```
       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 Do
      Title = Strip(Name)
      CPos = Index(Line,':')
      End

    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 '<dl>'
     j=0
     Do While j<i
       j=j+1
       If Substr(Line.j,CPos,1)=':' Then Do
         Parse Var Line.J Key ':' Stuff
         Key = Strip(Key)
         If Key='Subject Areas' Then Do
           Queue '<dt>'Key'<dd>'
           Do Until Stuff=''
             Parse Var Stuff Nonsense ',' Stuff
             Nonsense = Strip(Nonsense)
             Queue '<a href=/FIND/FREEHEP/SECTION/'Nonsense'/INDEX>'
             If Stuff='' Then Queue Nonsense'</a>'
             Else Queue Nonsense'</a>,'
             End
           End
         Else If Key='Abstract' Then Leave
         Else Queue '<dt>'Key'<dd>'Strip(Stuff)
         End
       Else Do
         templine = Strip(Line.j)
         If templine/='' Then Queue templine
         End
       End
     Queue '</dl>'

     if Ref/='' Then Queue 'See also'
     Do While Ref/=''
       Parse Var Ref R','Ref
       R = Strip(R)
       if Ref='' Then Punc='.'
                 Else Punc=','
       Queue '<A HREF=/FIND/FREEHEP/NAME/'R'/FULL>'R'</A>'Punc
       End

     If j<i Then Queue '<h2>'Key'</h2>'Strip(Stuff)

     Do While j<i
       j=j+1
       If Line.j="" Then Queue '<p>'
       Else Queue Strip(Line.j)
       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.

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.

Improvements to the FreeHEP/WWW interface

The following improvements have been made to the FreeHEP-WWW interface (September 1992).

A link to the hepnet.freehep news group has been inserted at appropriate places.

- The list of subject areas has been corrected.
- The list of editors updated and corrected. E-mail addresses added.
- Searches of recently changed entries added.
- Format of listings reworked. Can now click on subjects areas to get list of all packages in that subject area.
- This list of improvements has been added.
- Access to tutorials.

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

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 group (hepnet.freehep) for subject areas and software packages.
- A set of tutorials.

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.

You can also see a list of packages whose entries have been updated in the last day, last week, last two weeks, or last month.

The WWW/FreeHEP interface has been recently improved. Please report any problems or inaccuracies that you encounter when using WWW to access FreeHEP

```
<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.

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
- Data Modeling and I/O
- 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

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 Tutorials

The following tutorials are available. If you think there should be more, please let us know. (Of course even better would be to obtain/write it and send it to us!)

