

Using the Windows NT Wait Cursor

Creation Date: Feb 26, 1998

Keywords: Windows NT, Wait Cursor, Cursor

Applications built with OpenStep 4.2 Enterprise will not automatically call the Windows NT wait cursor during long operations such as launching applications and loading frameworks. If you know your application is about to begin a long operation, however, you can bring up the NT wait cursor manually. This document contains sample code for calling the NT wait cursor.

```
#ifdef WIN32
```

```
#import <Windows.h>
// Apple OpenStep Enterprise NSCursor private initializer method
@interface NSCursor (Secret)
- (id) initWithCursorHandle: (HCURSOR) handle;
@end
```

```
#endif
```

```
@implementation NSCursor (waitCursor)
```

```
+ (NSCursor *) waitCursor
```

```
{
    static NSCursor *waitCursor = nil;
```

```
#ifdef WIN32
    if (!waitCursor) {
        HCURSOR cursor = LoadCursor(NULL, IDC_WAIT);

        if (cursor && [NSCursor instancesRespondToSelector: @selector(initWithCursorHandle:)])
            waitCursor = [[NSCursor alloc] initWithCursorHandle: cursor];
    }
#endif
```

```
    return waitCursor;
}
```

@end

// some class implementation wanting to use a wait cursor

- (void) deactivateWaitCursor

```
{
#ifdef    WIN32
    NSCursor * const waitCursor = [NSCursor waitCursor];
    NSArray * const windows = [NSApp windows];
    const int count = [windows count];
    int i;

    [waitCursor setOnMouseEntered: NO];
    for (i = 0; i < count; i++)
        [[windows objectAtIndex: i] resetCursorRects];
#endif
    return;
}
```

- (void) activateWaitCursor

```
{
#ifdef    WIN32
    NSCursor * const waitCursor = [NSCursor waitCursor];
    NSArray * const windows = [NSApp windows];
    const int count = [windows count];
    int i;

    [waitCursor setOnMouseEntered: YES];
    for (i = 0; i < count; i++) {
        NSWindow * const window = [windows objectAtIndex: i];
        NSView * const contentView = [window contentView];
    }
}
```

```
        [window enableCursorRects];  
        [contentView addCursorRect: [contentView frame] cursor: waitCursor];  
    }  
#endif  
    return;  
}
```