on NeXT, SUN, RS/6000,
                      Ultrix, SGI, VAX/VMS, and VM/CMS.
```

Example FreeHEP .dbase file

The following is an example of a FreeHEP .dbase file. These simple text files are kept on the FreeHEP anonymous FTP machine and provide information on each of tha packages in FreeHEP. These files are also imported into the Spires and WWW databases daily.

```
                  Name: HippoPlotamus
               Version: 1.10
                  Date: May, 1992
                 Title: A package for 'tuple viewing and manipulation
            Authors(s): Mike Gravina(SLAC,mfg@ebnextk.slac.stanford.edu)
                        Paul Kunz(SLAC,pfkeb@kaon.slac.stanford.edu)
                        Paul Rensing(SLAC,rensing@unixhub.slac.stanford.edu)
               Contact: pfkeb@kaon.slac.stanford.edu
       Subject Area(s): graphics_vis_gui, analysis
    News Group or Email: hippo_comment@ebnextk.slac.stanford.edu (e-mail)
        Bug reports to: hippo_bug@ebnextk.slac.stanford.edu (e-mail)
       Software Needed: XDR ANSI-C
       Hardware Needed: A computer running VM, VMS or UNIX
                Access: anonymous ftp from heplib.slac.stanford.edu
             User Base:
         Documentation: Included in TAR file
   Published References: Proceedings of L'Agelonde workshop
              See Also: HippoDraw
              Abstract: Hippoplotamous is a n-tuple management and display
                        package written in ANSI C with an object orientation.
                        The management part is designed to be user friendly and
                        also has a FORTRAN binding.  Binary files use the XDR
                        format so binary ftp can be done between machines of
                        different architectures.   Files can also converted
                        from or to a plain text format and from HBOOK4 format
                        with supplied utilities.

                        The display package can produce histograms, scatter
                        plots, grey or color density plots, and x-y plots.  It
                        is designed to be friendly to one who implements an
                        interactive application for visualizing the n-tuple
                        data.   Drivers for Display Postscript, X11,
                        InterViews, UNIXPlot, line printer and PostScript
                        printer are supplied.

                        Hippoplotamus has been tested on NeXT, SUN, RS/6000,
                        Ultrix, SGI, VAX/VMS, and VM/CMS.
```

FreeHEP editors

Mark Edel, FNAL
    Graphics, visualization, GUIs (edel@fnal.gov)
Tony Gabriel, ORNL
    Detector Simulation (tag@ornl.gov)
Irwin Gaines, FNAL
    Data Acquisition (gaines@fnal.gov)
Lynn Garren, FNAL
    Detector Simulation (garren@fnal.gov)
Tom Handler, U.Tennessee
    Detector Simulation (th@utkhep.phys.utk.edu)
Tony Johnson, Boston Univ./SLAC
    Analysis and Data Reduction and compilation (tonyj@slacvm)
Harald Johnstad, SSCL
    Event Generators and Software Engineering (sscvx1::johnstad)
Werner Koellner, LBL
    General Libraries (csa::koellner)
Paul LeBrun, FNAL
    Graphics, visualization, GUIs (lebrun@fnal.gov)
Youhei Morita, KEK
    Parallelism and Distributed Computing (kekvax::morita)
Miguel Marquina, CERN
    Software Engineering (marquina@cernvm.cern.ch)
Andrea Palounek, LANL
    Compilation (vaxlnf::pace)
Jamie Shiers, CERN
    Data bases and file systems (jamie@cernvm.cern.ch)
John Womersley, FSU/SCRI
    CAD/CAE Systems (fsuhep::womersley)
Saul Youssef, SCRI
    (managing editor) (youssef@scri.fsu.edu)

Important Note

The programs and information provided by FreeHEP are offered in the hope that they will be of benefit to the HEP community. However neither the editors, nor any one else associated with the project can guarantee the accuracy of any information provided.

THE PROGRAMS AND DOCUMENTS IN FREEHEP ARE OFFERED WITH NO WARRANTY OF ANY KIND.

FreeHEP editors

Mark Edel, FNAL
Graphics, visualization, GUIs (edel@fnal.gov)
Tony Gabriel, ORNL
Detector Simulation (tag@ornl.gov)
Irwin Gaines, FNAL
Data Acquisition (gaines@fnal.gov)
Lynn Garren, FNAL
Detector Simulation (garren@fnal.gov)
Tom Handler, U.Tennessee
Detector Simulation (th@utkhep.phys.utk.edu)
Tony Johnson, Boston Univ./SLAC
Analysis and Data Reduction and compilation (tonyj@slacvm)
Harald Johnstad, SSCL
Event Generators and Software Engineering (sscvx1::johnstad)
Werner Koellner, LBL
General Libraries (csa::koellner)
Paul LeBrun, FNAL
Graphics, visualization, GUIs (lebrun@fnal.gov)
Youhei Morita, KEK
Parallelism and Distributed Computing (kekvax::morita)
Miguel Marquina, CERN
Software Engineering (marquina@cernvm.cern.ch)
Andrea Palounek, LANL
Compilation (vaxlnf::pace)
Jamie Shiers, CERN
Data bases and file systems (jamie@cernvm.cern.ch)
John Womersley, FSU/SCRI
CAD/CAE Systems (fsuhep::womersley)
Saul Youssef, SCRI
(managing editor) (youssef@scri.fsu.edu)

Important Note

The programs and information provided by FreeHEP are offered in the hope that they will be of benefit to the HEP community. However neither the editors, nor any one else associated with the project can guarantee the accuracy of any information provided.

THE PROGRAMS AND DOCUMENTS IN FREEHEP ARE OFFERED WITH NO WARRANTY OF ANY KIND.

How To use Anonymous FTP

If you are new to unix, you may not be familiar with ftp or copying files. On most machines, once you reach freehep via ftp you will see a prompt like this:

```
ftp>
```

From here, you can navigate the directory structure with % ls and % cd. To copy an ascii file to your home machine, do

```
ftp> get filename   (case sensitive)
```

You may also see files with extensions ".tar", ".Z", or typically, both. The ".Z" indicates a compressed binary file which can be fetched like so:

```
ftp> binary
ftp> get xxxx.Z
```

On your home unix machine, you can uncompress it with the command

```
% uncompress xxxx.Z
```

which produces the file "xxxx".

Files with the extension ".tar" are also binary files containing a packed collection of files possibly including subdirectories. To unpack such a file, do

