

3.3 Release Notes: Writing Loadable Kernel Servers

This file contains release notes for the 3.3, 3.2, 3.1, and 3.0 releases of loadable kernel servers. Items specific to the 3.1 release are listed first, and the Release 3.0 notes follow. There are no items specific to the 3.3 or 3.2 releases.

Notes Specific to Release 3.1

New Features

The following new features have been added to the kernel-server loader since Release 3.0:

- Support for object-oriented loadable kernel servers written in Objective C.

- Support for fat Mach-O files for loadable kernel servers.

See the Driver Kit release notes for more information.

Notes Specific to Release 3.0

In Release 3.0, the following changes were made:

- Changes to the flags required when compiling loadable kernel servers
- New kernel support functions
- Changes to the kernel-server loader and its header files

For more about these and related topics, see Part 2, "Writing Loadable Kernel Servers," of the *NEXTSTEP Operating System Software* manual.

C Preprocessor Flags

Incompatible Changes

The `KERNEL_FEATURES` flag must be replaced by the new flag, `MACH_USER_API`. To compile a loadable kernel server, you must define `MACH_USER_API`, `KERNEL`, and `MACH`, as shown in Chapter 9, "Building,

Loading, and Debugging Loadable Kernel Servers," of the *NEXTSTEP Operating System Software* manual. Flags such as these are typically defined either in the `cc` command line (e.g., "-DKERNEL") or inside of source code (e.g., "#define KERNEL 1").

Kernel Support Functions

New Functions

The following are new functions that have been added to the kernel since Release 2.0.

- Functions for getting and setting clock information: **clock_value()**, **set_clock()**, **clock_attributes()**. New functions for converting between clock formats have also been added: **ns_time_to_timeval()** and **timeval_to_ns_time()**. (**microboot()** and **microtime()** are obsolete.)
- Functions for setting and canceling timeouts: **ns_abstimeout()**, **ns_timeout()**, **ns_untimeout()**. These functions replace **us_abstimeout()**, **us_timeout()**, and **us_untimeout()**, which are obsolete.
- A function for getting the task port of the kernel: **kern_serv_kernel_task_port()**. This function makes it easier for loadable kernel servers to send and receive Mach messages that have out-of-line data.

- Functions to help UNIX-style drivers implement the **select()** system call: **selthreadcache()** and **selthreadclear()**.
- Functions to help network modules get information about other network modules: **if_class()**, **iflist_first()**, **iflist_next()**.
- A function that removes a netif from the system list of netifs: **if_detach()**.

Kernel-Server Loader

New Features

The following are new features that have been added to the kernel-server loader since Release 2.0.

- A DETACH keyword has been added to Load Command processing. If this command is used then the kernel-server loader will not attempt to unload the server when it shut down. This is necessary for the correctness of some network protocols.

Changes

The following compatible changes have been made to the kernel-server loader since Release 2.0.

- Header files are now exported into **/NextDeveloper/Headers/kernserv**.

Incompatible Changes

- The **kern_loader_reply_handler.h** header file is now generated by MiG. As a result, the `KERN_LOADER_REPLY_INMSG_SIZE` and `KERN_LOADER_REPLY_OUTMSG_SIZE` macros have been replaced with `kern_loader_replyMaxRequestSize` and `kern_loader_replyMaxReplySize`, respectively.