GRUB 2 is the default boot loader for Ubuntu. GRUB 2's method of building the GRUB menu is vastly different from GRUB 0.97. This page describes the new file structure and details how to the user can change the default settings used to create the boot menu.

The current version of GRUB is 1.99, which was introduced with Ubuntu 11.04, Natty Narwhal and is the version installed with 12.04, Precise Pangolin LTS. The documentation on this page applies to GRUB 1.99 unless otherwise noted. To determine your version, use **grub-install -v.**

**Introduction**

GRUB 2 builds its menu (grub.cfg) by running scripts found in the */etc/grub.d/folder* and the settings in the */etc/default/grub* file. This file is recreated automatically whenever the update-grub command is run. This command activates scripts which by default determine the status of the current Ubuntu OS and search the system for other operating systems which, if found, are added to the GRUB 2 menu. The update-grub

* command can be invoked by a user with Administrator (root) privileges and is automatically run when GRUB 2 packages or the kernel is updated.

**File Structure**

GRUB 2 incorporates a totally revised directory and file hierarchy. The *menu.lst* of GRUB legacy is no longer used.

To find out where GRUB 2 is installed, the user can run the following commands:

* Device: **sudo grub-probe -t device /boot/grub**
* UUID: **sudo grub-probe -t fs\_uuid /boot/grub**

From the GRUB 2 menu at boot, the user can also determine which Ubuntu is controlling the boot on a multi-OS system. By default, the first menuentry always lists an option from the installation which is *in charge* of GRUB 2. For example, if the first menuentry contains "*on sda5*, then the GRUB installed on sda5 OS is controlling the menu/boot.

**Main Directories & Files**

**The Menu: /boot/grub/grub.cfg**

*grub.cfg* is the file which replaced GRUB 0.97's *menu.lst*. This file contains the GRUB 2 menu information but unlike GRUB's *menu.lst* the *grub.cfg* file is not normally edited directly. The file is the result of various scripts which build the menu in sections, with each script responsible for a specific part of the menu.

* *grub.cfg* is updated by running the update-grub command as root.
  + The update-grub command is a stub for
    - **grub-mkconfig -o /boot/grub/grub.cfg**.
  + If the user wishes to create a boot menu with a different name or path, use the full command with the *-o* switch.
* *grub.cfg* is automatically rebuilt with kernel installations/removals, updates to some GRUB 2 packages, when *update-grub* is executed, or when other operations would affect the boot process.
* The *grub.cfg* file is not created or updated when the grub-install command is run.

**File Layout**

* Each section is clearly delineated with *"(### BEGIN)"* and references the script in the */etc/grub.d* directory from which the information was generated. These sections are more fully explained in the [Scripts](https://help.ubuntu.com/community/Grub2/Setup#Scripts:_.2Fetc.2Fgrub.d.2F) section.

**Editing grub.cfg directly**

* In the earliest versions of GRUB 2, not only was editing this file discouraged, it was made *read-only* even for "root". This changed and now the developers acknowledge that sometimes the user may wish to simply edit the file manually. Because of the automatic changes induced by update-grub however, manual edits are subject to being overwritten by the system.

If *grub.cfg* is manually edited, here are some considerations:

* + The user will need to edit the file after each running of update-grub, updates of some GRUB 2 packages, and addition or removal of the main Ubuntu's system kernels.
  + It is possible to "freeze" the current *grub.cfg* file by redirecting the update-grub command so that it doesn't actually run. The versatility of GRUB 2 probably makes this option the least desirable.
  + The parts of *grub.cfg* which the user actually sees are generated by the */etc/grub.d* scripts. The user can make one or more of those scripts non-executable and substitute a custom entry or edit the actual script. Either method allows menu customization while still permitting the system to update GRUB 2.

**User Settings: /etc/default/grub**

The entries in this file can be edited by a user with administrator (root) privileges and are incorporated into *grub.cfg* when it is updated. This configuration file contains information formerly contained in the upper section of GRUB Legacy's *menu.lst* and items contained on the end of the kernel line.