```
% tar xvf xxxx.tar
```

or see the tar command on your home unix machine.

You may also see files with a ".pac.Z" extension. These files can be unpacked with tar xvf and with the "% dupackag" command assuming that the "TYPES" package is installed.

How To use Anonymous FTP

If you are new to unix, you may not be familiar with ftp or copying files. On most machines, once you reach freehep via ftp you will see a prompt like this:

```
ftp>
```

From here, you can navigate the directory structure with % ls and % cd. To copy an ascii file to your home machine, do

```
ftp> get filename   (case sensitive)
```

You may also see files with extensions ".tar", ".Z", or typically, both. The ".Z" indicates a compressed binary file which can be fetched like so:

```
ftp> binary
ftp> get xxxx.Z
```

On your home unix machine, you can uncompress it with the command

```
  % uncompress xxxx.Z
```

which produces the file "xxxx".

Files with the extension ".tar" are also binary files containing a packed collection of files possibly including subdirectories. To unpack such a file, do

```
  % tar xvf xxxx.tar
```

or see the tar command on your home unix machine.

You may also see files with a ".pac.Z" extension. These files can be unpacked with tar xvf and with the "% dupackag" command assuming that the "TYPES" package is installed.

```
/*****************************************************************/
/*                                                               */
/* FHGET EXEC - Special processing of files for freehep          */
/*                                                               */
/* TonyJ, May 1992                                               */
/*                                                               */
/*****************************************************************/

   Parse upper arg ip_adr file '(' options

   Say time() 'FHGET received:' ip_adr file '(' options

   Parse Var file subfile'/'rest
   p = LastPos('/',rest)
   mode = Substr(rest,p+1)
   if p>0 Then criteria = Substr(rest,1,p-1)
          else criteria = ''
   temp = criteria
   n=0
   Do While temp/=''
     n = n+1
     Parse Var temp Index.n'/'Value.n'/'temp
     If Value.n='*' Then Value.n='>a'
     End

   NCriteria = n
   SpiresTerm = Index.1 Decode(Value.1)
   Do i=2 to NCriteria
     SpiresTerm = SpiresTerm 'AND' Index.i Decode(Value.i)
     End

Restart:

   If mode='SHOWIND' Then Do

     'EXEC QSPIRES SHOW IND ( STACK NOSTAR IN' subfile
     j=0;
     Do Queued()
       Parse Pull . 'Index:' term '('quals')'.
       If term='' Then Iterate
       j = j+1
       Term.j = Strip(term)
       End

     Queue '<TITLE>Index keywords available for FreeHEP</TITLE>'
     Queue '<H2>Index keywords available for FreeHEP</H2>'

     Queue 'Note: Commas separate synonyms.'
     Queue '<UL>'
/*
 *   We do not include indexes with no synonyms (for FreeHEP)
 */
     Do i=1 to j
       p = LastPos(',',Term.i)
       if p=0 Then Iterate
       term = Strip(SubStr(Term.i,p+1))
       Queue '<LI>'
       Queue '<A HREF=/FIND/'subfile'/'term'//BROWSE>'
       Queue Term.i'</A>'
       End
   Queue '</UL>'
   End

 Else If mode='BROWSE' Then Do

   n = nCriteria
   'EXEC QSPIRES BROWSE' Index.n Value.n '( STACK NOSTAR IN' subfile
   j = Queued()
   Do i=1 to j
     Parse Pull Term.i
     Term.i = Strip(Term.i)
     End

   'EXEC QSPIRES SHOW ELEM DESC' Index.n '(STACK NOSTAR IN' subfile
   Pull .
   Pull .
   jj = Queued()
   Do i=1 to jj
     Parse Pull Desc.i
     Desc.i = Strip(Desc.i)
     End

   Queue '<TITLE>Browse' Index.n Decode(Value.n) 'for FreeHEP</TITLE>'
   Queue '<H2>Browse' Index.n Decode(Value.n) 'for FreeHEP</H2>'
   Queue '<IsIndex>'
   Queue '<H3>Description</H3>'
   Do i=1 to jj
     Queue Desc.i
     End
   Queue '<H3>Typical values</H3>'
   Queue '<UL>'
   Do i=1 to j
     Queue '<LI>'
     Queue '<A HREF=/FIND/'subfile'/'Index.n'/'Encode(Term.i)'/RESULT>'
     Queue Term.i'</A>'
     End
   Queue '</UL>'
   Queue 'Choose one of the above or type a new value.'
   End

 Else If mode='INDEX' Then Do

   'EXEC QSPIRES FIND' SpiresTerm  '( STACK NOSTAR IN' subfile 'BRIEF'

   i = 0;
   Do Queued()
     Parse Pull Line
     if Substr(Line,1,10)='No records' Then Do
       Do Queued()
         Pull.
         End
       Mode = 'RESULT'
       Signal Restart
       End

     Parse Var Line Title":" Name
     Title = Strip(Title)
     Name = Strip(Name)
     If Title = 'Freehep Name' Then Do
       i=i+1
```

```
      Name.i  = Name
      Title.i  = ""
      End
    Else If Title = 'Title' Then Title.i = Name
    Else If Title.i = '' Then Name.i = Name.i Strip(Line)
                         Else Title.i = Title.i Strip(Line)
    End

  Queue '<DL>'
  Do j=1 to i
    Queue '<DT><A HREF=/FIND/FREEHEP/NAME/'Encode(Name.j)'/FULL>'
    Queue Name.j'</A>'
    If Title.J /= "" Then Queue '<DD>'Title.j
    End
  Queue '</DL>'
  End

Else If mode='RESULT' | mode='NARROW' Then Do

  'EXEC QSPIRES FIND' SpiresTerm  '( STACK NOSTAR RESULT IN' subfile
  Parse Pull Line
  If Line = 'Invalid index term' Then Do
    If NCriteria=1 Then Do
      Queue 'Your index keyword ('Index.1') is invalid.'
      End
    Else Do
      Queue 'One of your index keywords ('Index.1
      Do i=2 to NCriteria-1
        Queue ',' Index.i
        End
      Queue 'or' Index.NCriteria') is invalid.'
      End
    Queue 'You can obtain a list of'
    Queue '<A HREF=/FIND/FREEHEP/SHOWIND>valid keywords</A> or'
    Queue '<A HREF=/FIND/FHSPIRES.HTML>start a new search</A>.'
    End
  Else Do
    Parse Var Line 'Result' N .

    if NCriteria=1 Then latin = "criterium"
                   Else latin = "criteria"

    if NCriteria=1 Then are = "is"
                   Else are = "are"

    Queue '<P>'
    Queue 'Your current search' latin are':'
    Queue '<UL>'
    Do i=1 to NCriteria
      Queue '<LI>' Index.i Decode(Value.i)
      End
    Queue '</UL>'
    Queue '<P>'

    If n=''     Then Queue 'No packages matched your search' latin'.'
    Else If n=1 Then Queue '1 package matched your search' latin'.'
                Else Queue N 'packages matched your search' latin'.'