how_to_use_email
- how_to_save_disk_space
- how_to_access_freehep_using_spires
- how_to_ftp_things
- how_to_use_NETLIB

```
/***************************************************************/
/*                                                             */
/* 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
   incomm = comm
   subfile = ''
   comm = reverse(comm)
   if pos('(',comm) > 0 Then,
   parse var comm subfile '(' comm
   comm = reverse(comm)
   subfile = reverse(subfile)
   Say "FSEARCH "comm" Subfile="subfile
   parse var subfile subfile'/'IndexTerm .

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

   upper subfile
   Queue '<PLAINTEXT>'
   Parse var comm . token1 rest '(' options
   if Abbrev('FIND',token1,3) = 0 Then temp = 'FIND 'token1 rest
   else temp = token1 rest
   parse var temp token1 token2 rest
   upper token2
   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 options
      End
      When Find('EXPLAIN WHOIS WHATIS WHEREIS QUERY',token2) > 0 Then Do
         'EXEC QSPIRES 'token2 rest '(STACK NOSTAR 'options
      End
      When subfile = 'WHEREIS' Then Do
         Do Queued(); Parse Pull .; End
         Parse var temp . temp
         'EXEC WHEREIS' indexterm temp '( STACK'
         Push '<PLAINTEXT>'
      End
      When Abbrev('BROWSE',token2,3) > 0 Then Do
         'EXEC QSPIRES BROWSE 'rest '(STACK NOSTAR IN 'subfile options
         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' options
   End
   Otherwise Do
      'EXEC QSPIRES' temp,
        '( STACK NOSTAR OUTPUT TYPE BRIEF IN 'subfile  options
      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 .
   Say "FSEARCH "comm" ( "options

   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()
```

```
/*************************************************************/
/*                                                           */
/* 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
   incomm = comm
   subfile = ''
   comm = reverse(comm)
   if pos('(',comm) > 0 Then,
   parse var comm subfile '(' comm
   comm = reverse(comm)
   subfile = reverse(subfile)
   Say "FSEARCH "comm" Subfile="subfile
   parse var subfile subfile'/'IndexTerm .
   subfile = strip(subfile)

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

   upper subfile
   Queue '<PLAINTEXT>'
   Parse var comm . token1 rest '(' options
   if Abbrev('FIND',token1,3) = 0 Then temp = 'FIND 'token1 rest
   else temp = token1 rest
   parse var temp token1 token2 rest
   upper token2
   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 options
      End
      When Find('EXPLAIN WHOIS WHATIS WHEREIS QUERY',token2) > 0 Then Do
         'EXEC QSPIRES 'token2 rest '(STACK NOSTAR 'options
      End
      When subfile = 'WHEREIS' Then Do
         Do Queued(); Parse Pull .; End
         Parse var temp . temp
         'EXEC WHEREIS' indexterm temp '( STACK'
         Push '<PLAINTEXT>'
      End
      When Abbrev('BROWSE',token2,3) > 0 Then Do
         'EXEC QSPIRES BROWSE 'rest '(STACK NOSTAR IN 'subfile options
         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' options
   End
   Otherwise Do
      'EXEC QSPIRES' temp,
        '( STACK NOSTAR OUTPUT TYPE BRIEF IN 'subfile  options
      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
  incomm = comm
  subfile = ''
  comm = reverse(comm)
  if pos('(',comm) > 0 Then,
  parse var comm subfile '(' comm
  comm = reverse(comm)
  subfile = reverse(subfile)
  Say "FSEARCH "comm" Subfile="subfile
  parse var subfile subfile'/'IndexTerm .
  subfile = strip(subfile)

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

  upper subfile
  Queue '<XMP>'     /*  See we can get out! */
  Parse var comm . token1 rest '(' options
  if Abbrev('FIND',token1,3) = 0 Then temp = 'FIND 'token1 rest
  else temp = token1 rest
  parse var temp token1 token2 rest
  upper token2
  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 options
     End
     When Find('EXPLAIN WHOIS WHATIS WHEREIS QUERY',token2) > 0 Then Do
        'EXEC QSPIRES 'token2 rest '(STACK NOSTAR 'options
     End
     When subfile = 'WHEREIS' Then Do
        Do Queued(); Parse Pull .; End
        Parse var temp . temp
        'EXEC WHEREIS' indexterm temp '( STACK'
        Push '<PLAINTEXT>'
     End
     When Abbrev('BROWSE',token2,3) > 0 Then Do
        'EXEC QSPIRES BROWSE 'rest '(STACK NOSTAR IN 'subfile options
        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' options
   End
   Otherwise Do
      'EXEC QSPIRES' temp,
        '( STACK NOSTAR OUTPUT TYPE BRIEF IN 'subfile  options
      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()
```

```
/*****************************************************************/
/*                                                               */
/* GETWHERE - Driver for WHEREIS                                 */
/*                                                               */
/* George Crane, May   1992                                      */
/*                                                               */
/*****************************************************************/

   Arg string
   Say 'Entry GETWHERE: 'string
   Queue '<IsIndex>'
   Queue '<TITLE>GETWHERE: 'string'</TITLE>'
   Select
      When string = 'HELP' Then Do
         'EXECIO * DISKR WHEREIS HELPCMS * ( FINI'
      End
      When string = 'PCNUM' Then Do
         Queue '<H1>SLAC PROPERTY CONTROL</H1>'
         Queue '<P>'
         Queue 'PCNUM enables lookup of important information about'
         Queue 'capital equipment by the PC 5 digit D.O.E.'
         Queue 'Tag Number.  The database'
         Queue 'is maintained by Property Control.  PCNUM'
         Queue 'displays major fields for each record found matching'
         Queue 'the PC 5 digit criteria.'
         Queue '<P>'
         Queue 'Enter a valid property control number including the'
         Queue 'PC prefix string.  If your item does not have the PC'
         Queue 'prefix then you must begin with the letter K followed'
         Queue 'by a space to tell the system you are entering a '
         Queue 'property control number key rather than a WWW index'
         Queue 'number.'
         Queue '<XMP>'
         Queue '        PC33491'
         Queue '        PC12345'
         Queue '        K 38017'
         Queue '
```

' End When string = 'NODE' Then Do Queue '

NODE and DEVICE INFORMATION

' Queue '

' Queue 'Enter a node or device name as...' Queue '

'

```
         Queue '    SLACVX'
         Queue '    EB-PUB1-CL'
         Queue '
```

' End When string = 'ETHER' Then Do Queue '

