HIToolbar Reference

(Legacy)

Developer

Contents

Tables

HIToolbar Reference (Legacy)

Framework	Carbon/Carbon.h
Declared in	HIToolbar.h

Important: This document may not represent best practices for current development. Links to downloads and other resources may no longer be valid.

Overview

HIToolbar is the Carbon equivalent of the Cocoa toolbar (specifically, the `NSToolbar` and `NSToolbarItem` classes). The toolbar object and the items associated with a toolbar are both subclassed from HIObject. Toolbar items can have HIViews associated with them.

For more information, see Using HIToolbar.

Functions by Task

Creating Toolbars

`HIToolbarCreate` (page 11)
Creates a toolbar.

Manipulating Toolbars

`HIToolbarChangeAttributes` (page 9)
Changes the attributes of a toolbar.

`HIToolbarCopyIdentifier` (page 9)
Obtains the identifier for a toolbar.

`HIToolbarCopyItems` (page 10)
: Obtains the array of toolbar items for a toolbar.

`HIToolbarGetAttributes` (page 12)
: Obtains the attributes for the given toolbar.

`HIToolbarGetDelegate` (page 13)
: Returns the current delegate in use by a toolbar.

`HIToolbarSetDelegate` (page 33)
: Sets the delegate object for a toolbar.

`HIToolbarGetDisplayMode` (page 14)
: Obtains the current display mode of a toolbar.

`HIToolbarSetDisplayMode` (page 33)
: Sets the current display mode of a toolbar.

`HIToolbarGetDisplaySize` (page 14)
: Obtains the current display size of a toolbar.

`HIToolbarSetDisplaySize` (page 34)
: Sets the current display size of a toolbar.

Creating and Adding Toolbar Items

`HIToolbarCreateItemWithIdentifier` (page 11)
: Creates a toolbar item.

`HIToolbarAppendItem` (page 8)
: Appends an item to the toolbar.

`HIToolbarInsertItemAtIndex` (page 16)
: Inserts a toolbar item at a given index into a toolbar.

`HIToolbarRemoveItemAtIndex` (page 32)
: Removes an item at a given index from a toolbar.

`HIToolbarItemCreate` (page 23)
: Creates a toolbar item.

`HIToolbarSetItemsWithIdentifiers` (page 35)
: Sets a toolbar's items all at once.

Manipulating Toolbar Items

`HIToolbarGetSelectedItemInWindow` (page 15)
: Obtains the toolbar item that is selected in a window.

`HIToolbarItemGetToolbar` (page 26)
: Obtains the toolbar associated with a toolbar item.

`HIToolbarItemGetAttributes` (page 23)
: Obtains the attributes of a toolbar item.

`HIToolbarItemGetAttributesInWindow` (page 24)
: Obtains the attributes of a toolbar item in the specified window.

`HIToolbarItemChangeAttributes` (page 16)
: Changes the attributes of a toolbar item.

`HIToolbarItemChangeAttributesInWindow` (page 17)
: Changes the attributes of a toolbar item in a specific window.

`HIToolbarItemCopyHelpText` (page 19)
: Obtains the help tag text for a toolbar item.

`HIToolbarItemSetHelpText` (page 28)
: Sets the help tag text for a toolbar item.

`HIToolbarItemCopyIdentifier` (page 20)
: Obtains the identifier for a given toolbar item.

`HIToolbarItemCopyImage` (page 21)
: Obtains the image for a toolbar item.

`HIToolbarItemSetImage` (page 30)
: Sets the image for a toolbar item.

`HIToolbarItemCopyLabel` (page 21)
: Obtains the label for a toolbar item.

`HIToolbarItemSetLabel` (page 31)
: Sets the label of a toolbar item.

`HIToolbarItemCopyMenu` (page 22)
: Obtains the submenu for a toolbar item.

`HIToolbarItemSetMenu` (page 31)
: Sets the submenu for a toolbar item.

`HIToolbarItemGetCommandID` (page 25)
: Gets the command ID of a toolbar item.

`HIToolbarItemSetCommandID` (page 27)
: Sets the command ID of a toolbar item.

`HIToolbarItemIsEnabled` (page 26)
: Determines if a toolbar item is enabled.

`HIToolbarItemSetEnabled` (page 28)
: Enables or disables a toolbar item.

`HIToolbarItemSetIconRef` (page 29)
: Sets the icon for a toolbar item.

`HIToolbarItemConfigDataChanged` (page 19)
: Tells the toolbar that the configuration for a toolbar item has changed.

Functions

HIToolbarAppendItem

Appends an item to the toolbar.

```
OSStatus HIToolbarAppendItem (
   HIToolbarRef inToolbar,
   HIToolbarItemRef inItem
);
```

Parameters

`inToolbar`
: The toolbar to receive the new item.

`inItem`
: The item reference of the toolbar item you are adding.

Return Value

An operating system result code.

Discussion

This function appends an item to the end of a toolbar. Generally, you should always add items via identifier, and not with this routine.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarChangeAttributes

Changes the attributes of a toolbar.

```
OSStatus HIToolbarChangeAttributes (
   HIToolbarRef inToolbar,
   OptionBits inAttrsToSet,
   OptionBits inAttrsToClear
);
```

Parameters

`inToolbar`

The toolbar whose attributes you want to change.

`inAttrsToSet`

The attributes you want to set. For possible values, see "Toolbar Attributes" (page 36).

`inAttrsToClear`

The attributes you want to clear. For possible values, see "Toolbar Attributes" (page 36).

Return Value
An operating system result code.