    If Mode='RESULT' Then Do
      Queue '<P>You may now'
      if n=1 Then Do
        Queue '<A HREF=/FIND/FREEHEP/'criteria'/FULL>'
        Queue 'examine the entry that matched your' latin'</A>,'
        End
      Else if n>1 Then Do
        Queue '<A HREF=/FIND/FREEHEP/'criteria'/INDEX>'
        Queue 'examine a list of items that matched your' latin'</A>,'
        Queue 'continue to narrow down your search by specifying'
        Queue '<A HREF=/FIND/FREEHEP/'criteria'/NARROW>'
        Queue 'further criteria</A>,'
        End
      Else Do
        Queue '<A HREF=/FIND/FREEHEP/'criteria'/BROWSE>'
        Queue 'browse a list of values</A> that almost matched your'
        Queue 'last criterium,'
        End
      Queue 'or <A HREF=/FIND/FHSPIRES.HTML>start a new search</A>.'
      End
    Else Do /* NARROW */
      Queue '<P>'
      'EXECIO * DISKR FHNARROW HTML ( FINI'
      End
    End
  End
Else Do

  'EXEC QSPIRES FIND' SpiresTerm '( STACK NOSTAR IN' subfile

  Ref = ''
  Title = ''
  i=0
  Do Queued()
    Parse Pull Line
    if Substr(Line,1,10)='No records' Then Do
      Do Queued()
        Pull.
        End
      Mode = 'RESULT'
      Signal Restart
      End

    Parse Var Line . 'Freehep Name:' Name

    if Name /= '' Then Title = Strip(Name)

    Parse Var Line . 'See Also:' Refs
    if Refs = '' Then Do
      i = i+1
      Line.i = Line
      End
    Else Ref = Refs
    End

  If Title/='' Then Do
    Queue '<TITLE>'Title'</TITLE>'
    Queue '<H1>'Title'</H1>'
    End

  Queue '<XMP>'
  Do j=1 to i
```

```
   Queue Line.j
   End
  Queue '
```

' if Ref/=" Then Queue 'See also' Do While Ref/=" Parse Var Ref R','Ref R = Strip(R) if Ref=" Then Punc='.' Else Punc=',' Queue "R"Punc End End Return Queued() /* Spires search term values may have spaces in them, but WWW filespecs cannot, so here we encode names. */ Encode: Procedure Parse Arg String Bad = '% ()?/' Good = '%BOCQS' Out = " Do I=1 to Length(String) c = Substr(String,i,1) if Index (bad,c)/=0 Then Out = Out||'%'Translate(c,good,bad) Else Out = Out||c End Return Out Decode: Procedure Parse Arg String Bad = '% ()?/' Good = '%BOCQS' Out = " Esc = 0 Do I=1 to Length(String) c = Substr(String,i,1) if Esc Then Do Out = Out||Translate(c,bad,good) Esc = 0 End Else if c = '%' Then Esc = 1 Else Out = Out||c End Return Out

```
 /****************************************************************/
 /*                                                              */
 /* FHGET EXEC - Special processing of files for freehep         */
 /*                                                              */
 /* TonyJ, May 1992                                              */
 /*                                                              */
 /****************************************************************/

   Parse upper arg ip_adr file '(' options

   Say time() 'FHGET received:' ip_adr file '(' options

   Parse Var file subfile'/'rest
   p = LastPos('/',rest)
   mode = Substr(rest,p+1)
   if p>0 Then criteria = Substr(rest,1,p-1)
          else criteria = ''
   temp = criteria
   n=0
   Do While temp/=''
     n = n+1
     Parse Var temp Index.n'/'Value.n'/'temp
     If Value.n='*' Then Value.n='>a'
     End

   NCriteria = n
   SpiresTerm = Index.1 Decode(Value.1)
   Do i=2 to NCriteria
     SpiresTerm = SpiresTerm 'AND' Index.i Decode(Value.i)
     End

Restart:

   If mode='SHOWIND' Then Do

     'EXEC QSPIRES SHOW IND ( STACK NOSTAR IN' subfile
     j=0;
     Do Queued()
       Parse Pull . 'Index:' term '('quals')'.
       If term='' Then Iterate
       j = j+1
       Term.j = Strip(term)
       End

     Queue '<TITLE>Index keywords available for FreeHEP</TITLE>'
     Queue '<H2>Index keywords available for FreeHEP</H2>'

     Queue 'Note: Commas separate synonyms.'
/*
 *   We do not include indexes with no synonyms (for FreeHEP)
 */
     Do i=1 to j
       p = LastPos(',',Term.i)
       if p=0 Then Iterate
       term = Strip(SubStr(Term.i,p+1))
       If i=1 Then Prefix = '<UL>'
              Else Prefix = '<LI>'
       Queue Prefix'<A HREF=/FIND/'subfile'/'term'//BROWSE>'
       Queue Term.i'</A>'
       End
```

```
    Queue '</UL>'
    End

  Else If mode='BROWSE' Then Do

    n = nCriteria
    'EXEC QSPIRES BROWSE' Index.n Value.n '( STACK NOSTAR IN' subfile
    j = Queued()
    Do i=1 to j
      Parse Pull Term.i
      Term.i = Strip(Term.i)
      End

    'EXEC QSPIRES SHOW ELEM DESC' Index.n '(STACK NOSTAR IN' subfile
    Pull .
    Pull .
    jj = Queued()
    Do i=1 to jj
      Parse Pull Desc.i
      Desc.i = Strip(Desc.i)
      End

