STAR Fortran Code Template -- Draft Proposal

The following is a proposal for a new Fortran code template which enables the automatic generation of man pages using the *mangen* tool. It allows the user to generate a man page for each routine in a file, as well as one describing the file itself. I propose that this template be used for *all* Fortran code developed for STAR.

The rules for this template are described below. If you have any comments or objections please send them to Matt Bloomer .

Example of template for tfs.f

What follows is how the template should look in order for *mangen* to work.

```
C> tfs.f - a program to aid my understanding of mangen.sh C<

C> Copyright 1993  C<

C>
C  modification history
C  --------------------
C  07jul93,abh written
C<

C>
C
C DESCRIPTION:
C TAS implementaion of the FST standalone program.
C
C<

C>------------------------------------------------------------------------
C
C TFS_MAIN - a sample routine
C
C DESCRIPTION:
C TAS implementaion of the FST standalone program. Not all the
C functionality of the original program could be preserved.
C
C INPUT ARGUMENTS:
C  mhitsh - header for MCT mhits_tpc table
C  mhits  - rows of MCT mhits_tpc table
C  mkineh - header for MCT mkine table
C  mkine  - rows of MCT mkine table
C  harh   - header for TFS fspar table
C  par    - rows of TFS fspar table
C  bmparh - header for TFS bmpar table
C  bmpar  - rows of TFS bmpar table
C  ctrlh  - header for TFS ctrl table
C  ctrl   - rows of TFS ctrl table
C  auxhh  - header for TFS auxhit table
C  auxh   - rows for TFS auxhit table
C  tphith - header of TCL tphit table
C  tphit  - rows of TCL tphit table
C
C OUTPUT ARGUMENTS:
C  values in tphit table
C  values in auxh  table
C
C AUTHOR:
C Peter G. Jones  LBL  (510)-486-5436
```

```
C
C CREATION DATE:
C 04-Mar-1993
C Copyright 1993    Lawrence Berkeley Laboratory
C
C<---------------------------------------------------------------------
      INTEGER FUNCTION TFS_MAIN( mhitsh, mhith, mhit,
     >                           mkineh, mkine, parh, par,
     >                           bmparh, bmpar, ctrlh, ctrl,
     >                           auxhh, auxh, tphith, tphit )

      IMPLICIT NONE
```

Description of template and associated rules

mangen uses a special character sequences `C>` and `C<` to denote the beginning and ending of comment sequences which will appear within the generated man pages as categories. Certain comment sequences are expected to appear while the others are optional. The template format has been chosen to conform as much as possible to the man page conventions of UNIX. It is assumed that a man page will be generated separately

- for each routine, and
- for a file with more than one routine.

As a consequence, there are really two templates per file: a header template for the the collection of routines within the file (which may or may not be a standalone program), and a routine template for each routine within the file.

The rules for the template depend on (i) what is necessary for *mangen* to work properly, and (ii) what I think is an appropriate minimum of information which should be present in all our code. I will use the word "mandatory" to mean that the corresponding rule is necessary for *mangen* to work, while the word "required" will mean (ii) above.

The rules for the header template are as follows:

1. Each instance of `C> comments C<` must be separated from each other by a blank line.

2. The first line in the header template is the title line and is mandatory. The title line obeys the syntax:

   ```
   C> {source filename} - {one line comment}C<
   ```

   The source filename in the title *must* agree with the actual filename of the source code. The source filename must be separated from `C>` by only one space.

3. The next mandatory comment sequence is the modification history, which must contain the ascii string `modification history` in lower case, separated from the comment character by one space.

4. The last mandatory comment sequence is the file description, which can be as long as desired. The ascii string `DESCRIPTION` must appear in uppercase, separated from the comment character by one space.

5. Interspersed within these mandatory comment sequences can be any number of optional ones such as the copyright mark, which is required.

6. In general, each comment sequence constitutes a separate category on the man page, and the category is given by the a special ascii string descriptor. This ascii descriptor must be separated from the comment character(s) by a space.

The routine template begins with every instance of `C>----------` , and the rules which apply to it follow:

1. The information in the routine template (with the exception of the function or subroutine declaration) is contained entirely within two special comment lines as shown:

   ```
   C>------------------------------------------------------------------
   C
   C                          {some comments}
   C
   C<------------------------------------------------------------------
   ```

   At least 10 hypens are mandatory. Blank lines within the routine template which do not begin with `C` in the first column are not allowed.