Availability
Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarCopyIdentifier

Obtains the identifier for a toolbar.

```
OSStatus HIToolbarCopyIdentifier (
   HIToolbarRef inToolbar,
   CFStringRef *outIdentifier
);
```

Parameters

`inToolbar`

The toolbar whose identifier you want to obtain.

`outIdentifier`

The identifier. You must release it when you are finished with it.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarCopyItems

Obtains the array of toolbar items for a toolbar.

```
OSStatus HIToolbarCopyItems (
   HIToolbarRef inToolbar,
   CFArrayRef *outItems
);
```

Parameters

`inToolbar`

The toolbar whose items you want to receive.

`outItems`

The array of toolbar items owned by the toolbar. You must release the array when you are finished with it.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarCreate

Creates a toolbar.

```
OSStatus HIToolbarCreate (
   CFStringRef inIdentifier,
   OptionBits inAttributes,
   HIToolbarRef *outToolbar
);
```

Parameters

`inIdentifier`

The identifier of the toolbar. If you specify that the toolbar auto-saves its configuration, this identifier is used to mark the config info in your application's preferences. You must specify an identifier.

`inAttributes`

Any attributes you want to set. For possible values, see "Toolbar Attributes" (page 36).

`outToolbar`

The toolbar reference to your new toolbar.

Return Value
An operating system result code.

Availability
Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarCreateItemWithIdentifier

Creates a toolbar item.

```
OSStatus HIToolbarCreateItemWithIdentifier (
   HIToolbarRef inToolbar,
   CFStringRef inIdentifier,
   CFTypeRef inConfigData,
   HIToolbarItemRef *outItem
);
```

Parameters

`inToolbar`
: The toolbar you are adding to.

`inIdentifier`
: The identifier of the item you want to add.

`inConfigData`
: Any config data required by the item to safely construct. Standard items do not require any extra data, so `NULL` can be passed.

`outItem`
: The newly created toolbar item.

Return Value

An operating system result code.

Discussion

This function creates an item specified by a particular identifier. Using this function allows you to create any item a delegate supports by naming its identifier. It also allows you to create standard items supplied by the Toolbox, such as the separator item.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarGetAttributes

Obtains the attributes for the given toolbar.

```
OSStatus HIToolbarGetAttributes (
   HIToolbarRef inToolbar,
   OptionBits *outAttributes
);
```

Parameters

`inToolbar`

The toolbar whose attributes you desire.

`outAttributes`

The toolbar's attributes. For details, see "Toolbar Attributes" (page 36).

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarGetDelegate

Returns the current delegate in use by a toolbar.

```
HIObjectRef HIToolbarGetDelegate (
   HIToolbarRef inToolbar
);
```

Parameters

`inToolbar`

The toolbar you want the delegate from.

Return Value

An HIObjectRef.

Discussion

The delegate handles the event processing for the toolbar.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarGetDisplayMode

Obtains the current display mode of a toolbar.

```
OSStatus HIToolbarGetDisplayMode (
   HIToolbarRef inToolbar,
   HIToolbarDisplayMode *outDisplayMode
);
```

Parameters

`inToolbar`

The toolbar whose display mode you want to receive.

`outDisplayMode`

The display mode.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarGetDisplaySize

Obtains the current display size of a toolbar.

```
OSStatus HIToolbarGetDisplaySize (
   HIToolbarRef inToolbar,
   HIToolbarDisplaySize *outSize
);
```

Parameters

`inToolbar`

The toolbar whose display size you want to get.

`outSize`

The display size.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarGetSelectedItemInWindow

Obtains the toolbar item that is selected in a window.

```
OSStatus HIToolbarGetSelectedItemInWindow (
   HIToolbarRef inToolbar,
   WindowRef inWindow,
   HIToolbarItemRef *outItem
);
```

Parameters

`inToolbar`
: The toolbar in question.

`inWindow`
: A window containing the toolbar.

`outItem`
: On return, the toolbar item that is selected in the specified window, or `NULL` if no item is selected.

Return Value

An operating system result code.

Discussion

Each window that shares a toolbar may have a different selected item. The `HIToolbarGetSelectedItemInWindow` function returns the selected item in a particular window.

Availability

Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarInsertItemAtIndex

Inserts a toolbar item at a given index into a toolbar.

```
OSStatus HIToolbarInsertItemAtIndex (
   HIToolbarRef inToolbar,
   HIToolbarItemRef inItem,
   CFIndex inIndex
);
```

Parameters

`inToolbar`

The toolbar to receive the new item.

`inItem`

The item reference of the toolbar item you are adding.

`inIndex`

The index at which you want to add the item. This index is zero-based.

Return Value

An operating system result code.

Discussion

Generally, you should always add items via identifier, and not with this routine.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemChangeAttributes

Changes the attributes of a toolbar item.

```
OSStatus HIToolbarItemChangeAttributes (
   HIToolbarItemRef inItem,
   OptionBits inAttrsToSet,
   OptionBits inAttrsToClear
);
```

Parameters

`inItem`

The item in question.

`inAttrsToSet`

The attributes to set on the item. For possible attributes, see "Toolbar Item Attributes" (page 44). Use `kHIToolbarItemNoAttributes` if you are clearing attributes.

`inAttrsToClear`

The attributes to clear on the item. This value can be `kHIToolbarItemNoAttributes` if you are setting attributes.

Return Value

An operating system result code.

Discussion

Only those attributes defined by the `kHIToolbarItemMutableAttrs` constant can be passed into this function.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemChangeAttributesInWindow

Changes the attributes of a toolbar item in a specific window.

```
OSStatus HIToolbarItemChangeAttributesInWindow (
   HIToolbarItemRef inItem,
   WindowRef inWindow,
   OptionBits inAttrsToSet,
   OptionBits inAttrsToClear,
   OptionBits inAttrsToNoLongerOverride
);
```

Parameters

`inItem`

The item in question.

