

Run-Path Dependent Libraries

Published: 2012-07-23 · Archived: 2026-04-06 01:13:58 UTC

Application users often need to organize their applications within their file systems in a way that makes them more efficient to use. This capability is easy to provide for a single binary because the location of its dependent libraries is easy to determine: They may reside at a standard location in the file system or at a location relative to the binary itself. However, when dealing with a set of applications that share dependent libraries (for example, in an application suite), providing users the ability to relocate the suite directory is more difficult: Either the suite's dependent libraries must be located outside the suite directory, or each of the suite's executables must be linked taking into account its position within the suite. In OS X v10.5 and later the linker and dynamic loader offer a simple way of allowing multiple executables in an application suite directory to share dependent libraries while providing the suite's users the option of relocating the suite directory. Using *run-path dependent libraries* you can create a directory structure containing executables and dependent libraries that users can relocate without breaking it.

A *run-path dependent library* is a dependent library whose complete install name is not known when the library is created (see [How Dynamic Libraries Are Used](#)). Instead, the library specifies that the dynamic loader must resolve the library's install name when it loads the executable that depends on the library.

To use run-path dependent libraries, an executable provides a list of run-path search paths, which the dynamic loader traverses at load time to find the libraries.

This article describes how to create run-path dependent libraries and how to use them in executables.

Creating Run-Path Dependent Libraries

To create a run-path dependent library, you specify a run-path–relative pathname as the library's install name. A *run-path–relative pathname* uses the `@rpath` macro to specify a path relative to a directory to be determined at runtime. A run-path–relative pathname uses the following format:

```
@rpath/<path_to_dynamic_library>
```

These are examples of run-path–relative pathnames:

- `@rpath/libMyLib.dylib`
- `@rpath/MyFramework.framework/Versions/A/MyFramework`

A *run-path install name* is an install name that uses a run-path–relative pathname. You specify a run-path install name while creating the dependent library using the `gcc -install_name` option. See the `gcc` man page for more information.

Using Run-Path Dependent Libraries

To use run-path dependent libraries (those using run-path install names) on an executable, you specify one or more run-path search paths with the `ld -rpath` option (each `-rpath` clause specifies one run-path location). When the dynamic loader (`dyld`) loads the executable, it looks for run-path dependent libraries in the run-path search paths in the order in which they were specified at link time.

This is an example of a list of run-path search paths:

```
@loader_path/../Library/Frameworks
```

```
@loader_path/../Library/OpenSource
```

```
/usr/lib
```

Source: <https://developer.apple.com/library/archive/documentation/DeveloperTools/Conceptual/DynamicLibraries/100-Articles/RunpathDependentLibraries.html>