While the most common settings are present in the default file, additional environmental settings are available for inclusion in this file. These include items such as backgrounds and themes. The pre-defined variables are contained in /usr/sbin/grub-mkconfig and can be listed using the following command:

* grep "export GRUB\_DEFAULT" -A50 /usr/sbin/grub-mkconfig | grep GRUB\_

Common settings are detailed in the [Configuring GRUB 2](https://help.ubuntu.com/community/Grub2/Setup#Configuring_GRUB_2) section.

**Scripts: /etc/grub.d/**

The scripts in this directory are read during execution of the update-grub command and their instructions are incorporated into */boot/grub/grub.cfg*.

* The placement of the menu items in the *grub.cfg* menu is determined by the order in which the files in this directory are run. Files with a leading numeral are executed first, beginning with the lowest number. *10\_linux* is run before *20\_memtest*, which would run before *40\_custom*. If files with alphabetic names exist, they are run after the numerically-named files.
* Custom menu entries can be added to the *40\_custom* file or in a newly created file. Based on its name, *40\_custom* entries by default appear at the bottom of the menu. A custom file beginning with *06\_* would appear at the top of the menu since its alphanumeric sorting would place it ahead of *10\_* through *40\_* files.
* Only executable files generate output to *grub.cfg* during execution of update-grub. By default the files in the */etc/grub.d* folder are executable.

**Script Descriptions**

* **00\_header** Sets environmental variables such system file locations, video settings, and previously saved entries. It also imports preferences stored in */etc/default/grub*. Users normally do not need to make changes to this file.
* **05\_debian\_theme** The settings in this file set the GRUB 2 background image, text colors, selection highlighting and themes. In the absence of a splash image, this file sets a monochromatic theme for the initial menu display. Information on how to set font renderings and splash images are discussed in the [Grub2/Displays](https://help.ubuntu.com/community/Grub2/Displays) community documentation page.
* **10\_linux** Identifies kernels on the root device for the operating system in use and creates menu entries for these items. This includes the associated *recovery mode* option if enabled. In GRUB 1.99 and later, only the latest kernel is displayed on the main menu page, with additional kernels included in a submenu. See the [Grub2/Submenus](https://help.ubuntu.com/community/Grub2/Submenus) page for information on this feature.

Note: For earlier versions of GRUB 2, all kernels residing in the boot folder are included on the main menu. To reduce the number of displayed kernels, remove the older kernels from the */boot* folder or use one of the available GRUB 2 customization applications.

* **20\_memtest86+** Searches for */boot/memtest86+.bin* and includes it as an option on the GRUB 2 boot menu. There is currently no line option to remove this entry from the menu. The display of *memtest86+* can be inhibited by removing the executable bit from this file and running update-grub.
  + sudo chmod -x /etc/grub.d/20\_memtest86+
  + sudo update-grub
* **30\_os-prober** This script uses *os-prober* to search for Linux and other operating systems and places the results in the GRUB 2 menu.
  + The file's sections include options for Windows, Linux, OSX, and Hurd.
  + Variables in this file determine the format of the displayed names in */boot/grub/grub.cfg* and on the GRUB 2 menu. Users familiar with basic scripting can alter these variables to change the format of the displayed menu entries.
  + The user can insert an entry into */etc/default/grub* which disables this script (see [Configuring GRUB 2](https://help.ubuntu.com/community/Grub2/Setup#Configuring_GRUB_2)). Removing the executable bit from the file will also prevent it from searching for other operating systems.
  + By default *os-prober* ignores any disks that possess "dmraid" signatures, which can be listed via **sudo dmraid -r -c** (if installed).
  + If the script finds another Ubuntu/Linux operating system, it will attempt to use the titles contained in the *10\_linux* section of that installation's *grub.cfg* file. If not found, the script will construct a menuentry from the actual boot information it locates.
* **40\_custom** A template for adding custom menu entries which will be inserted into *grub.cfg* upon execution of the update-grub command.
  + The contents of this file, below the "*exec tail -n +3 $0*" line and the default comments, are imported directly into */boot/grub/grub.cfg* without any changes.
    1. As the comments at the top of the file state, add custom lines below those already contained in the *40\_custom* file.
  + The file can be renamed or copied. The file must be executable to be incorporated into the GRUB 2 menu.
    1. As mentioned in the introduction of this section, the name of the file determines the location of its contents in the GRUB 2 menu. As a general rule, if using numerals at the start it is advisable to allow the *00\_header* and *05\_debian\_theme* scripts to be run before the first custom menu. These files contain no menuentries and thus will not interfere with a menuentry from a custom script with a lower priority.