'

```
         Queue 'Enter a hardware or software Ethernet address as...'
```

```
          Queue '   '
          Queue '   AA-00-04-00-7E-B1'
          Queue '   00-00-0C-01-BC-08'
       End
       When string = 'TAP' Then Do
    Queue '<PLAINTEXT>'
          Queue 'Enter a network tap number as...'
          Queue '   '
          Queue '   T03025'
          Queue '   T01041'
       End
       When string = 'DEVICE' Then Do
    Queue '<PLAINTEXT>'
          Queue 'Enter a SLACVM physical device address as...'
          Queue '   '
          Queue '   1181'
          Queue '   1115'
       End
       When string = 'MICOM' Then Do
    Queue '<PLAINTEXT>'
          Queue 'Enter a Micom channel as...'
          Queue '   '
          Queue '   MICA1/115'
          Queue '   MICB1/110'
       End
       When string = 'PORT' Then Do
    Queue '<PLAINTEXT>'
          Queue 'Enter a Port address as...'
          Queue '   '
          Queue '   S1D-15'
          Queue '   LDCD-27'
       End
       When string = 'CABLE' Then Do
    Queue '<PLAINTEXT>'
          Queue 'Enter a Cable address as...'
          Queue '   '
          Queue '   D1029'
          Queue '   D995'
       End
       When string = 'PRT' Then Do
    Queue '<PLAINTEXT>'
          Queue 'Enter a RSCS Printer name as...'
          Queue '   '
          Queue '   IMCGB3A'
          Queue '   IMCLA1'
       End
       When string = 'VOLSER' Then Do
    Queue '<PLAINTEXT>'
          Queue 'Enter a valid tap VOLSER as...'
          Queue '   '
          Queue '   SA1234'
          Queue '   SB2221'
       End
       When string = 'INSTITUTION' Then Do
    Queue '<PLAINTEXT>'
          Queue 'Enter a valid HEP Institution as...'
          Queue '   '
          Queue '   SLAC'
          Queue '   CERN'
       End
```

```
      When string = 'SUMMARY' Then Do
         'EXEC GIME NETLIST (STACK'
         parse pull mode .
   Queue '<PLAINTEXT>'
         'EXECIO * DISKR TOPENET STATS 'mode' ( FINI'
         'EXEC DROP 'mode
      End
      Otherwise Do
   Queue '<PLAINTEXT>'
         Queue 'Invalid Parameter: 'string
      End
   End
   Exit 0
</XMP></BODY></HTML>
```

SLACVM SPIRES HEP Preprint Database

Search
    Perform search using standard SPIRES terms.

Help
    Get help for SPIRES

HEP (High-Energy Physics) is a joint project of the SLAC and DESY Libraries. As of Jan 1992, it included more than 239,000 bibliographic records dating from 1974 to the present.

Clone copies of HEP run under SPIRES at DESY, KEK, and Yukawa Inst. at Kyoto and are kept up to date by nightly updates via BITNET.

HEP is updated daily with new preprints received in the SLAC library and biweekly with new journal articles and conference proceedings papers indexed at the DESY Library. Input is also received via BITNET from CERN, Fermilab, KEK, Yukawa Inst. at Kyoto U., and SSCL (as of March 1990).

HEP includes all SLAC Library preprint and report holdings from 1974 (all SLAC items from 1962+) as well as all journal articles, conference papers, theses, etc. from the DESY High-Energy Physics Index, a comprehensive bibliography produced at our sister laboratory in Hamburg, Germany.

Here are some examples of typical searches:

```
-> FIND AUTHOR DORFAN, J.

-> FIND TITLE BLACK HOLE# AND DATE AFTER 1988
```

SLACVM SPIRES HEP Preprint Database

Search
    Perform search using standard SPIRES terms.

Help
    Get help for SPIRES

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

```
SLAC SPIRES HEP Preprint database search

Use standard SPIRES search terms such as...

   find author Perl, M
   find title tau and date 1980
```

A BRIEF GUIDE TO THE SPIRES-HEP DATABASE(via WWW)

HEP is a joint project of the SLAC and DESY Libraries, with contributions from the Yukawa Inst. at Kyoto, KEK, CERN, Fermilab, SSCL and Serpukov.

It includes SLAC Library preprint and report holdings since 1974, and all journal articles, etc., from the DESY High-Energy Physics Index (a comprehensive bibliographic tool produced at our sister laboratory in Hamburg, Germany). Also all SLAC documents from 1962. Serves as on-line catalog for SLAC Library and bibliographic search tool for particle physicists in 29 countries. Clone copies of HEP also run under SPIRES at DESY, KEK, and at the Yukawa Inst. The database is available remotely over BITNET via the QSPIRES server to all physicists on the Internet.

250,000 titles as of 1992. Expands at rate of 20,000 per year. Updated daily.

