

## MPW FORTRANS

With the introduction of support for Pascal-like records in Absoft MacFortran II version 3.1, the use of the Language Systems (LF) and Absoft (AF) compilers with FaceWare modules has become very similar.

**Include Files** The file "FaceStorXY.inc" declares command constants and the fRec shared global record type used by Facelt and ViewIt. A similar file is shipped with other FaceWare modules:

"DrawStorXY.inc" with DrawIt, "DBaseStorXY.inc" with DocuBase, etc. The contents of such files can be made available to program units via include statements:

```
include 'FaceStorLF.inc' [LF]
include "FaceStorAF.inc" [AF]
```

All program units which directly access elements of such files should also contain such include statements. Alternatively, each of these compilers has a provision for declaring such files "globally": LF supports a "!!G" directive for precompiled globals, and AF supports "GLOBAL DEFINE" which can be used to include constants and record types just once at the top of a source file.

The "FaceStorXY.inc" file contains the definition of the fRec record type, but does not allocate space for the record. To do this, fRec should be declared as a common block in all program units where it is used (LF and AF are same in this case):

```
record/ FaceRec /fRec
common/FaceStuff/fRec
```

The file "FaceProcLF.inc" contains routines which are required for jumping to FaceWare modules. To make these routines available to your program, insert an include statement such as the following somewhere in your main program file:

```
!!! FaceProcLF.inc
[LF]
include "FaceProcAF.inc"
[AF]
```

**Default Output Windows** The default output windows supported by LF and AF are not compatible with Facelt, but you probably won't miss them if using Facelt and ViewIt. For both compilers, the trick to turning off their default windows is to link a "stub" to your application. For LF, the file to link is "OutpWStubs.o", and for AF it is "f77stubs.o". See our demo program make files for examples of the use of these stubs to kill the default windows.

The big advantages of the default output windows are (1) "standard I/O" is redirected to them, and (2) they help support event-checking during processing. Both of these features, however, can be added to Facelt-based programs by taking advantage of the fact that both compiler makers have documented the special routines in their standard libraries that can be replaced to create "custom" output environments with these features.

For both the AF and LF compilers, we have created a Facelt-based environment that completely replaces their default window environments. This "FRWE" (FaceWare Runtime Window Environment) is shipped with the EditControls product (since it makes use of the TextControl module), and includes complete FORTRAN source code.

**Upgrading MacFortran II** If upgrading from a previous version (< 3.1) of the MacFortran II compiler, the following changes to existing programs will be required:

- The declaration of the fRec record as a common block was removed from the "FaceStorAF" file so that this file only includes constant and record type declarations. This makes it possible to include this file in a "GLOBAL DEFINE" block, but also means that any program unit that uses fRec must declare it explicitly (see above).

- All references to elements of fRec must now be preceded by the fRec record name: "fRec.uName" (new way) vs. "uName" (old way). For the sake of backward compatibility, however, we did not change the TextIt, GrafIt, or ShowIt include files to use records instead of equivalence blocks.

- Support for receiving true "values" when called from Pascal was added, making it much easier to write dialog user item procedures and ViewIt override procedures (see "vDemoAF" for an example).

- The "f77stubs.o" mentioned above must now be linked to turn off use of the default output window.

- "-model far" must be included in link instructions.

- New toolbox include file dependencies mean that you'll have to examine the "INCLUDE File

Dependencies" Appendix in the Absoft manual to be certain that you include the necessary files.

#### Other FORTRAN Programming Tips

- When practical, use "implicit none".
- Both source and data files should end with a carriage return.
- When Facelt is first called, you have the option of having it display a "deck of cards" cursor whenever your program has control. This is very useful when, for example, a FORTRAN error occurs within your program since you immediately know, by the form of the cursor, that the error is somewhere in your code and not within Facelt.
- If your program works under some startup conditions, or on some Macs, but not others, then look first for an uninitialized variable. Variables are not zeroed at launch time, and will acquire the value of whatever garbage is sitting in memory unless you take the trouble to initialize them.