    Queue '<TITLE>Browse' Index.n Decode(Value.n) 'for FreeHEP</TITLE>'
    Queue '<H2>Browse' Index.n Decode(Value.n) 'for FreeHEP</H2>'
    Queue '<IsIndex>'
    Queue '<H3>Description</H3>'
    Do i=1 to jj
      Queue Desc.i
      End
    Queue '<H3>Typical values</H3>'
    Do i=1 to j
      If i=1 Then Prefix = '<UL>'
             Else Prefix = '<LI>'
      Queue Prefix'<A HREF=/FIND/'subfile'/'Index.n'/'Encode(Term.i)'/RESULT>'
      Queue Term.i'</A>'
      End
    Queue '</UL>'
    Queue 'Choose one of the above or type a new value.'
    End

  Else If mode='INDEX' Then Do

    'EXEC QSPIRES FIND' SpiresTerm  '( STACK NOSTAR IN' subfile 'BRIEF'

    i = 0;
    Do Queued()
      Parse Pull Line
      if Substr(Line,1,10)='No records' Then Do
        Do Queued()
          Pull.
          End
        Mode = 'RESULT'
        Signal Restart
        End

      Parse Var Line Title":" Name
      Title = Strip(Title)
      Name = Strip(Name)
      If Title = 'Freehep Name' Then Do
        i=i+1
```

```
          Name.i  = Name
          Title.i  = ""
          End
        Else If Title = 'Title' Then Title.i = Name
        Else If Title.i = '' Then Name.i = Name.i Strip(Line)
                             Else Title.i = Title.i Strip(Line)
        End

      Queue '<TITLE>List of FreeHEP packages</TITLE>'
      Queue '<H1>List of FreeHEP packages</H1>'
      Queue '<DL>'
      Do j=1 to i
        Queue '<DT><A HREF=/FIND/FREEHEP/NAME/'Encode(Name.j)'/FULL>'
        Queue Name.j'</A>'
        If Title.J /= "" Then Queue '<DD>'Title.j
        End
      Queue '</DL>'
      End

    Else If mode='RESULT' | mode='NARROW' Then Do

      'EXEC QSPIRES FIND' SpiresTerm  '( STACK NOSTAR RESULT IN' subfile
      Parse Pull Line
      If Line = 'Invalid index term' Then Do
        If NCriteria=1 Then Do
          Queue 'Your index keyword ('Index.1') is invalid.'
          End
        Else Do
          Queue 'One of your index keywords ('Index.1
          Do i=2 to NCriteria-1
            Queue ',' Index.i
            End
          Queue 'or' Index.NCriteria') is invalid.'
          End
        Queue 'You can obtain a list of'
        Queue '<A HREF=/FIND/FREEHEP/SHOWIND>valid keywords</A> or'
        Queue '<A HREF=/FIND/FHSPIRES.HTML>start a new search</A>.'
        End
      Else Do
        Parse Var Line 'Result' N .

        if NCriteria=1 Then latin = "criterium"
                       Else latin = "criteria"

        if NCriteria=1 Then are = "is"
                       Else are = "are"

        Queue '<P>'
        Queue 'Your current search' latin are':'
        Do i=1 to NCriteria
          If i=1 Then Prefix = '<UL>'
                 Else Prefix = '<LI>'
          Queue Prefix Index.i Decode(Value.i)
          End
        Queue '</UL>'

        If n=''     Then Queue 'No packages matched your search' latin'.'
        Else If n=1 Then Queue '1 package matched your search' latin'.'
                    Else Queue N 'packages matched your search' latin'.'
```

```
  If Mode='RESULT' Then Do
   Queue '<P>You may now'
   if n=1 Then Do
    Queue '<A HREF=/FIND/FREEHEP/'criteria'/FULL>'
    Queue 'examine the entry that matched your' latin'</A>,'
    End
   Else if n>1 Then Do
    Queue '<A HREF=/FIND/FREEHEP/'criteria'/INDEX>'
    Queue 'examine a list of items that matched your' latin'</A>,'
    Queue 'continue to narrow down your search by specifying'
    Queue '<A HREF=/FIND/FREEHEP/'criteria'/NARROW>'
    Queue 'further criteria</A>,'
    End
   Else Do
    Queue '<A HREF=/FIND/FREEHEP/'criteria'/BROWSE>'
    Queue 'browse a list of values</A> that almost matched your'
    Queue 'last criterium,'
    End
   Queue 'or <A HREF=/FIND/FHSPIRES.HTML>start a new search</A>.'
   End
  Else Do /* NARROW */
   Queue '<P>'
   'EXECIO * DISKR FHNARROW HTML ( FINI'
   End
  End
 End
Else Do

 'EXEC QSPIRES FIND' SpiresTerm '( STACK NOSTAR IN' subfile

 Ref = ''
 Title = ''
 i=0
 Do Queued()
  Parse Pull Line
  if Substr(Line,1,10)='No records' Then Do
   Do Queued()
    Pull.
    End
   Mode = 'RESULT'
   Signal Restart
   End

  Parse Var Line . 'Freehep Name:' Name

  if Name /= '' Then Title = Strip(Name)

  Parse Var Line . 'See Also:' Refs
  if Refs = '' Then Do
   i = i+1
   Line.i = Line
   End
  Else Ref = Refs
  End

 If Title/='' Then Do
  Queue '<TITLE>'Title'</TITLE>'
  Queue '<H1>'Title'</H1>'
  End
```