COMMAND ABBREVIATIONS

Most SPIRES command words may be abbreviated to the first 3 characters, i.e. FIND, FIN. Common exceptions are CLEAN (CLN), KEEP (KEEP) and CONTINUE (CONT).

```
      BACKUP BAC
      BROWSE BRO
      EXPLAIN EXP
      FIND FIN
      IN ACTIVE TYPE IN ACT TYP
      OUTPUT CLEAN CLEAR OUT CLN CLE
      OUTPUT CLEAN CONTINUE OUT CLN CONT
      SELECT SEL
      SHOW FORMATS SHO FOR
      SHOW SELECT SHO SEL
      SHOW SUBFILES SHO SUB
      TYPE TYP
      TYPE KEEP TYP KEEP
      TYPE SKIP TYPE SKI
```

SUBFILE names cannot be abbreviated. SEARCH TERMS may be abbreviated as shown in each example or as shown in the SHO SEA TERMS listing.

**7. SAMPLE SEARCH

```
      -> select HEP
      -> find a dorfan, d
           -Result: 121 DOCUMENTS
      -> and date after Dec 1989
           -Result: 19 DOCUMENTS
      -> type pause
      -> find citation PRLTA,33,1406 and not author Richter
           -Result: 990 DOCUMENTS
      -> type pau
      -> browse topic HIGGS
```

```
       -> find tp HIGGS# and date after 1990
            -Result: 347 DOCUMENTS
       -> sequence author
       -> spisend ###
                             (where ### is someone"s e-mail
                              address, i.e., Addis@slacvm.bitnet
```

SELECT <database name>

To choose a SPIRES database for searching:

```
       -> SEL(ECT) <database name>
```

To change to another, simply select.

```
       -> SEL HEP
       -> SEL BOOKS
```

Database names (called subfiles in SPIRES) cannot be abbreviated. To find out what databases are available:

```
       -> show subfiles [sho sub]
       -> show subfile like HEP
                              [sho sub like HEP]
```

To find the database currently selected:

```
       -> SHOW SELECT [sho sel]
```

If you are searching from a 'SPIRES ONLY' account, a limited number of databases will be accessible.

SHOW INDEXES (simple indexes and qualifiers)

After a database has been selected, SHOW INDEXES to list the valid search terms:

```
       -> SEL HEP
       -> SHO IND

       Goal Records: DOCUMENT, DOCUMENTS, KEY
       Qualifier: D, DATE
       Qualifier: PPF-SUBJECT, PS, SCL
       Simple Index: A, AU, AUTHOR
       Simple Index: T, TI, TITLE ... etc.
```

ÔGoal-recordsÕ are results of a search; e.g.,

```
           -Result: 10 DOCUMENTS
```

SIMPLE INDEX terms may be used in primary searching. QUALIFIER terms (DATE, PPF- SUBJECT) may only be used to narrow a previous result. Any listed term is valid for search.

```
       -> find <simple index term> xxxx
            -Result: # DOCUMENTS
       -> and <qualifier> xxxx
```

Example:

```
      -> find title Z0 and date 1990
                                                   BUT NOT
      -> find date 1990 <Ñ(illegal search in HEP because date is
 a 'qualifier')
```

IMPORTANT SEARCH TERMS

LONG FORM	SHORT FORM	EXAMPLE
AUTHOR	A	Fin a Prescott, C.
FIRST-AUTHOR	FA	fin fa Prescott, c.
TITLE	T	fin t aleph detector
AFFILIATION	AF	fin af Rochester U.
DATE	D (qualifier, see above)	fin af CERN & date 1992
CITATION	C	fin c ZEPYA,C44,15
COLLABORATION	CN	fin cn Mark-II Collaboration
REPORT NUMBER	R	fin r SLAC-PUB-5000
PPF SUBJECT	PS (qualifier, see above)	fin t higgs & scl r
EXPERIMENT	EXP	fin exp fnal-e-0741
TOPIC	TP	fin tp experimental results

More examples:

```
      fin author amos, n.
           -Result: 22 DOCUMENTS
      ->
      fin fa amos, n
           -Result: 13 DOCUMENTS
      ->
      fin a amos, n and avila, c
           -Result: 6 DOCUMENTS
      ->
      fin a amos, n and avila, c and t rho
           -Result: 1 DOCUMENT
      ->
```

Note that it is not necessary to repeat same search term.

BEGIN A SEARCH (FIND)

To start a search (and discard the results of any previous search), type:

```
      -> FIND <search term> <something>

      -> FIND AF SLAC                     <Search begins>
           -Result: 8691 DOCUMENTS
      -> AND AF CERN                      <Same search continues>
           -Result: 188 DOCUMENTS
      -> FIND AF CERN                     <New search begins>
           -Result: 15407 DOCUMENTS
      -> TYP PAU
```

Until you reset the search with a FIND command, you are only adding to or subtracting from the previous result.

