Q: How do I disable/enable menu cells?

A: The method setUpdateAction:(SEL)aSelector forMenu:aMenu is responsible for the menu cell update, where aSelector is your own update method, and aMenu the menu you want to update. This method has to be called inside appDidInit: because NXApp is responsible for the update.   Doing so makes your program more efficient, since it needs to be done only once.

Q: Should I use the application method setAutoupdate:(BOOL)flag or the menu update method for updating the menu items? What are the trade-offs?

A: The application method setAutoupdate:YES causes the menus to stay up to date all the time (i.e., it only updates them when they are on-screen).   Calling update directly every time something changes might be a lot of extra work for nothing. However,   if things rarely change in the menu, it might be worth not incurring the window list traversal setAutoupdate: causes.

Example (adapted from /NextDeveloper/Examples/Draw)

```
static void initMenu(id menu)
/* A private C function:
 * Sets the updateAction for every menu item which sends to the
 * First Responder (i.e. their target is nil).
 * Returns the active menu if is found in this menu.
 */
{
    int count;
    id matrix, cell;
    id matrixTarget, cellTarget;

    matrix = [menu itemList];
    matrixTarget = [matrix target];

    count = [matrix cellCount];
    while (count--) {
            cell = [matrix cellAt:count :0];
            cellTarget = [cell target];
            if (!matrixTarget && !cellTarget) {
```

```
            [cell setUpdateAction:@selector(menuItemUpdate:) forMenu:menu];
        } else if ([cell hasSubmenu]) {
            initMenu(cellTarget);
        }
    }
}


/* Initializes the updateAction method for the menu cells.
  setAutoupdate:YES means that updateWindows will be called
  after every event is processed (to keep the menu items up to date).
*/
- appDidInit:sender
{
  ........
  initMenu([NXApp mainMenu]);
  [NXApp setAutoupdate:YES];
  return self;
}
```

```
#define                 TOGGLE  4                        /* Some integer cell tag value      */

- (BOOL)menuItemUpdate:menuCell
/* Here goes your own code for menu cell update. This is only a simple example
   which toggles the state of the selected cell. The return value YES means
   that the menu will be updated.
*/
{

  BOOL              cellState;


          switch ([menuCell tag])
          {
                  case TOGGLE: /* toggle menu cell state          */
                          cellState = [menuCell isEnabled];
                          [menuCell setEnabled:!cellState];
                          return YES;
                  default:
                          break;
```

```
        }

        return NO;
}
```

QA636

Valid for 2.0, 3.0