```
Queue '<XMP>'
Do j=1 to i
  Queue Line.j
  End
Queue '
```

```
' if Ref/='' Then Queue 'See also' Do While Ref/='' Parse Var Ref R','Ref R = Strip(R) if Ref='' Then
Punc='.' Else Punc=',' Queue ''R''Punc End End Return Queued() /* Spires search term values may have
spaces in them, but WWW filespecs cannot, so here we encode names. */ Encode: Procedure Parse Arg
String Bad = '% ()?/<>' Good = '%BOCQSGL' Out = '' Do I=1 to Length(String) c = Substr(String,i,1) if
Index(bad,c)/=0 Then Out = Out||'%'Translate(c,good,bad) Else Out = Out||c End Return Out Decode:
Procedure Parse Arg String Bad = '% ()?/<>' Good = '%BOCQSGL' Out = '' Esc = 0 Do I=1 to Length
(String) c = Substr(String,i,1) if Esc Then Do Out = Out||Translate(c,bad,good) Esc = 0 End Else if c =
'%' Then Esc = 1 Else Out = Out||c End Return Out
```

Implementation of the FreeHEP Spires/WWW interface

The interface between FreeHEP/HEPLIB and Spires/WWW was implemented by Tony Johnson (Boston University), with considerable help from Louise Addis, George Crane and the other SLAC WWWizards.

Please report any inaccuracies or problems you encounter when using this system to TonyJ@SlacVX.Slac.Stanford.EDU.

The information in the FreeHEP database was initially compiled by Saul Youssef(SCRI), Andrea Palounek(LANL) and Tony Johnson, based on an earlier compilation by Saul and Andrea.

Implementation of the FreeHEP Spires/WWW interface

The interface between FreeHEP/HEPLIB and Spires/WWW was implemented by Tony Johnson (Boston University), with considerable help from Louise Addis, George Crane and the other SLAC WWWizards.

Please report any inaccuracies or problems you encounter when using this system to TonyJ@SlacVX.Slac.Stanford.EDU.

The information in the FreeHEP database was initially compiled by Saul Youssef(SCRI), Andrea Palounek(LANL) and Tony Johnson, based on an earlier compilation by Saul and Andrea.

You can search this index. Type the keyword(s) you want to search for:

FreeHEP - Library of High Energy Physics Software

In spite of the large number of talented people writing and using software in High Energy Physics, much of this work is hampered simply by lack of information or by lack of easy access to software packages. This feeling was confirmed by our experience in compiling a simple list of existing packages in use in the field. Only a fraction of this software is generally known about. Clearly we need a better mechanism than word of mouth to find out about useful software. The same problem exists with respect to commercial software where there is a need to find out what exists and to share experiences. To help solve this problem, we have set up an organization which will perform the following services for the HEP community:

- A global software compilation -- an extension of our original compilation.
- ftp access to software packages, documentation, instructions for getting software from other locations, reviews and benchmarks.
- A news groups for subject areas and software packages.

FreeHEP is based on the principle that ALL software which might be useful to the HEP community should be included. This includes software from other fields and commercial software packages. Commercial software will appear in the form of instructions for getting software from a company and a news group to share experiences.

Using WWW you can find out more about FreeHEP or access information about freehep software in a variety of ways. If you know the name of the package you are interested in then type the name of the package as a keyword now. Otherwise you can search by subject area, by browsing an alphabetical list of all packages or by using the full power of the Spires database system to access the information that you are interested in.

The WWW interface to FreeHEP is in an initial testing period at the current time. Please report any problems or inaccuracies that you encounter when using WWW to access FreeHEP

You can search this index. Type the keyword(s) you want to search for: □

FreeHEP - Library of High Energy Physics Software

In spite of the large number of talented people writing and using software in High Energy Physics, much of this work is hampered simply by lack of information or by lack of easy access to software packages. This feeling was confirmed by our experience in compiling a simple list of existing packages in use in the field. Only a fraction of this software is generally known about. Clearly we need a better mechanism than word of mouth to find out about useful software. The same problem exists with respect to commercial software where there is a need to find out what exists and to share experiences. To help solve this problem, we have set up an organization which will perform the following services for the HEP community:

- A global software compilation -- an extension of our original compilation.
- ftp access to software packages, documentation, instructions for getting software from other locations, reviews and benchmarks.
- A news groups for subject areas and software packages.

FreeHEP is based on the principle that ALL software which might be useful to the HEP community should be included. This includes software from other fields and commercial software packages. Commercial software will appear in the form of instructions for getting software from a company and a news group to share experiences.

Using WWW you can find out more about FreeHEP or access information about freehep software in a variety of ways. If you know the name of the package you are interested in then type the name of the package as a keyword now. Otherwise you can search by subject area, by browsing an alphabetical list of all packages or by using the full power of the Spires database system to access the information that you are interested in.

The WWW interface to FreeHEP is in an initial testing period at the current time. Please report any problems or inaccuracies that you encounter when using WWW to access FreeHEP

This URL has changed!

The application or page you are calling has moved to a new location.

Go to the main SPIRES page
http://www.slac.stanford.edu/spires/
or:

Please ask the owner of the referring page to change. In general, the filenames have stayed the same, but the machine name has changed from:

slacvm.slac.stanford.edu
to
www.slac.stanford.edu

and, in the case of SPIRES, the search syntax may have changed.

If you were doing a search in the SLAC SPIRES-HEP database, please see:

http://www.slac.stanford.edu/spires/hep/

If you were searching a SPIRES database other than HEP, see:

http://www.slac.stanford.edu/spires

If you have a hard-coded search string, you may need to change the syntax of your search slightly, as shown in the following examples:

Old style (no longer works):

http://www.slac.stanford.edu/find/hep?find+a+beacom,j.f.%2C+%28using+wwwcite

New style:

http://www.slac.stanford.edu/spires/find/hep/wwwcite?a=beacom,j.f..

If you are uncertain about how to modify your particular search syntax, try the search from the appropriate search page and copy the url which is formed from it.

Send comments or questions to: library@slac.stanford.edu

HOC
27 Jun 2001

Addis
2 Jul 1998

```
<ISINDEX>

You can use Spires to continue to narrow down your search by adding
extra search criteria of the following form:

<UL>
<LI>AUTHOR Youssef, Saul
<LI>AUTHOR Rene Brun
<LI>AUTHOR T. Burnett
<LI>SECTION Analysis
<LI>DATE 1992
<LI>DATE March 1992
<LI>TITLE motif
<LI>ABSTRACT radiative
</UL>
<P>
Note that when searching for titles or abstracts any package that
contains the specified word in the title or abstract will match.
You can also form compound searches using the word "AND", for example:

<UL>
<LI>AUTHOR Youssef, Saul AND DATE 1992
<LI>AUTHOR Johnson AND TITLE Motif
</UL>
<P>
Commands are not case sensitive. You can also get a complete list of
<A HREF=/FIND/FREEHEP/SHOWIND>
spires keywords</A>(AUTHOR, SECTION, DATE etc.) available for use
with FreeHEP.
```

You can search this index. Type the keyword(s) you want to search for:

You can use Spires to continue to narrow down your search by adding extra search criteria of the following form:

AUTHOR Youssef, Saul

- AUTHOR Rene Brun
- AUTHOR T. Burnett
- SECTION Analysis
- DATE 1992
- DATE March 1992
- TITLE motif
- ABSTRACT radiative

Note that when searching for titles or abstracts any package that contains the specified word in the title or abstract will match. You can also form compound searches using the word "AND", for example:

AUTHOR Youssef, Saul AND DATE 1992

- AUTHOR Johnson AND TITLE Motif

Commands are not case sensitive. You can also get a complete list of spires keywords(AUTHOR, SECTION, DATE etc.) available for use with FreeHEP.

Original compilation of HEP software

A.Palounek and S.Youssef, "Monte Carlo Programs and other Utilities for High Energy Physics," LBL-29115 (1990). This paper is available via DECNET in postscript format as: SSCVX1::USER1:[APTP] DOCREF.PS

Original compilation of HEP software

A.Palounek and S.Youssef, "Monte Carlo Programs and other Utilities for High Energy Physics," LBL-29115 (1990). This paper is available via DECNET in postscript format as: SSCVX1::USER1:[APTP] DOCREF.PS

FreeHEP anonymous FTP machine

The FreeHEP anonymous ftp machine is called "FreeHEP.Scri.Fsu.Edu". Feel free to FTP to this machine and look at what is there. (You can sign in as user "anonymous", just give your userid on your home machine when you are prompted for a password).

The directories on this machine are organized by subject area. These areas contain data base records (.dbase files) and may also contain source code in some cases. There is a /reviews area for each subject area which is meant for general reviews, benchmarks etc. The file INDEX contains a daily updated index of entries in the data base. The file FILES contains a daily updated complete directory tree. The /tutorials section contains instructions for various common tasks.

Many of the directories on the FreeHEP machine contain software and documentation that you can access using anonymous FTP.

The information in the .dbase files on the FreeHEP machine can also be accessed using WWW.

FreeHEP anonymous FTP machine

The FreeHEP anonymous ftp machine is called "FreeHEP.Scri.Fsu.Edu". Feel free to FTP to this machine and look at what is there. (You can sign in as user "anonymous", just give your userid on your home machine when you are prompted for a password).

The directories on this machine are organized by subject area. These areas contain data base records (.dbase files) and may also contain source code in some cases. There is a /reviews area for each subject area which is meant for general reviews, benchmarks etc. The file INDEX contains a daily updated index of entries in the data base. The file FILES contains a daily updated complete directory tree. The /tutorials section contains instructions for various common tasks.

Many of the directories on the FreeHEP machine contain software and documentation that you can access using anonymous FTP.

The information in the .dbase files on the FreeHEP machine can also be accessed using WWW.

```
/***************************************************************/
/*                                                             */
/* FHSEARCH EXEC - Process search request from WWW             */
/*                                                             */
/* May 1 1992  TonyJ                                           */
/*                                                             */
/***************************************************************/

   Trace Off
   Arg Node Comm '(' options
   Say time() Node Comm '(' Options

   parse var options File'.'Type

   if File = 'FHMAIN' Then Do
      'EXEC FHGET' Node 'FREEHEP/NAME/'Encode(Strip(comm))'/LONG'
      End
   Else If File = 'FHSPIRES' Then Do
      File = 'FREEHEP'
      Do While Comm/=''
        Parse Var Comm Index Value 'AND' Comm
        Value = Encode(Strip(Value))
        File = file'/'Index'/'Value
        End
      File = File'/RESULT'
      'EXEC FHGET' Node File
      End
  Else Do

     p = LastPos('/',options)
     mode = Substr(options,p+1)
     rest = Substr(options,1,p-1)

     if mode='BROWSE' Then Do

       q = LastPos('/',rest)
       File = Substr(rest,1,q)||Encode(Strip(Comm))||'/RESULT'
       'EXEC FHGET' Node File
       End

     Else If mode='NARROW' Then Do

       File = rest
       Do While Comm/=''
         Parse Var Comm Index Value 'AND' Comm
         Value = Encode(Strip(Value))
         File = file'/'Index'/'Value
         End
       File = File'/RESULT'
       'EXEC FHGET' Node File
       End

    Else Do
      Queue '<PLAINTEXT>'
      Queue 'huh?'
      End

    End

 Exit Queued()
```

```
Encode: Procedure
Parse Arg String

Bad  = '% ()?/'
Good = '%BOCQS'

Out = ''
Do I=1 to Length(String)
  c = Substr(String,i,1)
  if Index(bad,c)/=0 Then Out = Out||'%'Translate(c,good,bad)
                     Else Out = Out||c
  End

Return Out
```

```
/***************************************************************/
/*                                                             */
/* FHSEARCH EXEC - Process search request from WWW             */
/*                                                             */
/* May 1 1992  TonyJ                                           */
/*                                                             */
/***************************************************************/

   Trace Off
   Arg Node Comm '(' options
   Say time() Node Comm '(' Options

   parse var options File'.'Type

   if File = 'FHMAIN' Then Do
      'EXEC FHGET' Node 'FREEHEP/NAME/'Encode(Strip(comm))'/LONG'
      End
   Else If File = 'FHSPIRES' Then Do
      File = 'FREEHEP'
      Do While Comm/=''
        Parse Var Comm Index Value 'AND' Comm
        Value = Encode(Strip(Value))
        File = file'/'Index'/'Value
        End
      File = File'/RESULT'
      'EXEC FHGET' Node File
      End
   Else Do

      p = LastPos('/',options)
      mode = Substr(options,p+1)
      rest = Substr(options,1,p-1)

      if mode='BROWSE' Then Do

        q = LastPos('/',rest)
        File = Substr(rest,1,q)||Encode(Strip(Comm))||'/RESULT'
        'EXEC FHGET' Node File
        End

      Else If mode='NARROW' Then Do

        File = rest
        Do While Comm/=''
          Parse Var Comm Index Value 'AND' Comm
          Value = Encode(Strip(Value))
          File = file'/'Index'/'Value
          End
        File = File'/RESULT'
        'EXEC FHGET' Node File
        End

      Else Do
        Queue '<PLAINTEXT>'
        Queue 'huh?'
        End

      End

   Exit Queued()
```

```
Encode: Procedure
Parse Arg String

Bad  = '% ()?/<>'
Good = '%BOCQSGL'

Out = ''
Do I=1 to Length(String)
  c = Substr(String,i,1)
  if Index(bad,c)/=0 Then Out = Out||'%'Translate(c,good,bad)
                     Else Out = Out||c
  End