In this example, to stop before you have seen all 15,407 records, type HE (for halt execution), and hit the RETURN or ENTER key.

BACKUP

You can back up one step at any time during a search (but not more than one):

```
      -> FIND T Z0#
          -Result: 672 DOCUMENTS
      -> AND DATE DEC 1989
          -Result: 10 DOCUMENTS
      -> BACKUP
          -Result: 461 DOCUMENTS
```

BOOLEAN SEARCH LOGIC (and, or, and not)

```
               AND (&)
               OR ( )
               AND NOT (&~)

      -> fin a smith AND jones, m.  <Finds papers with Smith &
                                     Jones as coauthors.>

      -> fin a smith AND NOT jones, m. <Finds papers with author
                                        Smith but not Jones.

      -> fin a smith OR jones, m.  <Finds papers with either Smith
                                    or Jones.  Includes papers with both.

      -> fin a smith & t photoproduction   <Finds papers with
                                            author Smith and with word
                                            photoproduction in title.
```

GROUPING AND NESTING (more about search logic)

Logic operators are processed in order of occurrence from left to right.

```
      -> fin A Jones AND Smith OR Brown

      is grouped as A (Jones and Smith) or (Brown)

      -> fin A Brown OR Smith AND Jones

      is grouped as A (Brown or Smith) AND (Jones)
```

Use parentheses to change processing order. Parens may be nested. Logical operators are processed left to right inside each set of parens, starting with inmost.

```
      -> fin A Jones AND (Smith OR Brown)
      -> fin A Jones & (Smith or (Brown & Astbury))
      -> typ pau
```

RIGHT TRUNCATION (#) , WILD CARD, and/or PREFIX SEARCH

---

is used as right truncation or 'wild card' sign in all simple indexes (author, affiliation, title, report no., topic, citation, experiment, etc.)

```
       -> FIND AUTHOR FEYN#
                  or
       -> FIND T HADRON# MODEL#
```

Above finds all title words with stem HADRON, i.e., hadron, hadrons, hadronic and stem MODEL, i.e., models, modelling, modelled. To check terms you will get with truncated search, browse (without #):

```
       -> bro t hadron
```

may be used multiply in title search as above, but singly in A, AF, TP, RN.

```
       -> FIND A SCHWAR#
       -> FIND TP TARGET, JET#
```

Some indexes in other databases do not support truncated searching (#) but the ÔprefixÕ search can always be used instead.

```
       -> FIND AUTHOR PRE FEYN
       -> FIND R PRE HEPEC
```

are equivalent to:

```
       -> FIND A FEYN#
       -> FIND R HEPEC#
```

The report number and collaboration indexes support imbedded wild card searching. For example:

```
       -> FIN R JINR#89-144
       -> FIN R JINR#89-122
```

will find JINR-E2-89-144 and JINR-P2-89-122 respectively. The # or Ôwild cardÕ stands for any character.

```
       -> FIN CN AACHEN#BEIJING
```

finds 8 documents out of the 227 starting with collaboration name AACHEN.. It finds only those collaboration names beginning with AACHEN and containing BEIJING someplace within the remainder of the collaboration name.

```
--------------------------------------------------------------
BROWSE
--------------------------------------------------------------
```

To examine random terms in an index:

```
       -> BRO(WSE) <search term>
       -> BRO AUTHOR
       -> BRO TITLE
```

To look at ALL terms in a particular alphabetic part of the index:

```
       -> BRO(WSE) <search term> <word>
```

```
-> BRO AUTHOR SMI
-> BRO T REGGE
-> BRO CODEN PHYSICAL
```

To look at index from A to Z (or Z to A):

```
-> BRO FIRST <search term>
-> BRO FIRST AUTHOR
-> BRO LAST <search term>
```

A few items will be listed and SPIRES will ask whether you wish to continue:

```
MORE?
                      continue = YES, Y, OK
                      no more = NO, N
                      backwards = BACKWARD, B, DOWN,
                      forwards = FORWARD, F, UP, +
```

WARNING: you must answer before you can continue searching. One of common error is to start a search in response to 'MORE?'

AUTHOR (A) SEARCH

Author search may be phrased:

```
-> FIND AUTHOR S.  DRELL
-> FIND A DRELL, S.
-> FIND A S.D.  DRELL
-> FIND A DRELL
```

Each example will find Sydney D.Drell.