**Configuring GRUB 2**

Configuration changes are normally made to */etc/default/grub* and to the custom files located in */etc/grub.d*. Any changes made directly to the */boot/grub/grub.cfg* are overwritten whenever update-grub is executed either by the user or when called automatically by various system functions.

After editing */etc/default/grub* or the scripts in the */etc/grub.d* folder the user should run **sudo update-grub** to incorporate the changes into the GRUB 2 menu.

Some of the most common changes, such as the default OS/kernel and menu timeout, can be changed from within a GUI applications such as Grub Customizer.

**/etc/default/grub**

This file contains basic settings which would be considered normal for the user to configure. Options include the time the menu is displayed, the default OS to boot, etc. The default options included in file upon installation are only a small number of the variables which GRUB 2 can recognize. The GRUB 2 defined variables can be identified using the command previously documented in the [GRUB 2 Settings](https://help.ubuntu.com/community/Grub2/Setup#User_Settings:_.2Fetc.2Fdefault.2Fgrub) section or at the end of this section.

In addition to the defined variables, the user may introduce self-created variable(s) and use them in the scripts contained in the */etc/grub.d* folder.

**Editing the File**

The *grub* file is a system file, therefore any editing must be done by a user with 'Administrator/root' privileges. The file is a simple text file and can be edited by any text editor. In Ubuntu the default text mode editor is **nano** and the graphical mode editor is **gedit**, and the file can be edited with one of the following commands. The "&" allows the terminal to be used to update GRUB 2 once the user saves the file.

* sudo nano /etc/default/grub

or in graphical mode

* gedit admin:///etc/default/grub &

After making changes and saving the file, the GRUB 2 menu must be updated to include the changes by running:

* sudo update-grub

**Specific Entries**

This section details some of the entries available in the */etc/default/grub* file. Much of this information can also be found by referring the the [Gnu GRUB manual](http://www.gnu.org/software/grub/manual) or on the system itself by running the following command:

* info -f grub -n 'Simple configuration'
* ***GRUB\_DEFAULT=***
  + Sets the default menu entry. Entries may be numeric, a complete menuentry quotation, or "saved"
    1. **GRUB\_DEFAULT=0** Sets the default menu entry by menu position. The first "menuentry" in *grub.cfg* is 0, the second is 1, etc.
       - Note: Grub 1.99 introduces a *submenu* menu structure. For a menu item in a submenu, the entry becomes a two-digit entry. The first entry is the position of the *submenu* title in the main menu. The second entry is the position within the *submenu*. If the *submenu* is the 3rd entry in the main entry, and the user wishes to boot the first entry in the *submenu*, it would be designated as "2>0". See the community documentation for a fuller explanation: [Grub2/Submenus](https://help.ubuntu.com/community/Grub2/Submenus).
    2. **GRUB\_DEFAULT="xxxx"** An exact menu entry, including the quotation symbols, may also be used. In this case, location in the menu will not matter.
       - Example: GRUB\_DEFAULT="Ubuntu, Linux 2.6.31-9-generic"
       - Example (Submenu Entry): GRUB\_DEFAULT="2>Ubuntu, Linux 2.6.38-8-generic"
    3. **GRUB\_DEFAULT=saved** The information in this section applies to GRUB 1.98 and later. The "saved" entry enables the "grub-reboot" and "grub-set-default" commands to set the default OS for future boots.
       - The default OS for future boots will *not* be set merely by selecting an OS when booting.
       - **grub-set-default** Sets the default boot entry until changed.
         * The format is **sudo grub-set-default X**, with *X* being the menu entry position (starting with 0 as the first entry) or the exact menu string.

