5. How do I meet one?

Believe it or not, all of the latest FaceWare modules can be called with a single dispatching procedure. This procedure is the "FaceIt" procedure and is made available to programs via an include file or by some other appropriate means.   Historically, the FaceIt procedure was once used to call only the FaceIt module (hence its name), but it was soon found that programmers preferred to deal with one procedure to call all modules and so the FaceIt procedure was modified to do this.

The FaceIt dispatching procedure accepts 6 parameters, and has the same form when used with different languages:

```
FaceIt(xPtr,command,a,b,c,d);        Pascal
FaceIt(xPtr,command,a,b,c,d);        /* C, C++ */
call FaceIt(xPtr,command,a,b,c,d)    !FORTRAN
```

where "xPtr" indicates the FaceWare module being called, "command" is a command understood by that module, and a, b, c, and d are 4-byte integer parameters passed with the command.   If the command is supported by FaceIt or ViewIt, then 0 or nil can be passed as the "xPtr".   If using C++, then trailing parameters equal to 0 can be dropped.

The use of a single dispatching procedure might appear limiting, but we have not found this to be the case.   First, extensive use is made of the four parameters, including the passing, when necessary, of handles or pointers to other structures in memory.   Second, commands supported by FaceWare modules are often "high-level" commands that replace many other toolbox calls, making your code much simpler and minimizing the number of calls made to the FaceIt dispatching procedure.   Third, use of a single FaceIt procedure establishes a clear distinction between the main program code and its use of FaceWare modules.   Fourth, the use of a single procedure makes it much easier for us to add new commands, enhance existing commands, and to support a wide range of Mac programming environments.