Return Out
```

Using Spires to Search FreeHEP

You can search this index. Type the keyword(s) you want to search for:

You can use Spires to search for specific packages in FreeHEP. You can issue search commands by typing commands of the following form as "keywords".

- AUTHOR Youssef, Saul
- AUTHOR Rene Brun
- AUTHOR T. Burnett
- SECTION Analysis
- DATE 1992
- DATE March 1992
- TITLE motif
- ABSTRACT radiative

Note that when searching for titles or abstracts any package that contains the specified word in the title or abstract will match. You can also form compound searches using the word "AND", for example:

- AUTHOR Youssef, Saul AND DATE 1992
- AUTHOR Johnson AND TITLE Motif

Commands are not case sensitive. You can also get a complete list of spires keywords(AUTHOR, SECTION, DATE etc.) available for use with FreeHEP, browse an alphabetical list of all packages or search for packages by subject area.

Using Spires to Search FreeHEP

You can search this index. Type the keyword(s) you want to search for:

You can use Spires to search for specific packages in FreeHEP. You can issue search commands by typing commands of the following form as "keywords".

AUTHOR Youssef, Saul
- AUTHOR Rene Brun
- AUTHOR T. Burnett
- SECTION Analysis
- DATE 1992
- DATE March 1992
- TITLE motif
- ABSTRACT radiative

Note that when searching for titles or abstracts any package that contains the specified word in the title or abstract will match. You can also form compound searches using the word "AND", for example:

AUTHOR Youssef, Saul AND DATE 1992
- AUTHOR Johnson AND TITLE Motif

Commands are not case sensitive. You can also get a complete list of spires keywords(AUTHOR, SECTION, DATE etc.) available for use with FreeHEP, browse an alphabetical list of all packages or search for packages by subject area.

FreeHEP sections

The software packages within FreeHEP are divided into sections, with an editor (or editors) for each section. Choose one of the following sections to see a list of all the packages in that section. Note that some packages may appear in more than one section.

- Graphics, visualization, GUIs
- Detector Simulation
- Data Acquisition
- Analysis and Data Reduction & compilation
- Event Generators
- Software Engineering
- General Libraries
- Parallelism and Distributed Computing
- Data bases and file systems
- CAD/CAE Systems
- Artificial Intelligence
- Computer Hardware
- Languages
- Mathematics
- Particle Properties
- Networking/Electronic mail/News
- Cad/Cae Systems

It is also possible to select packages from an alphabetical list of all packages, or to use the full power of Spires to track down the package you are interested.

FreeHEP sections

The software packages within FreeHEP are divided into sections, with an editor (or editors) for each section. Choose one of the following sections to see a list of all the packages in that section. Note that some packages may appear in more than one section.

- Graphics, visualization, GUIs
- Detector Simulation
- Data Acquisition
- Analysis and Data Reduction & compilation
- Event Generators
- Software Engineering
- General Libraries
- Parallelism and Distributed Computing
- Data bases and file systems
- CAD/CAE Systems
- Artificial Intelligence
- Computer Hardware
- Languages
- Mathematics
- Particle Properties
- Networking/Electronic mail/News
- Cad/Cae Systems

It is also possible to select packages from an alphabetical list of all packages, or to use the full power of Spires to track down the package you are interested.

```
/***************************************************************/
/*                                                             */
/* FSEARCH EXEC - Process search request from WWW              */
/*                                                             */
/* George Crane, April 1992                                    */
/*                                                             */
/***************************************************************/

   Trace Off
   Arg comm '(' options
   parse var options subfile .
   upper subfile
   Queue '<PLAINTEXT>'
   Parse var comm . token1 rest
   if Abbrev('FIND',token1,3) = 0 Then temp = 'FIND 'token1 rest
   else temp = token1 rest
   parse var temp token1 token2 rest
   upper token2
   Say temp '('options
   if subfile = 'SPIRES' Then subfile = 'HEP'
   Select
      When token2 = 'REACCESS' Then Do
         Address CMS 'Access 192 B'
         Queue 'Disk re-accessed RC='rc
      End
      When Abbrev('SHOW',token2,3) > 0 Then Do
         'EXEC QSPIRES SHOW 'rest '(STACK NOSTAR IN 'subfile
      End
      When Find('EXPLAIN WHOIS WHATIS WHEREIS QUERY',token2) > 0 Then Do
         'EXEC QSPIRES 'token2 rest '(STACK NOSTAR '
      End
      When Abbrev('BROWSE',token2,3) > 0 Then Do
         'EXEC QSPIRES BROWSE 'rest '(STACK NOSTAR IN 'subfile
         If queued() = 2 Then Do
            parse pull header
            parse pull string
            parse var string first second
            Queue header
            Queue string
            If first = 'Invalid' Then Do
               Queue ' '
               Queue 'Try:  SHOW INDEX for a list of valid terms'
               Queue 'Then: BROWSE term value'
               Queue 'i.e.: BROWSE AUTHOR DRELL'
            End
         End
      End
      When subfile = 'STORES' Then Do
         Parse var temp . temp
         'EXEC STORES' temp '( FIFO NOSTAR'
      End
      Otherwise Do
         'EXEC QSPIRES' temp,
           '( STACK NOSTAR OUTPUT TYPE BRIEF IN 'subfile
         If queued() = 2 Then Do
            parse pull header
            parse pull string
            parse var string first second
            Queue header
            Queue string
         If first = 'Invalid' Then Do
            Queue ' '
            Queue 'Try:  SHOW INDEX for a list of valid terms'
            Queue 'Then: FIND term value'
            Queue 'i.e.: FIND AUTHOR DRELL'
         End
      End
   End
End
Exit Queued()
```

```
/* */ Trace Off Arg comm '(' options parse var options subfile . upper subfile Queue '