Example: **sudo grub-set-default 3**

Example: sudo grub-set-default "Ubuntu, Linux 2.6.32-15-generic"

* + - * + To obtain the existing menu entry choice number (starting from 0) or the menu entry "string", run:

**grep menuentry /boot/grub/grub.cfg**

* + - * **grub-reboot** This command sets the default boot entry *for the next boot only*. The format of the command is the same as for grub-set-default (see above).
* ***GRUB\_SAVEDEFAULT=***
  + If set to ***true*** this setting will automatically set the last selected OS from the menu as the default OS on the next boot.
    1. No commands need be run to set the default OS.
    2. Any time a menu entry is manually selected from the GRUB 2 menu, it becomes the default OS.
    3. This option currently does not work if your /boot directory resides on an LVM partition or RAID.
    4. For this to work you must also set *GRUB\_DEFAULT=saved*.

Note: Ubuntu's version of Grub 1.99 introduced the submenu feature. Designating a default boot option located in the *Previous Linux versions* submenu requires special formatting. For information regarding the submenu feature, please refer to [Grub2/Submenus](https://help.ubuntu.com/community/Grub2/Submenus)

* ***GRUB\_HIDDEN\_TIMEOUT=0***
  + Wait this many seconds for the user to press a key. During this period no menu is shown unless the user presses a key. If no key is pressed, control is passed to GRUB \_TIMEOUT when the GRUB\_HIDDEN\_TIMEOUT expires. See the note regarding bugs to this feature at the end of this section.
    1. The developers envisioned using this setting with a GRUB\_TIMEOUT value of 0. This would give users a period of time (GRUB\_HIDDEN\_TIMEOUT) to display the menu by pressing a key, after which the system would boot without the menu being displayed (GRUB\_TIMEOUT=0).
  + **GRUB\_HIDDEN\_TIMEOUT=0**
    1. No menu is displayed. The system is immediately booted to the default OS.
    2. This is the default setting with only one identified operating system.
       - To display the menu under this condition, place a *#* symbol at the start of the line and ensure the GRUB\_TIMEOUT setting is a positive integer.
    3. If the value is set to *0*, a *keystatus* check is performed to determine if the *SHIFT* key is depressed. If GRUB 2 determines the *SHIFT* key is depressed during the boot process, the menu will be displayed. This gives the user a method of interrupting an automatic boot which would normally not display the menu.
  + **GRUB\_HIDDEN\_TIMEOUT=X**
    1. **X** is a positive integer (e.g. 1, 5, 10, etc)
    2. The boot process will pause and display a blank screen or the designated splash image for *X* seconds. At the end of the time period, the system will boot. No menu will be displayed.
    3. While GRUB\_HIDDEN\_TIMEOUT is active, the menu can be displayed by pressing any key.
  + **GRUB\_HIDDEN\_TIMEOUT=** 
    1. No value entered after the **=** sign
    2. The menu will be displayed for the number of seconds designated by GRUB\_TIMEOUT.
* ***GRUB\_HIDDEN\_TIMEOUT\_QUIET=true***
  + Determines whether a countdown timer is displayed on a blank screen when using the *GRUB\_HIDDEN\_TIMEOUT* feature.
    1. **true** No countdown is displayed. The screen will be blank.
    2. **false** A counter will display on a blank screen for the duration of the GRUB\_HIDDEN\_TIMEOUT value.

