

# **Reconfiguring the Kernel in HP-UX Release 11i Version 1.5**



**Manufacturing Part Number: 5185-6559**

**June 7, 2001**

© Copyright 2001 © Hewlett-Packard Company. All rights reserved..



# 1                    **Reconfiguring the kernel in HP-UX Release 11i Version 1.5**

The following white paper is an excerpt from Chapter 3, “Configuring a System,” in *Managing Systems and Workgroups: A Guide for HP-UX System Administrators*. It is updated here to reflect changes made in boot and system configuration for HP-UX Release 11i Version 1.5.

Other documents that are referred to in this white paper can be found at the following web site: <http://docs.hp.com>

## Reconfiguring the Kernel

For most systems, the default kernel configuration included with HP-UX will be sufficient for your needs. However, in each of the following instances you need to reconfigure the kernel:

- **Adding or removing device drivers**

The most common reason for adding or removing kernel modules is to add or remove device drivers. See *Configuring HP-UX for Peripherals* for full instructions on adding peripherals.

You may also want to remove a driver from your kernel if your system is not using any peripherals of that type. This is not required, but can be desirable if a smaller, more efficient kernel is needed.

To add or remove traditional drivers which are statically linked to the kernel, a whole kernel configuration and system reboot are required. See “Steps to Reconfigure the Kernel” on page 6, and “Steps to Perform Whole Kernel Configuration Using HP-UX Commands” on page 7 for details.

However, before you remove the driver, ensure that other drivers are not dependent on it by checking the files in the directory `/usr/conf/master.d/` for a table of driver dependencies in the section `DRIVER_DEPENDENCY`. The file `core-hpux` will have the most definitions, but other files in the directory can contain definitions as well.

If the device driver that controls the peripheral is a dynamically loadable module (a DLKM), you may be able to add or remove the drivers from the running kernel without needing to reboot. See the white paper *Managing and Developing Dynamically Loadable Kernel Modules* for instructions on adding or removing DLKM modules.

- **Modifying system parameters**

You may need to change one or more tunable system parameters. For example, to accommodate a specialized application or an exceptionally large number of users.

Historically, all tunables have been static, but as of HP-UX Release 11i, a tunable may be either static, dynamic, or automatic.

- A static tunable is one whose value cannot be changed without rebooting the system. Usually a kernel rebuild is also required.
- A dynamic tunable is one whose value can be changed without a reboot.

---

**NOTE** Although dynamic tunables are enabled on HP-UX Release 11i, they are not

- ❑ An automatic tunable is one that is constantly being tuned by the kernel itself in response to changing system conditions.

The list of dynamic and automatic tunables is continually growing. To determine which tunables are dynamic on your system, use the `kmtune` (1M) command, or see the **Kernel Configuration** portion of SAM. In SAM's **Configurable Parameters** screen, administrators can tell at a glance whether or not the value of a particular tunable can be changed without a reboot.

The tunable system parameters are edited using SAM or `kmtune`. When SAM is used to change a tunable, the administrator is informed whether or not that change requires a reboot. If no reboot is required, SAM makes the change to the tunable immediately.

For more information on dynamic tunables, see the *Dynamically Tunable Kernel Parameters in HP-UX 11i* white paper.

- **Adding certain Hewlett-Packard or third-party vendor software**

If you add certain optional add-on software, such as LAN (Local Area Network) or NS (Network Services), you might need to reconfigure the kernel. If kernel modules are installed with the software they may be traditional static kernel modules or DLKMs. Traditional kernel modules require whole kernel configuration and system reboot. DLKMs may be dynamically configured into the running kernel and may not require system reboot. Consult the manual that came with the software for installation instructions.

- **Creating a file system of a type other than JFS**

Depending on how your kernel is configured, you might have to reconfigure if you created a file system of a type other than the default file system (JFS). See *Managing Systems and Workgroups: A Guide for HP-UX System Administrators* for information on file system types.

- **Adding, removing, or modifying swap, dump, console devices or the root file system**

You will need to perform whole kernel configuration for adding or removing dump devices and modifying the location of primary swap or the system console. For information on swap space, see the section “Managing Swap and Dump” in Chapter 3, “Configuring a System” in *Managing Systems and Workgroups: A Guide for HP-UX System Administrators*.

To add, remove, or modify the root file system, you will not be able to use SAM. Instead, re-install your system or see “Creating Root Volume Group and Root and Root and Boot Logical Volumes” in Chapter 3, “Configuring a System” in *Managing Systems and Workgroups: A Guide for HP-UX System Administrators*, if you are using logical volumes.

