About Compatibility Guidelines

Apple's **Compatibility Guidelines** show you how to write applications that have the greatest chance of operating on any Macintosh computer, regardless of its hardware components or available system software, managers, and device drivers. They also address how you can take advantage of features that are new to system 7.0 in ways that are least likely to cause problems for users who are not running system 7.0. In a word,
The **Compatibility Guidelines** provide as much advice as possible to help you achieve maximum compatibility for your applications on all Macintosh computers, including those running system 7.0.

System 7.0 provides the most important test of software compatibility since the introduction of the Macintosh II, and you must understand how you may need to revise your current applications so that they operate correctly with this new system software. Fortunately, if you have followed the guidelines given previously, your applications stand a very good chance of working correctly in system 7.0 or later systems without any modification whatsoever. However, system 7.0 introduces many new features and capabilities that you may wish to use in your applications. The
Compatibility Guidelines provide a number of additional guidelines to help you take advantage of those features while retaining compatibility with previous system software.

Compatibility Guidelines pages discuss several aspects of writing software that is compatible with all Macintosh computers:

- what can cause compatibility problems and how in general to avoid those problems
- how to update your application to take maximum advantage of new features of system 7.0
- how to write software that can be easily modified for use in other regions
- how to write applications that execute under A/UX, Apple's version of the UNIX operating system
- how to determine what software and hardware features are available on a particular machine

The discussion of revising applications to take advantage of the new capabilities of system 7.0 also includes details about several new features of the **Dialog Manager** and **Menu Manager**, including

- the new pop-up menu control definition
- the system menus
- movable modal dialog boxes
- new **Dialog Manager** routines to count and manipulate items in dialog boxes

Compatibility Guidelines also describes the **Gestalt Manager**, a set of three new Operating System functions that provide applications with a simple

and efficient method for determining what software and hardware features are available on a given machine. You need to use the **Gestalt Manager** if your application takes advantage of particular hardware components (such as a floating-point unit) or software modules (such as **Color QuickDraw**) that are not available on all Macintosh computers. Your software can also use the **Gestalt Manager** to inform the Operating System (and hence other applications) that it is present in the current environment.

The **Gestalt Manager** is available in system software versions 6.0.4 and later. Your development system may supply code that allows you to call **Gestalt** on earlier system software versions; check the documentation provided with your development system to see if this is possible. Of course, because you cannot use **Gestalt** to determine if the **Gestalt Manager** itself is present, you must do that in some other way; one such method is illustrated in **Determining Whether Gestalt Is Available** under the section, **Using the Gestalt Manager** .

You need to read **Compatibility Guidelines** if you are interested in writing applications that execute on as many Macintosh computers as possible or under alternate operating systems. In particular, if you wish to enhance an existing product so that it supports new features of system 7.0 but executes correctly in earlier versions of system software, or if you wish to write a new product that executes only in system 7.0, see **Running in System 7.0**. Read the sections on the **Gestalt Manager** if you need to take advantage of specific software or hardware features that may not be present on all versions of the Macintosh, or if you wish to inform other applications of the presence of your application in the operating environment.

If you want your applications to run in system software versions earlier than 6.0.4 (where the **Gestalt** function is not available), you should be familiar with **Environs** , under **Operating System Utilities**, and the **SysEnvirons** function, discussed in **Compatibility Guidelines**. Both **Environs** and **SysEnvirons** perform the kind of function that **Gestalt** performs-they allow you to determine what features are available on a specific machine. For reasons outlined later, however, you should not use either of these routines if the **Gestalt** function is available.

Unfortunately, no single section can provide all the information you need to achieve the greatest possible compatibility for your applications. Most of the subsequent information contains numerous warnings and guidelines that you should heed if you wish to increase the likelihood that your applications will execute correctly on all members of the Macintosh family and under alternate operating systems such as A/UX. The Memory Management section, for example, contains a fuller account of 32-bit clean programming than is given here and is essential reading for all developers.

The **Worldwide Software Overview** gives complete details on the **Script Manager**, which can help you write applications that are compatible worldwide. Similarly, the guidelines given on writing A/UX-compatible Macintosh programs summarize and complement, but do not replace, the discussion in the separate publication *A/UX Toolbox: Macintosh ROM Interface*. So the complete story on Macintosh software compatibility does not end with the **Compatibility Guidelines**, but it does begin there.