2. The first mandatory comment sequence is the routine title, which has a syntax similar to the header template:

   ```
   C TFS_MAIN - a sample routine
   ```

   It is mandatory that the routine title must agree exactly with the name of the routine in the declaration (even in case). The title sequence must be no more than one line.

3. The remaining comment sequences within the template are delimited by headings in uppercase followed by related text:

   ```
   C HEADING1
   C       This is some descriptive text about heading1
   C that people would like to know.
   C
   C HEADING2
   C       This is some more descriptive text, but this
   C time about heading2.
   C
   C HEADING3
   C       etc.
   ```

   The descriptive text should be cogent but there is no intrinsic limit to its length. Multiple paragraphs spaced by one or more blank comment lines are allowed. Each heading appears as a separate category in the man page along with its descriptive text.

4. The only other mandatory comment sequence is the routine description, which is denoted by the ascii string `DESCRIPTION` in upper case. The comments under this category are meant to be a short description of the routine and its functionality.

5. The other required (when appropriate) headings are:

   ```
   INPUT ARGUMENTS
   ```

```
OUTPUT ARGUMENTS
AUTHOR
CREATION DATE
```

which have mostly self–evident meanings. In addition to these specified here, the user may add any other headings deemed appropriate, as long as the heading is completely in uppercase and occupies one entire line. *mangen* is not concerned with what the user actually puts as headings and comments, except for the title and description headings. However, there is one heading, `INTERNAL`, which is understood by *mangen* , and all text under this heading will not be included in the man page.

6. The last line of the routine template is `C<----------` . Immediately following this must be function or routine declaration, which appears in the `SYNOPIS` category. The function or routine declaration must be separated by a blank line from all other source lines and declaration below it.

The Resulting Man Pages

The man pages for the above file `tfs.f` can be generated by typing

```
> mangen lib tfs.f
```

where the lib argument tells *mangen* that the file contains several routines of Fortran source. The resulting files `tfs.1` and `tfs.2` are:

tfs.1:

```
tfs(1)                   USER COMMANDS                    tfs(1)



NAME
     tfs.f - a program to aid my understanding of mangen.sh

SYNOPSIS
     TFS_MAIN - a sample routine

           INTEGER FUNCTION TFS_MAIN( mhitsh, mhith, mhit,  ...

DESCRIPTION
     TAS implementaion of the FST standalone program.


{space}



STAR Reference Manual     Last change:                         1
```

tfs.2:

```
TFS_MAIN(2)                SYSTEM CALLS                TFS_MAIN(2)



NAME
     TFS_MAIN - a sample routine

SYNOPSIS
           INTEGER FUNCTION TFS_MAIN( mhitsh, mhith, mhit,
```

```
     >                           mkineh, mkine, parh, par,
     >                           bmparh, bmpar, ctrlh, ctrl,
     >                           auxhh, auxh, tphith, tphit )

DESCRIPTION
     TAS implementaion of the FST standalone program. Not all the
     functionality of the original program could be preserved.


INPUT ARGUMENTS
      mhitsh - header for MCT mhits_tpc table
      mhits  - rows of MCT mhits_tpc table
      mkineh - header for MCT mkine table
      mkine  - rows of MCT mkine table
      harh   - header for TFS fspar table
      par    - rows of TFS fspar table
      bmparh - header for TFS bmpar table
      bmpar  - rows of TFS bmpar table
      ctrlh  - header for TFS ctrl table
      ctrl   - rows of TFS ctrl table
      auxhh  - header for TFS auxhit table
      auxh   - rows for TFS auxhit table
      tphith - header of TCL tphit table
      tphit  - rows of TCL tphit table


OUTPUT ARGUMENTS
      values in tphit table
      values in auxh  table


AUTHOR
     Peter G. Jones  LBL  (510)-486-5436


CREATION DATE
     04-Mar-1993 Copyright 1993    Lawrence Berkeley Laboratory


SEE ALSO
     tfs(1)

STAR Reference Manual     Last change:                          1
```

Ideas to be implemented later

Some ideas for man page generation were proposed which are useful but require considerably more work than the present implementation of *mangen*. The ideas listed below are candidates for future work:

1. Including variable declarations of the routine which would be suitably commented for clarity:

   ```
   Integer         ncuts           ! number of cuts applied to data
   Real            a               ! a very real variable
   Character*8     string          ! hang 'em high
   ```

   This saves the effort of reproducing this information in the `INPUT ARGUMENTS` and `OUTPUT ARGUMENTS` headings.

2. Including special comments which can be found within the code but not in the routine template.