



# PippinFinder

version 002

Please send questions and comments via e-mail to [pippindev@apple.com](mailto:pippindev@apple.com).

1996, Apple Computer, Inc. All rights reserved. Apple, Macintosh, and Pippin are trademarks of Apple Computer, Inc. All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Every effort has been made to ensure the accuracy of information in this document. However, Apple assumes no responsibility for the accuracy of the information. Product information is subject to change without notice. Mention of non-Apple products is for informational purposes only, and constitutes neither an endorsement nor a recommendation. Apple assumes no responsibility with regard to the selection, performance or use of these products. All understandings, agreements, or warranties, if any, take place directly between the vendors and the prospective users.

---



The **PippinFinder** file is only to be used with CDs that have a multiple-application interface. Since most titles developed for Pippin will involve only a single application launched immediately at start-up, the **PippinFinder** file is not used and therefore should not be placed in the **System Folder**. Even with multiple application titles, the **PippinFinder** does not provide the interface for launching the applications within a title CD; the interface is provided by a **Pippin Launch** file.

The **PippinFinder** file, when used with multiple-application titles, is used to search and find the **Pippin Launch** file. The **Pippin Launch** file contains a resource of type PCde with an ID of zero. When launching, the PCde code resource is loaded and executed, and thus provides an interface for multiple application selection.

The API for the **Pippin Launch** code module is as follows:

```
pascal OSErr PippinCode(FSSpecPtr fileToLaunch, Boolean* restart);
```

The `PippinCode` module works is that when it is called, it expects to be returned an `FSSpec` in the `FSSpecPtr`. Another varred parameter, called `shutdown` is used to let **PippinFinder** know that the code module, as requested by the user, wants to restart the machine.

With version 1.0 of **PippinFinder**, a return of any error will result in **PippinFinder** hanging in an endless loop. By returning an invalid `FSSpec` will also result in **PippinFinder** hanging in the same loop. It is therefore vital that the code module does adequate error checking.

To help limit memory leaks before the code module is called, a heap of 2Meg is created in temporary memory space and a zone is initialized there. If you need more memory, use additional temporary memory space. When returned from the `PippinCode` module, the handle which contains the 2Meg code module is released.

This describes the total functionality of **PippinFinder**. Apple is also providing a `PippinCode` module that will be described in a separate Pippin Technical Note. For functionality beyond what Apple's `PippinCode` provides, developers are directed to write their own PCde code modules.