

**EnableItem**

Undim a menu or a menu item; make selectable

#include &lt;Menus.h&gt;

**Menu Manager**

```
void      EnableItem(theMenu, whichItem );
MenuHandle theMenu ;      handle of menu to affect
short     whichItem ;      item ID of item to enable; 0=entire menu
```

**EnableItem** undims a menu title or an individual item in a menu. Use this after dimming an item or a menu title via **DisableItem**.

*theMenu* is a handle leading to a variable-length MenuInfo structure. It is a value obtained via **NewMenu** or **GetMenu**. It identifies the menu containing the item to enable, or if *whichItem* = 0, the menu to enable.

*whichItem* identifies which item to enable or specifies to enable an entire menu. When *whichItem*=0, the menu title is undimmed and all items in that menu are enabled. If it is a valid item ID, it specifies an item in *theMenu*. Items are numbered sequentially with the topmost item having an ID of 1; the ID of the last item can be obtained via **CountMItems**

**Returns:** none

---

Notes: After using **DisableItem** on a menu item (or including the "⌘" metacharacter in its item text), use **EnableItem** to make the item selectable and to include it in the search performed by **MenuKey**.

When you disable or re-enable an entire menu (*whichItem* =0), you should call **DrawMenuBar** to display the changed menu title.

This works by setting bit number *whichItem* in the enableFlags field of the relevant MenuInfo structure. For instance, the following calls are functionally equivalent:

```
EnableItem( myMenu, 2 );
(*myMenu)->enableFlags |= 4;      /* set bit 2 */
```

All menus and items are enabled when inserted into the menu unless explicitly disabled by including the "⌘" metacharacter in the item text. Items with IDs higher than 31 are always enabled.