

FEAP - - A Finite Element Analysis Program

Version 7.4 Installation Manual

Robert L. Taylor
Department of Civil and Environmental Engineering
University of California at Berkeley
Berkeley, California 94720-1710
E-Mail: rlt@ce.berkeley.edu

January 2002

Contents

1	Introduction	1
2	UNIX Installations	2
2.1	Installation using RCS	2
2.1.1	Editing <code>makefile.in</code>	2
2.1.2	Editing <code>makefile</code>	3
2.2	Installation without RCS	4
2.2.1	Editing <code>makefile.in</code>	4
2.2.2	Editing <code>makefile</code>	5
2.3	Final installation]	5
3	Windows Installation	6
3.1	Build of Libraries	6
3.2	Build of Executable	8
3.3	Alternate Windows graphics form	9

Chapter 1

Introduction

The source files for the *FEAP* system are delivered on one CD-ROM. In addition, the disk contains printable files (using Acrobat Reader) for the manuals. The program is furnished under license by the University of California, Berkeley. It is for use by the licensee only and may not be redistributed in any form to others without prior authorization by the University of California, Berkeley.

It is recommended that a directory with the name 'feap', or similar be created and all information on the CD-ROM copied into this directory. The source files will reside in a directory with the name 'ver74'. This directory should contain the subdirectories: 'contact', 'elements', 'f77', 'f90', 'hp', 'ibm', 'include', 'main', 'maintain', 'plot', 'program', 'unix', 'user', 'windows' and 'window2'.

An executable version and/or an archive (library) file(s) together with the files 'feap74.f' (main program) and 'contact.f' (dummy file to eliminate contact module) on a server to be used within the licensed unit. In addition any user files in the directory 'user' (e.g., 'elmt01.f', 'umacr1.f', 'umesh1.f', 'usetm1.f', etc.) may be posted.

The routine 'feap74.f' may be modified to increase size of available work space and set other parameters as necessary.

Please report any installation problems by e-mail to: feap-help@vulture.ce.berkeley.edu.

Chapter 2

UNIX Installations

The build in a unix environment (including Linux) is controlled by the data contained in files `makefile` and `makefile.in` located in the directory "ver74".

If your system has a revision control system (RCS) it is highly recommended that *FEAP* source files utilize this feature. This permits changes to the program without losing any previous versions. This is a basic decision which must be made before proceeding with additional installation steps.

Two options are available to set up the program: With RCS and without RCS. We discuss each in turn below.

2.1 Installation using RCS

Change directories until you are in `ver74`.

2.1.1 Editing `makefile.in`

It is necessary to edit the file `makefile.in` as indicated below. Note that comments in this file are set by placing the character `#` in the first column.

Edit file `makefile.in` as follows:

1. In section *Which compilers to use* set the name of your Fortran compiler after `FF =` (This is currently set to `g77`). Also set the name of your C compiler after `CC =` (currently set to `gcc`).

2. In Section *Source Types* set `FSOURCE = RCS/` and `CSOURCE = RCS/`. You may leave the `#FSOURCE =` and `#CSOURCE =` as these are treated as comments.
3. In the section *Source Extender* set `FEXT = f,v` and `CEXT = c,v`, again you may leave the `#FEXT = f` and `#CEXT = f` lines as comments.
4. Generally, no options are needed for `FOPTIONS =` or `COPTIONS =;` however, if you experience difficulties some may need to be inserted.
5. In section *What options to be used by the loader* a path is set for standard installations of the X-library. If a non-standard installation is made some changes may be required.
6. In section *Location of feap include files* place the **full** path to where your installed the *FEAP* include files after `FINCLUDE =`. **This must be changed from that delivered.**
7. In section *What archiving to use* standard options are given. Usually no change is necessary.
8. In section *Archive name* place the **full** path to where your `ver74` directory is located. Placing elsewhere will require the makefile in the `main` directory to be changed.

2.1.2 Editing makefile

It is necessary to edit the file `makefile` to select appropriate subroutines for either a Fortran 77 compiler (i.e., `g77` in Linux) or a Fortran 90 (or 95) compiler. Make changes in the two sections indicated below:

1. In section `install:` place a comment in the first column (i.e., using a `#`) for the lines which contain the compiler which is **not** to be used. That is, if you will use a Fortran 77 compiler place the comments before the lines which read

```
(cd f90; co makefile ; make checkout)
(cd f90; make install)
(cd f90; make clean)
```

whereas if you will use a Fortran 90 or 95 compiler place the comments on the lines

```
(cd f77; co makefile ; make checkout)
(cd f77; make install)
(cd f77; make clean)
```

2. This is optional as it will not be used with an RCS installation. In section `install_norcs`: place a comment in the first column for the line which corresponds to the compiler **not** to be used: For use of a Fortran 77 compiler place the comment on the line:

```
(cd f90; make install)
```

whereas for a Fortran 90 or 95 compiler place the comments on the line

```
(cd f77; make install)
```

After editing the files `makefile.in` and `makefile` enter the command `make rcs`. All the files should be deposited in subdirectories named `RCS` for each of the source subdirectories (e.g., `plot`, `program`, etc.). Check that this has occurred.