`inWindow`
: The window containing the item in question.

`inAttrsToSet`
: The attributes to set on the item. For possible attributes, see "Toolbar Item Attributes" (page 44). Use `kHIToolbarItemNoAttributes` if you are clearing attributes and have no attributes to set.

`inAttrsToClear`
: The attributes to clear on the item. This value can be `kHIToolbarItemNoAttributes` if you are setting attributes and have no attributes to clear.

`inAttrsToNoLongerOverride`
: The attributes that are to no longer be overridden. Calling this function causes the attributes to be removed from the override mask for the toolbar item in the specified window. Their effective values revert to their non-window-specific attribute values.

Return Value
An operating system result code.

Discussion
This function allows the attributes of a toolbar item in the specified window to be overridden. The attributes used to draw the view of a toolbar item in a particular window are determined by combining the non-window-specific attributes for the item set by `HIToolbarItemChangeAttributes` (page 16) with the window-specific attributes set by this function. As a result, your application can have a toolbar that is shared across several windows with a toolbar item that is enabled in one window and disabled in another window.

When `HIToolbarItemChangeAttributesInWindow` is called to set or clear attributes, the toolbar item adds the changed attributes to a bitmask of attributes, thereby recording which attributes are overridden for a particular window. Once an attribute is overridden for a window (regardless of whether the attribute is set or cleared), the attribute remains overridden for that window until `HIToolbarItemChangeAttributesInWindow` is called with that attribute specified in the `inAttrsToNoLongerOverride` parameter.

Only those attributes defined by the `kHIToolbarItemMutableAttrs` constant can be passed into this function. For details, see "Toolbar Item Attributes" (page 44).

Availability
Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarItemConfigDataChanged

Tells the toolbar that the configuration for a toolbar item has changed.

```
OSStatus HIToolbarItemConfigDataChanged (
   HIToolbarItemRef inItem
);
```

Parameters

`inItem`

The item whose configuration changed.

Return Value

An operating system result code.

Discussion

This function tells the toolbar that the config data for a toolbar item has changed and should be written to the toolbar configuration preferences. This function is used when a custom toolbar item has extra configuration data that has changed (for example, you've changed an alias that a toolbar item points to). This function does nothing if the toolbar is not set to auto-save its configuration.

Availability

Available in Mac OS X v10.3 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemCopyHelpText

Obtains the help tag text for a toolbar item.

```
OSStatus HIToolbarItemCopyHelpText (
   HIToolbarItemRef inItem,
   CFStringRef *outShortText,
   CFStringRef *outLongText
);
```

Parameters

`inItem`

The item in question.

`outShortText`
: The short help text. This is what is displayed normally by the help tag system when the user hovers over the toolbar item with the mouse. You should release this string when you are finished with it. If you do not want to receive the short help text, pass `NULL` for this parameter.

`outLongText`
: The long help text. This is what is displayed by the help tag system when the user hovers over the toolbar item with the mouse and holds the command key down. You should release this string when you are finished with it. If you do not want to receive the long help text, pass `NULL` for this parameter.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemCopyIdentifier

Obtains the identifier for a given toolbar item.

```
OSStatus HIToolbarItemCopyIdentifier (
   HIToolbarItemRef inItem,
   CFStringRef *outIdentifier
);
```

Parameters

`inItem`
: The item in question.

`outIdentifier`
: The identifier of the item. You should release this string when you are finished with it.

Return Value

An operating system result code.

Discussion

The toolbar uses this identifier when writing the config information to the preferences (if set up for auto-config).

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemCopyImage

Obtains the image for a toolbar item.

```
OSStatus HIToolbarItemCopyImage (
   HIToolbarItemRef inItem,
   CGImageRef *outImage
);
```

Parameters

`inItem`

The item in question.

`outImage`

The retained image. You should release it when finished with it.

Return Value

An operating system result code.

Discussion

This image is already retained by the time you receive it, so you can release it when you are done with it.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemCopyLabel

Obtains the label for a toolbar item.

```
OSStatus HIToolbarItemCopyLabel (
   HIToolbarItemRef inItem,
   CFStringRef *outLabel
);
```

Parameters

`inItem`

The item in question.

`outLabel`

The label of the item. You should release this when you are finished with it.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemCopyMenu

Obtains the submenu for a toolbar item.

```
OSStatus HIToolbarItemCopyMenu (
   HIToolbarItemRef inItem,
   MenuRef *outMenu
);
```

Parameters

`inItem`

The item in question.

`outMenu`

The retained menu. You should release it when you are finished with it.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarItemCreate

Creates a toolbar item.

```
OSStatus HIToolbarItemCreate (
   CFStringRef inIdentifier,
   OptionBits inOptions,
   HIToolbarItemRef *outItem
);
```

Parameters

`inIdentifier`
: The identifier of the item in question.

`inOptions`
: Any options for the item.

`outItem`
: The item you created.

Return Value
An operating system result code.

Discussion
This function creates a toolbar item for use in a toolbar. Typically, you call `HIToolbarItemCreate` from inside your delegate when your delegate is asked to create a toolbar item.

Availability
Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarItemGetAttributes

Obtains the attributes of a toolbar item.

```
OSStatus HIToolbarItemGetAttributes (
   HIToolbarItemRef inItem,
   OptionBits *outAttributes
);
```

Parameters

`inItem`

The item in question.

`outAttributes`

The attributes of the item. For details, see "Toolbar Item Attributes" (page 44).