Note: There is a longstanding confirmed bug on the hidden menu feature in GRUB 1.97 to GRUB 1.99. The menu may not hide as specified in the description on this page. While editing the *30\_os-prober* script can fix this issue, it is beyond the scope of this page.

* ***GRUB\_TIMEOUT=10***
  + Sets the time period in seconds for the menu to be displayed before automatically booting unless the user intervenes.
    1. This instruction begins at the expiration of GRUB\_HIDDEN\_TIMEOUT.
    2. Setting this value to -1 will cause the menu to display until the user makes a selection.
    3. On a single OS system, by default this setting is not used and the menu will not display.
       - To display the menu on each boot:
         * Place a comment symbol (*#*) in front of the GRUB\_HIDDEN\_TIMEOUT entry
         * Make the GRUB\_TIMEOUT value 1 or higher.
* ***GRUB\_DISTRIBUTOR=lsb\_release -i -s 2> /dev/null || echo Debian***
  + Retrieves the descriptive name in the menu entry. (Ubuntu, Xubuntu, Debian, etc.)
* ***GRUB\_CMDLINE\_LINUX***
  + Entries on this line are added to the end of the 'linux' command line (GRUB legacy's "kernel" line) for both normal and recovery modes. It is used to pass options to the kernel.
* ***GRUB\_CMDLINE\_LINUX\_DEFAULT=***"quiet splash"
  + This line imports any entries to the end of the 'linux' line (GRUB legacy's "kernel" line). The entries are appended to the end of the normal mode only.
    1. To view a black screen with boot processes displayed in text, remove "quiet splash". To see the grub splash image plus a condensed text output, use "splash".
* ***#GRUB\_TERMINAL=console***
  + Uncomment to disable graphical terminal. This may provide help if the GRUB 2 menu is too large or unreadable. It also may help when using the GRUB\_HIDDEN\_TIMEOUT feature.
* ***#GRUB\_DISABLE\_LINUX\_UUID="true"***
  + Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
    1. The *search* line will still use UUIDs. The *linux* line will use the */dev/sdXY* convention when this option is enabled.
* ***#GRUB\_GFXMODE=640x480***
  + GRUB 2 will automatically set the menu resolution to what it thinks is the best option. Uncommenting this line will set the resolution to 640x480, or you may change the value to another GRUB-compatible setting.
    1. The setting applies only to the boot menu display, not the resolution of the operating system that boots.
       - Tip: Setting the same resolution in GRUB 2 and the operating system will decrease boot times slightly.
    2. Although not required, the user can also specify the color bit depth by appending it to the resolution setting. An example would be 1280x1024x24 or 640x480x32.
    3. The user can also add multiple resolutions. If GRUB 2 cannot use the first entry, it will try the next setting. Settings are separated by a comma. Example: *1280x1024x16,800x600x24,640x480*.
    4. If using a splash image, the resolution setting and the splash image size should be compatible for best results.
    5. If using an entry that produces a "not found" message when running *update-grub*, try adding or changing the color bitdepth.
    6. Resolutions available to GRUB 2 can be displayed by typing **videoinfo** in the GRUB 2 command line. The command line is accessed by typing "c" when the main GRUB 2 menu screen is displayed.
    7. If this line is commented (#) or the resolution is unavailable GRUB 2 uses the default setting determined by */etc/grub.d/00\_header*.
       - In GRUB 1.99 (Natty) the 'optimum' resolution is selected by GRUB if no resolution is specified.
* ***#GRUB\_DISABLE\_LINUX\_RECOVERY=true***
  + Allows the user to prevent the recovery options from inclusion on the GRUB 2 menu.
    1. Uncomment (remove the # symbol) to prevent the "Recovery" mode kernel options from appearing in the menu. If you want a "Recovery" option for only one kernel, make a special entry in /etc/grub/40\_custom.
* ***GRUB\_INIT\_TUNE="480 440 1"***
  + Removing the comment symbol (*#*) allows GRUB 2 to play a single beep just prior to the Grub 2 menu display. More complex tunes can be designed by expanding the pitch/duration values.
    1. The format is *tempo [pitch1 duration1] [pitch2 duration2] ...* 
       - *tempo* is set once and applies to all *duration* settings.
       - *duration* is the result of *60/tempo*. A *duration* of 1 with a *tempo* of 60 would produce a 1 second beep. A *duration* of 1 with a *tempo* of 480 produces a .125 second beep.
    2. Caution: The menu is displayed after the tune is complete. Creating a long tune will delay the appearance of the menu.
    3. Online documentation is available by typing **info grub --index-search play** in a terminal. It mentions the ability to play a tune with the command play file if the tune is composed with little-endian numbers.
* ***GRUB\_BACKGROUND***
  + Sets the background image, enter the full path to the image here. See splash image configuration above for further details and other options.
* ***GRUB\_DISABLE\_OS\_PROBER=true***
  + This entry is used to prevent GRUB from adding the results of os-prober to the menu. A value of "true" disables the os-prober check of other partitions for operating systems, including Windows, Linux, OSX and Hurd, during execution of the update-grub command. Using this option rather than removing the executable bit from the */etc/grub.d/30\_os-prober* file has several advantages:
    1. The setting can be easily changed while making other changes to the *grub* file.
    2. While both methods prevent os-prober from running and placing items in the menu display, using this setting allows the *30\_os-prober* script, but not the *os-prober* command, to run. This script enables the GRUB\_HIDDEN\_TIMEOUT setting and/or the ability to display a hidden menu by pressing the ESC key (depending on other settings). This functionality is lost if the *30\_os-prober* script is disabled by making it unexecutable.