If the RCS is successful proceed to instructions in Section 2.3.

2.2 Installation without RCS

Change directories until you are in `ver74`.

2.2.1 Editing `makefile.in`

It is necessary to edit the file `makefile.in` as indicated below. Note that comments in this file are set by placing the character `#` in the first column.

Edit file `makefile.in` as follows:

1. In section *Which compilers to use* set the name of your Fortran compiler after `FF =` (This is currently set to `g77`). Also set the name of your C compiler after `CC =` (currently set to `gcc`).
2. In Section *Source Types* set `FSOURCE =` and `CSOURCE =` . You may place comments and leave the statements `# FSOURCE = RCS/` and `# CSOURCE = RCS/`.
3. In the section *Source Extender* set `FEXT = f` and `CEXT = c`, again you may leave the `#FEXT = f,v` and `#CEXT = f,v` lines as comments.
4. Generally, no options are needed for `FOPTIONS =` or `COPTIONS =`; however, if you experience difficulties some may need to be inserted.

5. In section *What options to be used by the loader* a path is set for standard installations of the X-library. If a non-standard installation is made some changes may be required.
6. In section *Location of feap include files* place the **full** path to where you installed the *FEAP* include files after `FINCLUDE =`. **This must be changed from that delivered.**
7. In section *What archiving to use* standard options are given. Usually no change is necessary.
8. In section *Archive name* place the **full** path to where your `ver74` directory is located. Placing elsewhere will require the makefile in the `main` directory to be changed.

2.2.2 Editing makefile

It is necessary to edit the file `makefile` to select appropriate subroutines for either a Fortran 77 compiler (i.e., `g77` in Linux) or a Fortran 90 (or 95) compiler. Make changes in section:

1. In section `install_norcs:` place a comment in the first column for the line which corresponds to the compiler **not** to be used: That is, for use of a Fortran 77 compiler place the comment on the line:

```
(cd f90; make install)
```

whereas for a Fortran 90 or 95 compiler place the comments on the line

```
(cd f77; make install)
```

No other changes should be required, proceed to final installation as described next.

2.3 Final installation]

To complete installation enter the command `make install`. Each subdirectory should be processed and the compiled object files placed in the archive named in the `makefile.in`. A successful compilation should deposit the executable (named `feap`) in the subdirectory `main`.

If errors occur it is necessary to correct them and then recompile the program using the command `make install`.

Chapter 3

Windows Installation

An executable version of *FEAP*, including all graphics options, may be built using Compaq Visual Fortran.

Generally, it is desirable to place parts of the program into separate libraries and then finally build a main (executable) program. For example, a build with libraries named *program*, *plot*, *element*, and *contact* keeps basic parts of the program in clearly identifiable parts. A main program called *feap* may then be constructed which includes these libraries. However, alternate names and combinations may be selected. Below, a build is described for the names given above.

3.1 Build of Libraries

The following steps may be used to build the necessary libraries for the *FEAP* program:

1. Open the Developer Studio.
2. Under *File* select *New*. (N.B. Options to be selected are shown in italics).
 - (a) Under *Projects* tab select *Fortran static library*. **Do not select a dynamic link library (DLL)**.
 - (b) In *location* window set path to location for build files. The path must exist, if not use standard Windows steps to create the folder before doing this step.
 - (c) In *Project name* assign a library name (e.g., program). (N.B. Items to be selected and named by the user are indicated by underlines).
 - (d) Press *OK* button to start (N.B. small upper window should now have the notation **Workspace program**).

3. Under *Build*:
 - (a) Select *Set Active Configuration* and choose between *Release* and *Debug* (generally I use *Release* for most builds).
4. Under *Project* select *Project Settings*:
 - (a) Choose *Fortran* tab and set Category window to *Preprocessor*.
 - (b) In *INCLUDE and USE paths* window insert the path to where the include files are located. (The path will generally be set when you copy the program - e.g., c:
feap7
ver74
include). (N.B. This step is essential in getting any compile to work!)
 - (c) Press *OK* button to finish settings.

WARNING! STEPS 3 and 4 must be set in the sequence shown above. In particular if a change between *Release* and *Debug* is made it is necessary to set the *INCLUDE* path again.