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemGetAttributesInWindow

Obtains the attributes of a toolbar item in the specified window.

```
OSStatus HIToolbarItemGetAttributesInWindow (
   HIToolbarItemRef inItem,
   WindowRef inWindow,
   OptionBits *outOverriddenAttributes,
   OptionBits *outCombinedAttributes
);
```

Parameters

`inItem`

The item in question.

`inWindow`

The window containing `inItem`. Passing `NULL` is the same as calling `HIToolbarItemGetAttributes` (page 23), which obtains the non-window-specific attributes for the item.

`outOverriddenAttributes`
: On return, a bitmask indicating the attributes for the item that are overridden in the specified window. If you don't need this information, pass `NULL`. For details on this bitmask, see "Toolbar Item Attributes" (page 44).

`outCombinedAttributes`
: On return, a bitmask indicating the effective attributes for the item in the specified window. The bitmask is produced by combining the non-window-specific attributes for the item with the overridden attributes for this window. If you don't need this information, pass `NULL`. For details on this bitmask, see "Toolbar Item Attributes" (page 44).

Return Value
An operating system result code.

Discussion
In addition to obtaining the attributes of the given item in a given window, `HIToolbarItemGetAttributesInWindow` obtains information about which attributes are overridden for that window.

Availability
Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarItemGetCommandID

Gets the command ID of a toolbar item.

```
OSStatus HIToolbarItemGetCommandID (
   HIToolbarItemRef inItem,
   MenuCommand *outCommandID
);
```

Parameters
`inItem`
: The item whose command ID is to be obtained.

`outCommandID`
: On return, the item's command ID.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemGetToolbar

Obtains the toolbar associated with a toolbar item.

```
HIToolbarRef HIToolbarItemGetToolbar (
   HIToolbarItemRef inItem
);
```

Parameters

`inItem`

The item in question.

Return Value

The toolbar item reference of the toolbar this item is bound to, or `NULL` if this toolbar item is not attached to any toolbar.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemIsEnabled

Determines if a toolbar item is enabled.

```
Boolean HIToolbarItemIsEnabled (
   HIToolbarItemRef inItem
);
```

Parameters

`inItem`

The item in question.

Return Value

A Boolean result indicating whether the item is enabled.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemSetCommandID

Sets the command ID of a toolbar item.

```
OSStatus HIToolbarItemSetCommandID (
   HIToolbarItemRef inItem,
   MenuCommand inCommandID
);
```

Parameters

`inItem`

The item whose command ID is to be set.

`inCommandID`

The command ID to set.

Return Value

An operating system result code.

Discussion

When an toolbar item is clicked, the default behavior is to send out the command assigned to the item. This function lets you to set that command ID. The command is sent out via the `ProcessHICommand` API.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarItemSetEnabled

Enables or disables a toolbar item.

```
OSStatus HIToolbarItemSetEnabled (
   HIToolbarItemRef inItem,
   Boolean inEnabled
);
```

Parameters

`inItem`
: The item in question.

`inEnabled`
: The new enabled state.

Return Value
An operating system result code.

Availability
Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarItemSetHelpText

Sets the help tag text for a toolbar item.

```
OSStatus HIToolbarItemSetHelpText (
   HIToolbarItemRef inItem,
   CFStringRef inShortText,
   CFStringRef inLongText
);
```

Parameters

`inItem`

The item in question.

`inShortText`

The short help text. This is what is displayed normally by the help tag system when the user hovers over the toolbar item with the mouse.

`inLongText`

The long help text. This is what is displayed by the help tag system when the user hovers over the toolbar item with the mouse and holds the command key down. This parameter is optional, you may pass `NULL`.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemSetIconRef

Sets the icon for a toolbar item.

```
OSStatus HIToolbarItemSetIconRef (
   HIToolbarItemRef inItem,
   IconRef inIcon
);
```

Parameters

`inItem`

The item in question.

`inIcon`

The icon reference. The IconRef is retained by the toolbar item. You can release the reference for this icon when `HIToolbarItemSetIconRef` returns.

Return Value

An operating system result code.

Availability
Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarItemSetImage

Sets the image for a toolbar item.

```
OSStatus HIToolbarItemSetImage (
   HIToolbarItemRef inItem,
   CGImageRef inImage
);
```

Parameters

`inItem`
: The item in question.

`inImage`
: The image. This image is retained by the toolbar item. You should release the image when you are finished with it.

Return Value
An operating system result code.

Discussion
Currently, the image should be no higher than 32 pixels. This image is used both in the toolbar as well as the configuration sheet, if the toolbar is configurable.

Availability
Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarItemSetLabel

Sets the label of a toolbar item.

```
OSStatus HIToolbarItemSetLabel (
   HIToolbarItemRef inItem,
   CFStringRef inLabel
);
```

Parameters

`inItem`

The item in question.

`inLabel`

The label. The toolbox will copy the string passed in.

Return Value

An operating system result code.

Discussion

This is what the toolbar view will display underneath the image. It is also used in the configuration palette for configurable toolbars.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarItemSetMenu

Sets the submenu for a toolbar item.

```
OSStatus HIToolbarItemSetMenu (
   HIToolbarItemRef inItem,
   MenuRef inMenu
);
```

Parameters

`inItem`

The item in question.

`inMenu`

The menu. It is retained by the toolbar item, so you can safely release it after calling this API. On Mac OS X v10.3 and later, you can pass `NULL` for this parameter to remove and release any menu that might be attached.

Return Value

An operating system result code.