---

**NOTE**

If you have cold-installed an HP 9000 Model T500 and you are

**Reconfiguring the Kernel**

configuring a large number of file systems (approximately 100 or more), some default table sizes in the kernel may be too small for your system to successfully boot. To boot your system, reconfigure the install kernel before the first boot. Refer to the section “Steps to Reconfigure the Kernel” on page 6 to perform this, keeping in mind that SAM is not available at this point. The following settings, although not necessarily optimal for the system, will allow the kernel to be booted:

**Table 1-1 useradd Options**

<b>Kernel Parameters</b>	<b>Default</b>	<b>Recommended Setting</b>
<i>ninode</i>	476	<b>2048</b>
<i>nproc</i>	276	<b>1024</b>
<i>nfile</i>	790	<b>2048</b>

Alternatively, you can do the following:

- Reconfigure the kernel and change the value of *mixers* to a large value, such as 200.
  - Select an appropriate bundle of SAM-tuned parameters by doing the following:
    - Open the “SAM Kernel Configuration” menu item
    - Select “Configured Parameters”
    - Pull down the “Actions” menu
    - Select “Apply Tuned Parameter Set”

For further details, refer to *Installing HP-UX 11.0 and Updating HP-UX 10.x to 11.0*.

**Steps to Reconfigure the Kernel**

You can use SAM or HP-UX commands to reconfigure the kernel.

To use SAM to reconfigure the kernel, log in as the superuser, ensure you are logged on to the machine for which you are regenerating the kernel, and start SAM. Select the **Kernel Configuration** menu item; use SAM’s online help if needed. Generally, SAM is simpler to use than the equivalent HP-UX commands.

SAM will also help you determine whether a whole kernel configuration and/or system reboot are necessary, or if a dynamic configuration of the running kernel is all you need (for example,

when adding or removing DLKMs).

Since the introduction of DLKMs, dynamic kernel configuration of the running kernel is possible when adding or removing a DLKM. Although HP-UX Release 11i Version 1.5 is not released with DLKM content, certain optional add-on HP-UX software is available as DLKM modules.

If using HP-UX commands to add or remove static traditional kernel modules, follow the instructions outlined in “Steps to Perform Whole Kernel Configuration Using HP-UX Commands” on page 7. To add or remove DLKM kernel modules, refer to the white paper *Managing and Developing Dynamically Loadable Kernel Modules*.

---

**NOTE** If you would like to permanently save the running kernel before performing whole kernel configuration for the first time, refer to “Saving a Backup Kernel Permanently” on page 10.

---

### Steps to Perform Whole Kernel Configuration Using HP-UX Commands

To use HP-UX commands to reconfigure the kernel:

1. Log in as superuser on the machine for which a new kernel is being generated. You can log in remotely from another location by using the `/usr/bin/rlogin` command.
2. Change directory to the build environment (`/stand/build`). There, execute a system preparation script, `system_prep`. `system_prep` writes a system file based on your current kernel in the current directory. (That is, it creates `/stand/build/system`.) The `-v` provides verbose explanation as the script executes.

```
cd /stand/build
/usr/sbin/sysadm/system_prep -v -s system
```

3. Use the `kmsystem` (1M) command:

```
/usr/sbin/kmsystem -S /stand/build/system
```

to view the kernel modules that were already explicitly selected for the next kernel build:

Note that additional kernel modules may be configured because of module dependencies or because they are required kernel modules as indicated in their master files.

Add absent kernel modules (like device drivers) using the `kmsystem` command. The `-c Y` option specifies the module name to be configured into the system:

```
/usr/sbin/kmsystem -S /stand/build/system -c Y driver-name
```

---

**NOTE** Direct edits to the HP-UX system description files to add or remove kernel

---

## Reconfiguring the Kernel

modules is not a supported kernel configuration interface, and is likely to introduce configuration errors. Instead, use the commands *kmsystem* (1M) and *kmtune* (1M), available since Release 11.0.

---

4. Build the new kernel by invoking *mk\_kernel* (1M):

```
/usr/sbin/mk_kernel -s /stand/build/system
```

This builds a new kernel ready for testing: */stand/build/vmunix\_test* and the associated kernel components.

5. Perform a kernel update using *kmupdate* (1M) :

```
/usr/sbin/kmupdate
```

This is a required step that will make the newly configured kernel bootable. *kmupdate* also saves the currently running kernel under a new name, */stand/vmunix.prev*, and schedules the update of the system to use the default name */stand/vmunix* for the newly configured kernel.

