

About Compilers 2.2

©FaceWare 1986-94. All Rights Reserved.

Introduction

The following notes describe language- and compiler-specific issues encountered when programming with FaceWare modules, and assume that you have read the "About FaceWare" program on the Utilities disk, or the "Startup" topics in ViewIt's on-line documentation.

A Blessing & A Burden The fact that FaceWare modules are language- and compiler-independent is a blessing for the programmer, but a burden for the module developer who must create example programs and interface files that are language- and/or compiler-specific. This is balanced by the fact that most modules have relatively simple programming interfaces, making for simple example programs that are easily translated.

FaceWare modules distributed by FaceWare always include at least one example or "demo" program written in C, Pascal, and FORTRAN. The C and Pascal programs are typically created using the Symantec or Metrowerks C and Pascal compilers, but the source can also be compiled using the MPW C and Pascal compilers (discussed below). The FORTRAN programs are created using the Absoft and Language Systems FORTRAN compilers.

Language Differences For the most part, programs written in C, Pascal, or FORTRAN can be directly translated from one language to another on a line-by-line basis. (If you don't believe this, compare the example programs shipped with each FaceWare module.) Most differences between languages that affect FaceWare programmers involve just four areas: numbers, pointers, strings, and support for jumping to code resources. The latter problem is handled by providing pre-done, compiler-specific "StorXY" and "ProcXY" include files. Other topics in this program review the use of numbers, pointers, and strings with different languages.

Compiler Versions & Abbreviations FaceWare documentation uses "XY" as a generic compiler abbreviation and the following for specific compilers:

LP = Symantec THINK Pascal 4.0 (617-275-4800)
LC = Symantec THINK C 6.0
SC = Symantec C++ 6.0
MP = Metrowerks Pascal 1.0d4 (514-747-5999)
MC = Metrowerks C 1.0a1
MC+ = Metrowerks C++ 1.0a1
WP = Apple MPW Pascal 3.3 (APDA: 800-282-2732)
WC = Apple MPW C 3.3
LF = Language Systems MPW FORTRAN 3.3 (703-478-0181)
AF = Absoft MPW MacFortran II 3.2 (313-853-0050)

where the version numbers correspond to the current versions as of January 1994, and the company phone numbers are included in case you need to order an upgrade or get technical support.

Compiler upgrades have always presented us with problems. If we ship examples using the latest version of a compiler, then this causes problems for programmers who have not yet upgraded their compilers. If we ship examples using earlier versions, then programmers who have upgraded their compilers complain that we should upgrade our examples! This is compounded by the fact that successive compiler versions are often not compatible with one another, or address bug fixes and compatibility issues that make it difficult for us to not upgrade our example programs.

The best solution is for you to use the latest version of your compiler. If using an earlier version, our example programs may or may not compile without modification. Call us if you have any problems.

MPW C & Pascal Programmers The latest versions of THINK C & Pascal are very nearly source-compatible with the MPW compilers. When compiling the THINK Pascal examples with MPW Pascal, remove the "\$I-" directive which is used in THINK programs to turn off automatic toolbox initializations.

Both MPW C and Pascal users will also need to expand the include file lists to include a larger number of toolbox-related include files. With MPW C, for example, add the following lines to the FaceStorLC file,

```
#include <QuickDraw.h>
#include <Menus.h>
#include <Events.h>
#include <Files.h>
#include <Controls.h>
```

and the following lines to the FaceProcLC file,

```
#include <Resources.h>
#include <ToolUtils.h>
#include <OSEvents.h>
#include <Fonts.h>
#include <Windows.h>
#include <Menus.h>
#include <TextEdit.h>
#include <Dialogs.h>
#include <Memory.h>
```

The statements in FaceProcLC involving "thePort" and C-to-Pascal string conversion are also slightly different:

```
InitGraf(&qd.thePort);
```

```
...
```

```
c2pstr(fRec.uName);
```

If using MPW C++, then you'll also need to use "extern" to force C++ to accept references within C++ files to include files that contain C-based prototypes:

```
extern "C"
```

```
#include "FaceStorLC.h"
```