Discussion

Normally, items do not have a submenu. You can attach one with this API. The submenu will, by default, show up both in the 'more items' indicator popup attached to the item name. It will also appear if the toolbar is in text only mode and the label is clicked. You should attach a Carbon Event handler to the menu to handle updating the menu items as appropriate before the menu is displayed.

Availability

Available in Mac OS X v10.2 and later

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarRemoveItemAtIndex

Removes an item at a given index from a toolbar.

```
OSStatus HIToolbarRemoveItemAtIndex (
   HIToolbarRef inToolbar,
   CFIndex inIndex
);
```

Parameters

`inToolbar`

The toolbar you are removing the item from.

`inIndex`

The index of the item to remove. This index is zero-based.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarSetDelegate

Sets the delegate object for a toolbar.

```
OSStatus HIToolbarSetDelegate (
   HIToolbarRef inToolbar,
   HIObjectRef inDelegate
);
```

Parameters

`inToolbar`

The toolbar whose delegate you want to set.

`inDelegate`

The HIObjectRef to act as the delegate.

Return Value

An operating system result code.

Discussion

If a toolbar has custom toolbar items, a delegate is required for the toolbar to work properly. The delegate is asked to create the toolbar items. If the delegate does not respond, the toolbar attempts to create the toolbar items, but it can only create standard items.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarSetDisplayMode

Sets the current display mode of a toolbar.

```
OSStatus HIToolbarSetDisplayMode (
   HIToolbarRef inToolbar,
   HIToolbarDisplayMode inDisplayMode
);
```

Parameters

`inToolbar`
: The toolbar whose display mode you want to set.

`inDisplayMode`
: The display mode. If the toolbar is visible, it will be redrawn as necessary.

Return Value

An operating system result code.

Availability

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in

`HIToolbar.h`

HIToolbarSetDisplaySize

Sets the current display size of a toolbar.

```
OSStatus HIToolbarSetDisplaySize (
   HIToolbarRef inToolbar,
   HIToolbarDisplaySize inSize
);
```

Parameters

`inToolbar`
: The toolbar whose display size you want to set.

`inSize`
: The display size. If the toolbar is visible, it will be redrawn as necessary.

Return Value

An operating system result code.

Availability
Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

HIToolbarSetItemsWithIdentifiers

Sets a toolbar's items all at once.

```
OSStatus HIToolbarSetItemsWithIdentifiers (
   HIToolbarRef inToolbar,
   CFArrayRef inArray
);
```

Parameters

`inToolbar`
The toolbar whose items you want to set.

`inArray`
The array of toolbar items to create.

Return Value
An operating system result code.

Discussion
This function allows you to set a toolbar's items all at once. The entries in the array `inArray` must be either CFStringRefs or CFDictionaryRefs. You do not need to use the same type for all entries; different entries can be of different types. If an array entry is a CFStringRef, the string must contain a toolbar item identifier. If an array entry is a CFDictionaryRef, the dictionary must contain a CFStringRef specifying a toolbar item identifier and may optionally contain a CFTypeRef specifying the toolbar item's configuration data, if the item requires it. The key for the identifier string is `kHIToolbarIdentifierKey` and the key for the configuration data string is `kHIToolbarDataKey`.

Availability
Available in Mac OS X v10.3 and later.

Not available to 64-bit applications.

Declared in
`HIToolbar.h`

Constants

Toolbar Attributes

Specify constants for toolbar attributes.

```
enum {
   kHIToolbarNoAttributes = 0,
   kHIToolbarAutoSavesConfig = (1 << 0),
   kHIToolbarIsConfigurable = (1 << 1),
   kHIToolbarValidAttrs = kHIToolbarAutoSavesConfig | kHIToolbarIsConfigurable
};
```

Constants

`kHIToolbarNoAttributes`

Pass this to indicate no attributes at all.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarAutoSavesConfig`

Pass this attribute to allow the toolbar to save its configuration automatically to your application's preferences. You must make sure to synchronize the preferences at some point to ensure it gets written to disk. The toolbar will also read its configuration from the preferences if this attribute is set.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarIsConfigurable`

This attribute indicates that the toolbar is configurable, that is, the user can drag items around and bring up the configuration palette, etc.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarValidAttrs`

The set of all valid toolbar attributes.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

Toolbar Command ID Constants

Specify constants for toolbar Command IDs.

```
enum {
   kHICommandCustomizeToolbar = 'tcfg',
   kHICommandShowToolbar = 'tbsh',
   kHICommandHideToolbar = 'tbhd',
   kHIToolbarCommandPressAction = 'tbpr'
};
```

Constants

`kHICommandCustomizeToolbar`

When sent to a window with a toolbar, this command causes the configuration sheet to appear. You can set a menu item's command to this command ID and it will be handled and updated automatically for you. (Available in Mac OX X v10.2 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHICommandShowToolbar`

Sending this command causes a window's toolbar to be shown. You can set a menu item's command to this ID and it will be handled and updated automatically for you. (Available in Mac OX X v10.2 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHICommandHideToolbar`

Sending this command causes a window's toolbar to be hidden. You can set a menu item's command to this ID and it will be handled and updated automatically for you. (Available in Mac OX X v10.2 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarCommandPressAction`

This command, when specified as a toolbar's command ID, causes a `kEventToolbarItemPerformAction` event to be generated when the toolbar item's menu item in the toolbar overflow menu is selected. If the item has any other command ID, a `kEventCommandProcess` event, containing the item's command ID, is generated instead. (Available in Mac OX X v10.2.3 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

Toolbar Display Mode Constants

Specify constants for toolbar display modes.

```
enum {
   kHIToolbarDisplayModeDefault = 0,
   kHIToolbarDisplayModeIconAndLabel = 1,
   kHIToolbarDisplayModeIconOnly = 2,
   kHIToolbarDisplayModeLabelOnly = 3
};
```