BEWARE: First and second examples would also find P.S. Drell. Third example (S.D. Drell) will not find S. Drell or Sid Drell. Fourth example (D. Drell) would find John D. Drell.

```
-> find author Smith, John
```

will find all the John Smiths including John A. Smith and John X. Smith.

VAN DER LANS may be found as:

```
-> FIND author LANS
-> find a der lans
-> find a van der lans
```

Because of the complexity and variety of names, CHECK RESULTS OF AUTHOR SEARCHES CAREFULLY. If you are uncertain of a spelling, browse:

```
-> bro a dumbrais
```

Beware: When truncating author names, only truncate the surname. Do not truncate first names or initials (which will intefere with the natural truncation and produce unexpected and results).

Where two authors share the same surname and initials, you may need to try to eliminate the 'wrong' one by using the 'AND NOT' command. For instance, AND NOT AF something or AND NOT AUTHOR somebody.. Sometimes that's not possible and you'll need to resort to TYPE KEEP and check the papers one by one.

AFFILIATION (AF) SEARCH

Affiliation names must be written exactly or truncated. To find exact form, be sure to browse:

```
      -> bro af lbl
      -> bro af caltech
```

To find all papers having authors from a particular institution:

```
      -> find af <institution name>
      -> find af LBL, Berkeley
      -> fin af LBL#
      -> fin af Cal Tech
```

The Ôauthority fileÕ for exact form is a SPIRES file called ÔINSTITUTIONSÕ. You can also find the correct form for affiliation by using the QSPIRES WHEREIS command (for which you donÕt need to interrupt your session).

```
                  QSPIRES WHEREIS LBL
                  QSPIRES WHEREIS VANCOUVER
                  QSPIRES WHEREIS SSC
                  QSPIRES WHEREIS HAMBURG
                  QSPIRES WHEREIS SERP#
```

DATE (D) QUALIFIER SEARCH

DATE is a Ôqualifier Ô and must be combined with title, author, tp, etc. search. All forms of date are acceptable except European.

```
                        16 Jul 1989
                        7/16/89
                        7/89
                        Jul 89
                        July 16, 1989
                        NOT 16-7-89 (European style)

      -> find a Drell and d after 1979
      -> find a bjorken & d before 2/81
      -> find a richter & d nov 74
                        NOT
      -> find d 1982 (illegal search)
```

If a preprint is undated, date search defaults to date received at SLAC.

TITLE (T) SEARCH

Common words (the,then, etc.) are ignored in title search. The following example

```
      -> find t regge# pole#
```

finds all documents with word stems ÒreggeÓ and ÒpoleÓ anywhere in title in any order. (See TRUNCATION). Reserved characters must be quoted.

```
      -> find t Òsu(3)Ó

      -> find t ÒRHO(1600)Ó
```

TOPIC (TP) SEARCH

Topic searches are often large so prepare by getting extra memory.

```
      -> SPIRES space 250p
```

Topic must be stated precisely. BROWSE topic indices or consult printed lists (available from SLAC Library) to find correct terms. IMPORTANT: Remember to combine your topic search with a title search to insure coverage of the most recent preprints. If you are doing a subject search for a review paper or to verify that work has not been done previously, consult a reference librarian. It may be necessary to consult more than one database to get complete coverage of your topic. Call SLAC extension 2411 and ask for ÔReference.Õ

```
      -> select hep
      -> browse tp higgs
      -> find tp higgs particle#
      -> and tp experimental result#
      -> find tp Òrho(765),productionÓ
      -> find tp rho# production
```

Truncation sign (#) may be used only once, at end or internally. Quotes must be used when parens or other reserved words or symbols appear in topic phrase.

Topic search SHOULD ALWAYS BE COMBINED WITH A TITLE SEARCH to include recent items which do not yet have topics assigned. Only 80% of items in HEP have topic words assigned.

```
      -> fin tp rho# or ti rho#
```

To see the topics (keywords) assigned to each paper:

```
      -> set format allkeys
      -> typ pau
```

To return to the regular format:

```
      -> set for default
```

To use a very brief format:

```
      -> set format brief
```

TITLE & TOPIC TERMINOLOGY (how do we show Greek letters, subscripts, etc.)

Current character set lacks Greek letters, subscripts, etc. Conventions are:

```
      pi- pi+ pi0 anti-k0
      rho-prime tau+- Z0
```

Browse title or topic to verify terms.

```
      -> bro t RHO
```

Most superscripts are shown in Fortran notation, i.e. e=mc**2. Subscripts are usually parenthesized or dashed. When in doubt, browse.

Most hyphens and are stripped from titles, with a few exceptions like HIGH-ENERGY. To check, browse the title index. If the word doesnÕt appear there with a hyphen, then the hyphen isnÕt used in searching.