'
Parse var comm . token1 rest
if Abbrev('FIND',token1,3) = 0 Then temp = 'FIND 'token1 rest
else temp = token1 rest
parse var temp token1 token2 rest
upper token2
Say temp '('options
if subfile = 'SPIRES' Then subfile = 'HEP'
Select
   When token2 = 'REACCESS' Then Do
       Address CMS 'Access 192 B'
       Queue 'Disk re-accessed RC='rc
       End
   When Abbrev('SHOW',token2,3) > 0 Then Do
       'EXEC QSPIRES SHOW 'rest '(STACK NOSTAR IN 'subfile
   End
   When Find('EXPLAIN WHOIS WHATIS WHEREIS QUERY',token2) > 0 Then Do
       'EXEC QSPIRES 'token2 rest '(STACK NOSTAR '
   End
   When Abbrev('BROWSE',token2,3) > 0 Then Do
       'EXEC QSPIRES BROWSE 'rest '(STACK NOSTAR IN 'subfile
       If queued() = 2 Then Do
          parse pull header
          parse pull string
          parse var string first second
             Queue header
             Queue string
          If first = 'Invalid' Then Do
             Queue ' '
             Queue 'Try:  SHOW INDEX for a list of valid terms'
             Queue 'Then: BROWSE term value'
             Queue 'i.e.: BROWSE AUTHOR DRELL'
          End
       End
   End
   When subfile = 'STORES' Then Do
       Parse var temp . temp
       'EXEC STORES' temp '( FIFO NOSTAR'
   End
   Otherwise Do
       'EXEC QSPIRES' temp,
       '( STACK NOSTAR OUTPUT TYPE BRIEF IN 'subfile
       If queued() = 2 Then Do
          parse pull header
          parse pull string
          parse var string first second
             Queue header
             Queue string
          If first = 'Invalid' Then Do
             Queue ' '
             Queue 'Try:  SHOW INDEX for a list of valid terms'
             Queue 'Then: FIND term value'
             Queue 'i.e.: FIND AUTHOR DRELL'
          End
       End
   End
```

```
End
Exit Queued()
```

```
/****************************************************************/
/*                                                              */
/* FSEARCH EXEC - Process search request from WWW               */
/*                                                              */
/* George Crane, April 1992                                     */
/*                                                              */
/****************************************************************/

   Trace Off
   Arg comm '(' options
   parse var options subfile .
   upper subfile
   Queue '<PLAINTEXT>'
   Parse var comm . token1 rest
   if Abbrev('FIND',token1,3) = 0 Then temp = 'FIND 'token1 rest
   else temp = token1 rest
   parse var temp token1 token2 rest
   upper token2
   Say temp '('options
   if subfile = 'SPIRES' Then subfile = 'HEP'
   Select
      When token2 = 'REACCESS' Then Do
         Address CMS 'Access 192 B'
         Queue 'Disk re-accessed RC='rc
      End
      When Abbrev('SHOW',token2,3) > 0 Then Do
         'EXEC QSPIRES SHOW 'rest '(STACK NOSTAR IN 'subfile
      End
      When Find('EXPLAIN WHOIS WHATIS WHEREIS QUERY',token2) > 0 Then Do
         'EXEC QSPIRES 'token2 rest '(STACK NOSTAR '
      End
      When Abbrev('BROWSE',token2,3) > 0 Then Do
         'EXEC QSPIRES BROWSE 'rest '(STACK NOSTAR IN 'subfile
         If queued() = 2 Then Do
            parse pull header
            parse pull string
            parse var string first second
            Queue header
            Queue string
            If first = 'Invalid' Then Do
               Queue ' '
               Queue 'Try:  SHOW INDEX for a list of valid terms'
               Queue 'Then: BROWSE term value'
               Queue 'i.e.: BROWSE AUTHOR DRELL'
            End
         End
      End
      When subfile = 'STORES' Then Do
         Parse var temp . temp
         'EXEC STORES' temp '( FIFO NOSTAR'
      End
      Otherwise Do
         'EXEC QSPIRES' temp,
           '( STACK NOSTAR OUTPUT TYPE BRIEF IN 'subfile
         If queued() = 2 Then Do
            parse pull header
            parse pull string
            parse var string first second
            Queue header
            Queue string
         If first = 'Invalid' Then Do
            Queue ' '
            Queue 'Try:  SHOW INDEX for a list of valid terms'
            Queue 'Then: FIND term value'
            Queue 'i.e.: FIND AUTHOR DRELL'
         End
      End
   End
End
Exit Queued()
```

```
/*****************************************************************/
/*                                                               */
/* FSEARCH EXEC - Process search request from WWW                */
/*                                                               */
/* George Crane, April 1992                                      */
/*                                                               */
/* Modified to provide escape for FreeHEP stuff May 1 1992       */
/*                                              TonyJ            */
/*                                                               */
/*****************************************************************/

   Trace Off
   Arg comm '(' options
   parse var options subfile .

   If SubStr(subfile,1,2)='FH' | SubStr(subfile,1,7)='FREEHEP' Then Do
     'EXEC FHSEARCH' comm '(' options
     Exit Rc
   End

   upper subfile
   Queue '<PLAINTEXT>'
   Parse var comm . token1 rest
   if Abbrev('FIND',token1,3) = 0 Then temp = 'FIND 'token1 rest
   else temp = token1 rest
   parse var temp token1 token2 rest
   upper token2
   Say temp '('options
   if subfile = 'SPIRES' Then subfile = 'HEP'
   Select
      When token2 = 'REACCESS' Then Do
         Address CMS 'Access 192 B'
         Queue 'Disk re-accessed RC='rc
      End
      When Abbrev('SHOW',token2,3) > 0 Then Do
         'EXEC QSPIRES SHOW 'rest '(STACK NOSTAR IN 'subfile
      End
      When Find('EXPLAIN WHOIS WHATIS WHEREIS QUERY',token2) > 0 Then Do
         'EXEC QSPIRES 'token2 rest '(STACK NOSTAR '
      End
      When Abbrev('BROWSE',token2,3) > 0 Then Do
         'EXEC QSPIRES BROWSE 'rest '(STACK NOSTAR IN 'subfile
         If queued() = 2 Then Do
            parse pull header
            parse pull string
            parse var string first second
            Queue header
            Queue string
            If first = 'Invalid' Then Do
               Queue ' '
               Queue 'Try:  SHOW INDEX for a list of valid terms'
               Queue 'Then: BROWSE term value'
               Queue 'i.e.: BROWSE AUTHOR DRELL'
            End
         End
      End
      When subfile = 'STORES' Then Do
         Parse var temp . temp
         'EXEC STORES' temp '( FIFO NOSTAR'
      End
   Otherwise Do
      'EXEC QSPIRES' temp,
        '( STACK NOSTAR OUTPUT TYPE BRIEF IN 'subfile
      If queued() = 2 Then Do
         parse pull header
         parse pull string
         parse var string first second
         Queue header
         Queue string
         If first = 'Invalid' Then Do
            Queue ' '
            Queue 'Try:  SHOW INDEX for a list of valid terms'
            Queue 'Then: FIND term value'
            Queue 'i.e.: FIND AUTHOR DRELL'
         End
      End
   End
End
Exit Queued()
```