Constants

`kHIToolbarDisplayModeDefault`

Use the default display mode. Currently, this is defined as being both icon and label, but could change in the future.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarDisplayModeIconAndLabel`

Display the image as well as the label of the toolbar items.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarDisplayModeIconOnly`

Display only the image.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarDisplayModeLabelOnly`

Display only the label.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

Toolbar Display Size Constants

Specify constants for the display size of items in the toolbar.

```
enum {
   kHIToolbarDisplaySizeDefault = 0,
   kHIToolbarDisplaySizeNormal = 1,
   kHIToolbarDisplaySizeSmall = 2
};
```

Constants

`kHIToolbarDisplaySizeDefault`

This indicates to use the default display size. Currently, this is defined as using 32 x 32 icons ("normal" size).

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarDisplaySizeNormal`

This size uses a larger text and icon size.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarDisplaySizeSmall`

This size uses a smaller text and icon size.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

Toolbar Events

Specify toolbar Carbon event constants.

```
enum {
   kEventToolbarGetDefaultIdentifiers = 1,
   kEventToolbarGetAllowedIdentifiers = 2,
   kEventToolbarCreateItemWithIdentifier = 3,
   kEventToolbarCreateItemFromDrag = 4,
   kEventToolbarItemAdded = 5,
   kEventToolbarItemRemoved = 6,
   kEventToolbarDisplayModeChanged = 7,
   kEventToolbarDisplaySizeChanged = 8,
   kEventToolbarLayoutChanged = 9,
   kEventToolbarGetSelectableIdentifiers = 10,
   kEventToolbarBeginMultiChange = 12,
   kEventToolbarEndMultiChange = 13
};
```

Constants

`kEventToolbarGetDefaultIdentifiers`

This event is sent to the delegate to get a list of all of the default item identifiers that should be created for a toolbar. You are passed a mutable array to fill in with the identifiers. (

Available in Mac OS X v10.2.)

Declared in `HIToolbar.h`.

`kEventToolbarGetAllowedIdentifiers`

This event is sent to the delegate to get a list of all the items which could possibly be added to the toolbar. This is sent out when the configuration sheet is about to be displayed.You are passed a mutable array to fill in with the identifiers. (

Available in Mac OS X v10.2.)

Declared in `HIToolbar.h`.

`kEventToolbarCreateItemWithIdentifier`

This event is sent to the delegate to when the toolbar needs to create an item from an identifier. (

Available in Mac OS X v10.2.)

Declared in `HIToolbar.h`.

`kEventToolbarCreateItemFromDrag`

This event is sent to the delegate to when the toolbar needs to create an item from a drag. This allows you to be able to drag items into a toolbar that aren't normal toolbar items. You might use this to allow your toolbar to accept file system items, for example. (

Declared in `HIToolbar.h`.

Available in Mac OS X v10.2.)

`kEventToolbarItemAdded`

This event is sent when an item is added to the toolbar. The toolbar object sends this event to itself, so you need to install a handler on the toolbar to receive this event. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemRemoved`

This event is sent when an item is removed from toolbar. It is called after the item has already been removed. The toolbar object sends this event to itself, so you need to install a handler on the toolbar to receive this event. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarDisplayModeChanged`

This event is sent when the display mode is changed for a toolbar. The toolbar object sends this event to itself, so you need to install a handler on the toolbar to receive this event. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarDisplaySizeChanged`

This event is sent when the display size is changed for a toolbar. The toolbar object sends this event to itself, so you need to install a handler on the toolbar to receive this event. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarLayoutChanged`

Sent when the layout of a toolbar changes (either an item has been moved, or the entire contents have been replaced). Basically it is sent for changes that would require a total resync with the current state of things. The toolbar object sends this event to itself, so you must install a handler on the toolbar to receive this event. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarBeginMultiChange`

This event is sent when multiple attributes are going to be changed at the same time. For example, the display mode and size can change at the same time. When this happens, instead of reacting twice (once for a display mode change and once for a display size change), you can listen to see if more than one attribute is about to change and wait to redraw the toolbar until you receive the `kEventToolbarEndMultiChange` event. The toolbar object sends this event to itself, so you need to install a handler on the toolbar to receive this event. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarEndMultiChange`

This event is sent when multiple changes in the toolbar have been made. For details, see the description of `kEventToolbarBeginMultiChange`. The toolbar object sends this event to itself, so you need to install a handler on the toolbar to receive this event. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarGetSelectableIdentifiers`

This event is sent to the delegate after the user clicks a toolbar item in order to get a list of all the items that can acquire a selection highlight when clicked. You are passed a mutable array to fill in with the identifiers. If you pass back a non-empty array and the clicked item's identifier matches one of the identifiers that is in the list, the toolbar automatically draws that item with a selected highlight and removes highlighting from the previously selected item. Note that the selection only changes in the clicked window — it does not change in other windows that share the same toolbar. To share the selection across all windows that use the same toolbar, you need to manually change the `kHIToolbarItemSelected` attribute for the clicked item using `HIToolbarItemChangeAttributes` (page 16). In this case, you should not handle the `kEventToolbarGetSelectableIdentifiers` event. (Available in Mac OS X v10.4 and later.)

Available in OS X v10.4 and later.

Declared in `HIToolbar.h`.

Discussion

Table 1 lists the parameters and types for toolbar events.