To inspect the currenlty-available predesignated variables on your system's GRUB 2 library files, run:

* + **grep DEVICE -A40 /usr/sbin/grub-mkconfig**

**User-Assigned Variables**

* Variables in the */etc/default/grub* file are read and acted upon as needed. This allows users to add their own variables in a central location and then alter the GRUB 2 scripts to meet their needs. Keeping the user-defined variables within this file allows easier management of, and change to, those variables when desired. To add a variable, use the following format:
  + export <VARIABLE>
* The following example would define a variable used to omit a specific Ubuntu partition from the menu:
  + Define and export the variable in */etc/default/grub* :
    - *export GRUB\_EXCLUDE\_PARTITION="/dev/sda1"*
  + Establish the conditional in */etc/grub.d/30\_os-prober* in the *10\_linux* section:
    - *if [ $GRUB\_EXCLUDE\_PARTITION = $LROOT ]; then*
    - or
    - *if [ ! $GRUB\_EXCLUDE\_PARTITION = $LROOT ]; then*
    - Complete the conditional at the appropriate point in the script
  + Save the files and run sudo update-grub to incorporate the changes into the GRUB 2 menu.

**Links**

* [Grub2](https://help.ubuntu.com/community/Grub2)
* [Grub2/CustomMenus](https://help.ubuntu.com/community/Grub2/CustomMenus)
* [Grub2/Displays](https://help.ubuntu.com/community/Grub2/Displays)
* [Grub2/Installing](https://help.ubuntu.com/community/Grub2/Installing)
* [Grub2/Passwords](https://help.ubuntu.com/community/Grub2/Passwords)
* [Grub2/Submenus](https://help.ubuntu.com/community/Grub2/Submenus)
* [Grub2/Troubleshooting](https://help.ubuntu.com/community/Grub2/Troubleshooting)
* [Grub2/Upgrading](https://help.ubuntu.com/community/Grub2/Upgrading)
* [Grub2: chainload and multiboot](https://help.ubuntu.com/community/Grub2%20Other%20Os#Chain_Loading)
* [Grub 2](https://wiki.ubuntu.com/Grub2) (Ubuntu Wiki)

**External Links**

* [GNU GRUB2 Manual](http://www.gnu.org/software/grub/manual)
* [Grub Customizer (Launchpad)](https://launchpad.net/grub-customizer)