---

**NOTE** A whole kernel configuration *must* be followed by a kernel update step initiated with the *kmupdate* (1M) command in order to be able to successfully boot the new kernel. A configured, bootable kernel is no longer composed of a single file. Each new kernel configured with *mk\_kernel* (1M) or *config* (1M) is composed of several files, including the main kernel file, the loadable images for DLKMs, a symbol table file, and a kernel-specific KRS file. The main kernel file (for example, */stand/vmunix*) carries the name of the kernel, but is not sufficient to boot a kernel. All component files of the kernel are essential for the kernel to boot and run properly. The *kmupdate* command understands the organization of these files and prepares a directory structure accessible to the boot loader that will allow a newly generated kernel to boot and run properly.

---

**CAUTION** Avoid copying kernel files; doing so may result in corrupting the running kernel, or any other configured kernel on the system.

---

6. Notify users that the system will be shut down. You can use the */usr/sbin/wall* command and/or the interactive capabilities of the */usr/sbin/shutdown* command to broadcast a message to users before the system goes down. For details, see *wall* (1M), *shutdown* (1M), and “Shutting Down Systems,” in Chapter 3, “Configuring a System,” in *Managing Systems and Workgroups: A Guide for HP-UX System Administrators*.

---

**NOTE** You only need to do the next steps if you are changing hardware, such as adding new peripherals. If you are only adding or removing a kernel module, or simply changing a static kernel parameter, reboot the system to active the new kernel with `shutdown -r`.

---

7. Bring the system to a halt using the `shutdown` command.
8. Turn off the power to all peripheral devices and *then* to the SPU.
9. Install the hardware or remove interface cards or peripheral devices. Refer to the documents shipped with the products being installed and to *Configuring HP-UX for Peripherals* for specific instructions.
10. Turn on the power to all peripheral devices. Wait for them to become “ready”, *then* turn on power to the SPU. The system will attempt to boot the new kernel.

## If the New Kernel Fails to Boot

If the new kernel fails to boot

1. Boot the system from the backup kernel (`/stand/vmunix.prev`) and repeat the process of creating a new kernel. See the section “Booting From an Alternate Kernel,” in Chapter 3, “Configuring a System,” in *Managing Systems and Workgroups: A Guide for HP-UX System Administrators* for information on rebooting from a backup kernel.
2. Remove the “bad” kernel. As soon as you boot from the backup kernel, remove the “bad” unbootable kernel using the `-d` option of the `kmupdate` command:

```
kmupdate -d /stand/vmunix
```

---

**NOTE** This step is necessary to avoid removing from the system the component files of the currently running backup kernel (`/stand/vmunix.prev`). With the `-d` option of the `kmupdate` command you have removed all the kernel component files associated with the un-bootable kernel `/stand/vmunix`. Now it is safe to repeat the process of configuring a new kernel and performing a kernel update without overwriting `/stand/vmunix.prev` with the “bad” kernel. Once `/stand/vmunix` has been removed, there is no longer a default kernel from which to boot, until a new kernel is configured and updated via the `kmupdate` command.

---

## Saving a Backup Kernel Permanently

System administrators and kernel developers often like to save a trusted good kernel (for example, the default kernel, `/stand/vmunix`, running at the end of a clean OS installation), before starting the process of building new kernels.

Although the `kmupdate (1M)` command always saves the previous `/stand/vmunix` kernel to the kernel name `/stand/vmunix.prev`, you must be aware that a subsequent use of `kmupdate` will again overwrite the `vmunix.prev` kernel. The recommended way to permanently save `/stand/vmunix` is to move (do **not** copy) `/stand/vmunix` to a safe name such as `/stand/vmunix.bkup`.

Furthermore, if `/stand/vmunix` had been the running kernel, we recommend an immediate reboot from the renamed location:

```
# mv /stand/vmunix /stand/vmunix.bkup
# shutdown -r
```

Once `/stand/vmunix` has been moved, there is no longer a default kernel from which to boot, until a new kernel is configured and updated with the `kmupdate` command. To boot from a non-default kernel, refer to “Booting from an Alternate Kernel” in the chapter “Administering a System: Booting and Shutdown,” in *Managing Systems and Workgroups: A Guide for HP-UX System Administrators*.

---

**CAUTION** Although the main kernel file (for example `/stand/vmunix`) carries the name of the kernel, you must remember that a kernel is no longer contained in that single file, but is composed of several essential files, and depends on a specific directory structure and symbolic links to be set properly for the kernel to be able to boot. The old unsupported practice of *copying* the `/stand/vmunix` file may result in the eventual loss of the kernel component set necessary to boot the copy.

---