Table 1 Parameter names and types for toolbar events

Event kind	Parameter name	Parameter type
`kEventToolbarGet-DefaultIdentifiers`	`kEventParamToolbar`	`typeHIToolbarRef`
	`kEventParamMutableArray`	`typeCFMutableArrayRef`
`kEventToolbarGet-AllowedIdentifiers`	`kEventParamToolbar`	`typeHIToolbarRef`
	`kEventParamMutableArray`	`typeCFMutableArrayRef`
`kEventToolbarGet-SelectableIdentifiers`	`kEventParamToolbar`	`typeHIToolbarRef`
	`kEventParamMutableArray`	`typeCFMutableArrayRef`
`kEventToolbarCreate-ItemWithIdentifier`	`kEventParamToolbar`	`typeHIToolbarRef`
	`kEventParamToolbar-ItemIdentifier`	`typeCFStringRef`
	`kEventParamToolbar-ItemConfigData`	`typeCFTypeRef`

	`kEventParamToolbarItem`	`typeHIToolbarItemRef`
`kEventToolbarCreate-ItemFromDrag`	`kEventParamDragRef`	`typeDragRef`
	`kEventParamToolbarItem`	`typeHIToolbarItemRef`
`kEventToolbarItemAdded`	`kEventParamToolbarItem`	`typeHIToolbarItemRef`
	`kEventParamIndex`	`typeCFIndex`
`kEventToolbarItemRemoved`	`kEventParamToolbarItem`	`typeHIToolbarItemRef`
`kEventToolbarDisplay-ModeChanged`	*None*	
`kEventToolbarDisplay-SizeChanged`	*None*	
`kEventToolbarLayout-Changed`	*None*	
`kEventToolbarBegin-MultiChange`	*None*	
`kEventToolbarEnd-MultiChange`	*None*	

Toolbar Event Parameters and Types

Specify constants for toolbar event parameters and types.

```
enum {
   kEventParamToolbar = 'tbar',
   kEventParamToolbarItem = 'tbit',
   kEventParamToolbarItemIdentifier = 'tbii',
   kEventParamToolbarItemConfigData = 'tbid',
   typeHIToolbarRef = 'tbar',
   typeHIToolbarItemRef = 'tbit'
};
```

Constants

`kEventParamToolbar`

> The toolbar to which an event was sent.
>
> Available in OS X v10.2 and later.
>
> Declared in `HIToolbar.h`.

`kEventParamToolbarItem`

> The toolbar item to which an event was sent.
>
> Available in OS X v10.2 and later.
>
> Declared in `HIToolbar.h`.

`kEventParamToolbarItemIdentifier`

> The toolbar item's identifier.
>
> Available in OS X v10.2 and later.
>
> Declared in `HIToolbar.h`.

`kEventParamToolbarItemConfigData`

> The toolbar item's configuration information.
>
> Available in OS X v10.2 and later.
>
> Declared in `HIToolbar.h`.

`typeHIToolbarRef`

> HIToolbarRef
>
> Available in OS X v10.2 and later.
>
> Declared in `HIToolbar.h`.

`typeHIToolbarItemRef`

> HIToolbarItemRef
>
> Available in OS X v10.2 and later.
>
> Declared in `HIToolbar.h`.

Toolbar Item Attributes

Specify constants for toolbar item attributes.

```
enum {
   kHIToolbarItemNoAttributes = 0,
   kHIToolbarItemAllowDuplicates = (1 << 0),
   kHIToolbarItemCantBeRemoved = (1 << 1),
   kHIToolbarItemAnchoredLeft = (1 << 2),
   kHIToolbarItemIsSeparator = (1 << 3),
   kHIToolbarItemSendCmdToUserFocus = (1 << 4),
   kHIToolbarItemLabelDisabled = (1 << 5),
   kHIToolbarItemDisabled = (1 << 6),
   kHIToolbarItemSelected = (1 << 7),
   kHIToolbarItemValidAttrs = kHIToolbarItemAllowDuplicates |  kHIToolbarItemIsSeparator |
kHIToolbarItemCantBeRemoved |  kHIToolbarItemAnchoredLeft | kHIToolbarItemSendCmdToUserFocus
 |  kHIToolbarItemLabelDisabled | kHIToolbarItemDisabled | kHIToolbarItemSelected,
   kHIToolbarItemMutableAttrs = kHIToolbarItemCantBeRemoved |  kHIToolbarItemAnchoredLeft |
 kHIToolbarItemLabelDisabled |  kHIToolbarItemDisabled | kHIToolbarItemSelected
};
```

Constants

`kHIToolbarItemNoAttributes`

Pass this to indicate no attributes at all.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarItemAllowDuplicates`

This indicates that an item can have more than one instance of itself in the toolbar. If this is not set, only one can be present. By default, the determining factor for what determines if two items are identical is the toolbar identifier.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarItemCantBeRemoved`

This item can be rearranged, but it cannot be removed from the toolbar by the user.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarItemAnchoredLeft`

This item cannot be moved at all by the user. It is anchored to the left of the toolbar. It is important that there not be any unanchored items to the left of this item, else dragging items around will be a bit wacky. You are responsible for making sure that anchored items are all on the left. This allows you to do toolbars like the one in the System Preferences app, where the first couple of items are stuck in place.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarItemIsSeparator`

This indicates the item acts as a separator. This means two things at present. First, it means that it automatically shows up as a divider line in a menu representation of the toolbar, and second it means the view that represents this item can draw in the full top to bottom space that the toolbar item occupies in the toolbar.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarItemSendCmdToUserFocus`

If this attribute bit is set, the command that gets sent out will be directed at the user focus instead of at the window the toolbar is attached to.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarItemLabelDisabled`

