CHAPTER 4

The Menu System

Drupal’s menu system is complex but powerful. The term “menu system” is somewhat of a misnomer. It may be better to think of the menu system as having three primary responsibilities: callback mapping, access control, and menu customization. Essential code for the menu system is in includes/menu.inc, while optional code that enables such features as customizing menus is in modules/menu.

In this chapter, we’ll explore what callback mapping is and how it works, see how to protect menu items with access control, learn to use menu wildcards, and inventory the various built-in types of menu items. The chapter finishes up by examining how to override, add, and delete existing menu items, so you can customize Drupal as non-intrusively as possible.

Callback Mapping

When a web browser makes a request to Drupal, it gives Drupal a URL. From this information, Drupal must figure out what code to run and how to handle the request. This is commonly known as routing or dispatching. Drupal trims off the base part of the URL and uses the latter part, called the path. For example, if the URL is http://example.com/?q=node/3, the Drupal path is node/3. If you are using Drupal’s clean URLs feature, the URL in your browser would be http://example.com/node/3, but your web server is quietly rewriting the URL to be http://example.com/?q=node/3 before Drupal sees it; so Drupal always deals with the same Drupal path. In the preceding example, the Drupal path is node/3 regardless of whether clean URLs are enabled. See “The Web Server’s Role” in Chapter 1 for more detail on how this works.

Mapping URLs to Functions

The general approach taken is as follows: Drupal asks all enabled modules to provide an array of menu items. Each menu item consists of an array keyed by a path and containing some information about that path. One of the pieces of information a module must provide is a page callback. A callback in this context is simply the name of a PHP function that will be run when the browser requests a certain path. Drupal goes through the following steps when a request comes in:
1. Establish the Drupal path. If the path is an alias to a real path, Drupal finds the real path and uses it instead. For example, if an administrator has aliased http://example.com/?q=about to http://example.com/?q=node/3 (using the path module, for example), Drupal uses node/3 as the path.

2. Drupal keeps track of which paths map to which callbacks in the menu_router database table and keeps track of menu items that are links in the menu_links table. A check is made to see if the menu_router and menu_links tables need rebuilding, a rare occurrence that happens after Drupal installation or updating.

3. Figure out which entry in the menu_router table corresponds with the Drupal path and build a router item describing the callback to be called.

4. Load any objects necessary to pass to the callback.

5. Check whether the user is permitted to access the callback. If not, an “Access denied” message is returned.

6. Localize the menu item’s title and description for the current language.

7. Load any necessary include files.

8. Call the callback and return the result, which index.php then passes through theme_page(), resulting in a finished web page.

A visual representation of this process is shown in Figures 4-1 and 4-2.