5. Under *Project* select *Add to Project* which causes a pop-up window to appear. Select *Files* which will pop-up another window called *Insert files into project*. Use the *Look in* window to select the folder where source programs are located and find *feap7* folder. The select *ver74* (double click on folder button will change path), followed by *program*. If *Files type* window is set to *Fortran files(*.for,*.f90,...)* all the files to be compiled will appear in the large window. To select all files place mouse cursor over last file in folder and while holding the "Shift" key press the left mouse button. All files should now be highlighted. Press *OK* button to have highlighted files placed in project.

N.B. Instead of using the *Look in* window to find directories, it is possible to use the *Up one level* button to traverse the folder structure to locate where source files are located.

6. Repeat step 5 for the source folder names: F90 and user. DO NOT include files from any other folder.
7. Under *Build* tab select *Build program.lib* (or name you selected for this project or *Rebuild all*).

Compiler should process each file in the project and finish with a statement: "program.lib - 0 error(s), 0 warning(s)". If errors are present changes are necessary. First thing to ensure is that the path to the *INCLUDE* files is properly set (see step 4. above).

8. Repeat Steps 2 to 7 for (you do not have to set the include path again):

- (a) A library named plot which contains files in directory `c :`
 - `feap7`
 - `ver74`
 - `plot` and `c :`
 - `feap7`
 - `ver74`
 - `windows.`
- (b) A library named element which contains files in all the subdirectories of `c :`
 - `feap7`
 - `ver74`
 - `elements` [i.e., "frame", "material" (and its "small" and "finite" subdirectories), "shells", "solid2d", "solid3d", "thermal"].
- (c) A library named contact which contains files in all the subdirectories of `c :`
 - `feap7`
 - `ver74`
 - `contact` [i.e., "main", "nts2d", "nts3d", "ptpnd", and "util"]. N.B. This library is optional if you do not intend to solve any contact problems [see Step 4 below].

At this stage the libraries "program.lib", "plot.lib", "elements.lib" and "contact.lib" for the *FEAP* program have been built. It is now necessary to build the final executable program.

3.2 Build of Executable

The following steps may be used to build an executable for the *FEAP* program:

1. Under *File* select *New*.
 - (a) Under *Projects* tab select *Fortran Standard Graphics* or *QuickWin Application*.
 - (b) In *location* window path to location for build files should still be set for the library builds. This is ok, but can be changed if you wish (recommend no change for this). The path must exist, if not use standard Windows steps to create the folder before doing this step.
 - (c) In *Project name* assign a program name (e.g., feap).
 - (d) Press *OK* button to start (N.B. small upper window should now have the notation *Workspace 'feap'*).

- (e) New pop-up window gives choice between a QuickWin and a Standard Graphics mode. Select *QuickWin* and then press *Finish*.
2. Repeat steps 3 and 4 above which are now applicable to this project. (e.g., must set *Release* or *Debug* mode and path for INCLUDE files).
 3. Under *Projects* tab select *Settings*, followed by the *Link* tab. In *Category* window select *Input*. In *Ignore libraries* window add ,libc.lib (with no blanks before the ","). N.B. Leave the existing entry (dfconsul.lib).
 4. Use *Project* tab and select *Add to Project*. Then select *Files* and select the folder *Main* (see step 5 above). Add the main program file 'feap74.f' to the project. If you did not build the contact library also include the file 'contact.f'.
 5. Use *Project* tab and select *Add to Project*. Select *Files* tab and go to folder where library "program.lib" is located. This is the path you set in the first build followed by the name of the library (e.g., "program") and either *release* or *debug* depending on which you built. Nothing will appear in the main window until a selection is made in the *Files of type* window is set to: *Library Files (lib)*. It may be necessary to scroll to find this or just enter "l" in the window and scrolling will occur automatically.

Add the program.lib to the project by placing the mouse over the name in the window and double clicking.

Repeat for all libraries built above (e.g., "plot", "elements", "contact").

6. Under *Build* tab select *Build feap.exe* (or name you selected for this project or *Rebuild all*. Compiler should process each file in the project and finish with a statement: "feap.exe - 0 error(s), 0 warning(s)". If errors are present changes are necessary. First thing to ensure is that the path to the INCLUDE files is properly set (see Step 4. in the instructions for building the libraries).

Program is ready to use. The executable will be placed in the *release* or *debug* directory where the build of the executable was designated (see Step 3 and 4 in Section 3.2). It is usually desirable to place an executable icon on the desktop.

3.3 Alternate Windows graphics form

An alternate to the graphics structure may be built by using the source files in the directory `window2` instead of those in `windows`. Note that the remaining files in `windows` must still be included. The alternate form uses the source files from `window2` when building the executable instead of the files with the same name that are in the `windows`

directory (the files to be included are: `pfullscr.f`, `plopen.f`, `plstrt.f`, `plstdos.f`, `pwopn.f`).

Bibliography