

# OPENSTEP 4.2 Release Notes: Compiler Tools

This file contains release notes for the 4.2 release of the Compiler Tools. It contains information about the following topics:

- The NeXT Mach-O GNU-based assemblers
- The NeXT Mach-O static link editor
- The NeXT Mach-O dynamic link editor
- Mach-O object tools (**nm**, **otool**, and so on)

## Notes Specific to Release 4.2

### New Features

There are no new features for the 4.2 release of the compiler tools.

## Notes Specific to Release 4.1

## New Features

There is only one new feature for the 4.1 release:

- The dynamic linker now has the environment variable `DYLD_FRAMEWORK_PATH` to better support the development of frameworks. See the man page for details.

## Notes Specific to Release 4.0

### New Features

#### Dynamically Linked Shared Libraries

The compiler tools now allow you to build and develop dynamic shared libraries and support the programs that use these libraries, including programs that use bundles. The tools to build or use dynamic shared libraries are in the 4.1 updates for the m68k, i386, sparc, and hppa target architectures.

All object files that are part of a dynamic shared library or that are to be in an executable should be compiled with the **-dynamic** flag. The **-dynamic** flag is now the default. Executables using shared libraries must also be linked with this flag when using **cc(1)** or **ld(1)** and must be using **crt1.o** (this is done automatically with

`cc(1)` and the **-dynamic** flag). To build a dynamic shared library, use `libtool(1)` with the **-dynamic** option; typically you also specify the **-install\_name** *library\_name* option as well as other options (see the `libtool(1)` man page).

The dynamic linker is becoming more fully-featured. It contains some programmatic support for runtime loading (but as of yet no unloading or replacing).

The static link editor uses the symbol tables of dynamically-linked shared libraries to cause modules for undefined symbols to be pulled in from static libraries and to check for undefined symbols. The `ld(1)` flags for undefined checking **-undefined {error, warning, suppress}** can be used; the default is to treat undefined symbols as errors. This default holds for dynamically-linked shared libraries and bundles. In these cases the dependent libraries should be listed if available. If not available then **-U *\_symbol*** or **-undefined {warning, suppress}** can be used until they are.

## Using and Building Dynamic Shared Libraries from Project Builder

The 4.1 version of Project Builder supports a library project type which by default is a dynamic shared library and a framework project type which by default contains a dynamic shared library. Project Builder supports the use of dynamic libraries in other project types. To build static libraries, un-comment the `LIBRARY_STYLE` definition in the project makefile (`LIBRARY_STYLE` is set to `STATIC` and commented out).

## Tools that are complete

- The assembler, link editor, and **otool** support position-independent code for the

m68k, i386, sparc, and hppa architectures. The link editor, assembler and **otool** fully support indirect undefined references using symbol pointers and symbol stubs. And the assembler now fully supports Mach-O files.

- The dynamic link editor now has the first level of support for runtime loading. Currently only loading of MH\_BUNDLE files is supported. Comments in the header file **/NextDeveloper/Headers/mach-o/dyld.h** describe what is not yet implemented.

### **Tools that are not complete**

- The dynamic link editor now has the first level of support for runtime loading. Currently only loading of MH\_BUNDLE files is supported. Comments in the header file **/NextDeveloper/Headers/mach-o/dyld.h** describe what is not yet implemented.
- The tool **segedit(1)** does not have support for dynamic shared libraries.

## **The NeXT Mach-O GNU-based Assemblers**

The NeXT assembler now fully supports Mach-O files with the ability to create arbitrary sections with the **.section** and **.zerofill** directives. Contents of sections now correctly reflect the assembly code with respect to the section alignment and no unnecessary padding is added.

### **Major New Features**

Support for position-independent code through the use of a new relocatable form of an expression: "*add\_symbol - subtract\_symbol + offset*" where *add\_symbol* and *subtract\_symbol* can be defined in different sections.

## Documentation

### Assembler Manual

The assembler manual has been updated to reflect support for dynamically-linked shared libraries. It contains appendices of the instructions for the i386, M68K, and PA-RISC processor architectures. An RTF version of the Assembler manual is in:

**`/NextLibrary/Documentation/NextDev/Reference/DevTools/Assembler`**