RESERVED WORDS & CHARACTERS

Several words & symbols are reserved for special use in SPIRES. If any occur inside of topic phrases or titles, the whole search phrase must be enclosed in quotes.

```
                                or and & not ~ >< = ()

      -> find Òtp counters and detectorsÓ
      -> find tp ÒANTI-P N Ñ> PI+#Ó
      -> find title ÒSU(1,3)Ó
```

REPORT-NUM (R) SEARCH

Reports and preprints received at SLAC are indexed and shelved by report numbers.

```
      -> FIND R SLAC-PUB-5223
      -> FIN R SLAC PUB 5227
      -> find r SLAC PUB#
      -> FIND R CERN UA1 TN 90 01
```

Notice that special characters Ô(/.-Õ may be typed as blanks. SLAC TNÕs, PUBÕS, etc. have leading zeros added to numbers for sorting purposes. Other report numbers are unaltered, although a few are sometimes ÔfixedÕ to be consistent with others in a series. In a few instances such as HEBREW, FREIBURG, EP-CPT, JINR MIT-CTP, LURE, GOTEBORG and SACLAY prefixes are assigned to numbered series which have no prefix of their own. It is very important to browse report numbers to see what form is being used. Another good technique is to first find a group of reports by ÔaffiliationÕ and check the report numbers for form.

```
      -> FIND AF UC, Santa Cruz
      -> FIND R SCIPP#
```

Unnumbered preprints are assigned ÒPRINTÓ numbers for filing purposes. The first unnumbered preprint received at SLAC in 1990 was PRINT-90-0001 (LOUISIANA STATE). The information in parens after the print number shows the source of the preprint and can be ignored in the search.

```
      -> find r print-90-0001
```

All Louisiana State Prints can be searched:

```
      -> FIND R PRINT#LOUISIANA STATE
```

Some institutions are woefully inconsistent in their assignment of report numbers, even duplicating numbers. It's really important to browse the report number index and use other methods of searching if you don't find the report number you're looking **for. The SLAC Library reference librarians will be glad to help. Call x2411 and ask fo 'reference'.

CITATION (C) SEARCH

Citation search finds most preprints which cited a particular journal article. You must know journal (coden), volume & page.

```
      -> find c PHRVA,D4,3388
```

finds papers which cite the Phys.Rev., Vol. D4, page 3388. PHRVA is the standard five letter coden for Physical Review.

Section letters are attached to volume. For example, Phys.Rev.D,vol.1 is written PHRVA,D1. Nuovo Cimento A, vol.66 is written NUCIA,66A. For searching it is not necessary to know whether the letter precedes or follows the volume number.

```
      -> FIN C PHRVA,D30,1002
      -> FIN C PHRVA,30D,1002
```

Both the above searches will yield the same result.

WARNING: Not all papers indexed in HEP include citations. At present, March 1992, only papers received at the SLAC Library as preprints and announced in Preprints in Particles and Fields have their citation list entered. So the citation search is useful but not definitive and should be used carefully. In 1989, 7469 preprints were included in the set, less than half the almost 17000 items added to HEP in that year. We wish we could do more but our resources are woefully limited.

SPICITE (automated citation searching)

If you wish to find citing papers for a whole group of papers, for instance, to find who has cited each of your own published papers, you can use the ÔSPICITEÕ exec developed by Dr. K. Aoki at Kyoto University. Just get your result, narrow it down to published papers, and issue the command: ÔSPICITEÕ. (Not available via QSPIRES)

COMMON CODEN (browse coden <journal title>)

J.Math.Phys.	JMAPA
Int.J.Mod.Phys.	IMPAE
Mod.Phys.Lett.	MPLAE
Nucl.Instr.Meth.	NUIMA
Nucl.Phys.	NUPHA
Phys.Lett.	PHLTA
Phys.Rev.	PHRVA
Phys.Rev.Lett.	PRLTA
Prog.Theor.Phys	PTPKA
Rev.Mod.Phys.	RMPHA
Z. Phys.	ZEPYA

To find other CODEN:

```
      -> BROWSE CODEN <journal name>
      -> BRO CODEN ANNALS OF PHYSICS
```

EXPERIMENT NUMBER SEARCH

A new index was created in HEP in response to requests from various users. Currently, March 1992, most papers from SLAC, Fermilab and Cornell experiments are tagged with experiment numbers indexed in the EXP index. SLAC, Fermilab, and Cornell have done the tagging of their own papers. Many papers from CERN and BNL are also tagged. This is an ongoing project and we hope that other labs will join in with experiment coding in the future. At present, this index should be used with caution since it may yield only partial results.

```
      -> BROWSE EXP SLAC-SLC
      -> BROWSE EXP FNAL
      -> FIN EXP FNAL-E-0741
      -> SET FORMAT ALLKEYS
      -> TYPE
```

Setting the format ÔallkeysÕ lets you see the experiment numbers assigned to papers in the result and all the topic phrases.
For a briefer format which also shows experiment number:

```
      -> SET FORMAT BRIEFX
```

PPF SUBJECT (PS) QUALIFIER SEARCH

All preprints announced on PPF are given one or more of the following codes: I,C,E,T,R.

```
      C Computer hardware or software
                          for experiment or data analysis
      ET Theory with discussion of data
      E New experimental data
      T Pure theory
      ER Experimental review
      TR Theory review
      ETR Experimental & Theory review
      IE Experimental data & detailed
                            discussion of instrumentation
      I Instrumentation for experiments
                            (not accelerator technology)
```

Searches using the PS qualifier must be appended to a simple index search.

```
      -> find t TAU# and PS e
```

SEQUENCE (seq) <elements (How to sort your result)

After a search, results are ordered in roughly reverse chronological order. (There are some variations because not all entries to file were made in chronological order.) To reorder the the result, use the SEQ command. For example, to sequence by author and title:

```
      -> find t Z0#
      -> seq A T
```

Result will be sequenced by first author and sub-sequenced by title. (Name of the element used to SEQ is not always same mnemonic as search term.) SHO ELEMENTS to see element names.

```
      -> SHO ELEM
```

To obtain a true reverse chronological order:

```
      -> seq ds (d)
```

This sorts records in descending order by a special composite ÔvirtualÕ field called Ôdate-sortÕ or DS. It contains document date, if available, or Ôdate-receivedÕ.

SET FORMAT xxxx (write out your result in TeX source

Some SPIRES subfiles have more than one output format (or view) available. For instance:

```
      -> SEL HEP
      -> SHO FORMATS
                            KEYS
                            CARDS
                            DEFAULT (set) **...  The format
currently selected is marked.  To see a complete ÔunformattedÕ
record, get a result:
```

```
      -> CLEAR FORMAT
      -> TYPE
```

To set other formats:

```
      -> SET FOR(MAT) <name>
      -> SET FOR BRIEF
      -> SET FOR ALLKEYS
      -> SET FOR DEFAULT.ALLAUTH
      -> SET FOR TEX
```

In the HEP subfile, the format named DEFAULT is the standard bibliographic format. ALLKEYS shows all TOPICS and experiment numbers assigned to each document.. DEFAULT.ALLAUTH shows all authors no matter how many hundreds. TEX produces a TeXable file for a bibliography.

FOR MORE INFORMATION about SPIRES

SPIRES (Stanford Public Information REtrieval System) is a very powerful, 4th generation database management system which has hundreds of features of which the average user is usually quite unaware. To find out more about SPIRES on-line, use the EXPLAIN and the SHOW EXAMPLE commands.

```
      -> explain explain command
      -> explain show example
      -> show example find
      -> exp filters
      -> exp store result
      -> exp store stack
      -> exp sequence
```

To find out about a particular database (called a subfile):

```
      -> EXPLAIN <subfile name>
```

For more detailed printed information about features of the SPIRES search, consult:

SPIRES SEARCHING AND UPDATING MANUAL

To obtain this and other SPIRES searching, reporting and database design and management technical documents:

```
SPIRES -> perform publish
```

and select the documents you want to print. (If you're not at SLAC, you can obtain SPIRES documentation thru the SPIRES Consortium at Stanford University, E-mail to GA.RLL@STANFORD.BITNET).

Remember, however, that these documents do not specifically explain the library databases. You'll still need the specialized information in the brochures and on-line documentation available from the SLAC Library.

QSPIRES users and many regular SPIRES HEP users may want to send for the recently completed and very comprehensive GUIDE TO QSPIRES by H. Galic. Request it by Email from QSPI@SLACVM.

For help in searching one of the library files, ask a reference librarian at the SLAC Library, x2411 or send E-mail to ADDIS@SLACVM (Louise Addis) or LIRYG@SLACVM (Bob Gex) . For help with QSPIRES, send E-mail to QSPI@SLACVM (Harv Galic).

OTHER SOURCES OF INFORMATION, i.e. INSPEC

Remember that not all subjects are covered in the HEP database and a reference librarian can help you find other databases which may help you solve a specific problem. For example, Physics Abstracts, Electrical and Electronic Engineering Abstracts, and Computer and Control Abstracts are all available as 'INSPEC' thru a commercial database service.