If this attribute bit is set, clicking on the label of an item does nothing. This attribute is only considered when a custom view is present. When set, this attribute makes the toolbar item view unresponsive to clicks while still allowing clicks to be sent to the custom view. When the toolbar is in text-only mode and this attribute is set, the label is displayed in a disabled (grayed) appearance. You might want to change this attribute when switching between display modes. For example, the view switcher in the Finder does not respond to clicks on the label when in icon and text mode, but it does respond to clicks when in text-only mode. To change this behavior on the fly, listen for `kEventToolbarItemViewConfigForMode` in your custom view and adjust this attribute as you want. (Available in Mac OS X 10.3 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kHIToolbarItemDisabled`

If this attribute bit is set, the item is disabled. Setting this attribute is the same as calling `HIToolbarItemSetEnabled` (page 28) on the item with the `inEnabled` parameter set to `false`. (Available in Mac OS X v10.4 and later.)

Available in OS X v10.4 and later.

Declared in `HIToolbar.h`.

`kHIToolbarItemSelected`

If this attribute bit is set, the item is drawn with a selected appearance. (Available in Mac OS X v10.4 and later.)

Available in OS X v10.4 and later.

Declared in `HIToolbar.h`.

`kHIToolbarItemValidAttrs`

The set of all valid toolbar item attributes.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kHIToolbarItemMutableAttrs`

The set of toolbar item attributes that can be changed by `HIToolbarItemChangeAttributes` (page 16) and `HIToolbarItemChangeAttributesInWindow` (page 17). Any other attribute must be specified when it is created.

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

Toolbar Item Events

Specify constants for toolbar item events.

```
enum {
   kEventToolbarItemImageChanged = 1,
   kEventToolbarItemLabelChanged = 2,
   kEventToolbarItemHelpTextChanged = 3,
   kEventToolbarItemCommandIDChanged = 4,
   kEventToolbarItemGetPersistentData = 5,
   kEventToolbarItemCreateCustomView = 6,
   kEventToolbarItemEnabledStateChanged = 7,
   kEventToolbarItemPerformAction = 8,
   kEventToolbarItemWouldAcceptDrop = 10,
   kEventToolbarItemAcceptDrop = 11,
   kEventToolbarItemSelectedStateChanged = 12
};
```

Constants

`kEventToolbarItemImageChanged`

This event is sent to the item when the image changes. Any interested parties can install handlers on the toolbar item to receive notifications. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemLabelChanged`

This event is sent to the item when the label changes. Any interested parties can install handlers on the toolbar item to receive notifications. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemHelpTextChanged`

This event is sent to the item when the help text changes. Any interested parties can install handlers on the toolbar item to receive notifications. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemCommandIDChanged`

This event is sent to the item when the command ID changes. Any interested parties can install handlers on the toolbar item to receive notifications. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemGetPersistentData`

This event is sent to the item when the toolbar is going to write out the configuration information for the item. Any custom items can listen for this event and add any extra information to what is written out into the config so that it can be reanimated later on from the same config data. Typically, you do not need to handle this event. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemCreateCustomView`

This event is sent to the toolbar item when it is time to create a view for it to display its contents. Implementors of custom toolbar items can install a handler for this event to create their own custom views for their items. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemEnabledStateChanged`

This event is sent to the item when the enabled state changes. Any interested parties can install handlers on the toolbar item to receive notifications. (Available in Mac OS X v10.3 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemPerformAction`

This event is sent when a toolbar item is clicked. Subclasses of toolbar items can choose to do special actions by overriding this event. If this event is unhandled, the default action of sending a command event will occur. (Available in Mac OS X v10.2 and later.)

Available in OS X v10.2 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemWouldAcceptDrop`

This event is sent when a toolbar item is clicked. Subclasses of toolbar items can choose to do special actions by overriding this event. If this event is unhandled, the default action of sending a command event will occur. (Available in Mac OS X v10.3 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemAcceptDrop`

This event is sent when a drag enters a toolbar item. If the toolbar item wants to accept drags, it should respond to this event and return `true` in the `kEventParamResult` parameter. The toolbar item will be highlighted appropriately. If you are using a custom view, you do not need to respond to this event because the view system provides full drag and drop capability in order to support custom items that use the standard view. (Available in Mac OS X v10.3 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemSelectedStateChanged`

This event is sent to the item itself when the selected state changes. Any interested parties can install handlers on the toolbar item to receive notifications. (Available in Mac OS X v10.4 and later.)

Available in OS X v10.4 and later.

Declared in `HIToolbar.h`.

Discussion

Table 2 lists the parameters and types for toolbar item events.

Table 2 Parameter names and types for toolbar item events

Event kind	Parameter name	Parameter type
`kEventToolbarItemImageChanged`	*None*	
`kEventToolbarItemLabelChanged`	*None*	
`kEventToolbarItemHelpTextChanged`	*None*	
`kEventToolbarItemCommandIDChanged`	*None*	
`kEventToolbarItemGet-PersistentData`	`kEventParamToolbar-ItemConfigData`	`typeCFTypeRef`
`kEventToolbarItemCreateCustomView`	`kEventParamControlRef`	`typeControlRef`
`kEventToolbarItem-EnabledStateChanged`	`kEventParamWindowRef` (Optional)	`typeWindowRef`
`kEventToolbarItemPerformAction`	*None*	
`kEventToolbarItemWouldAcceptDrop`	`kEventParamDragRef`	`typeDragRef`
	`kEventParamResult`	`typeBoolean`
`kEventToolbarItemAcceptDrop`	`kEventParamDragRef`	`typeDragRef`

`kEventToolbarItem–SelectedStateChanged`	`kEventParamWindowRef` (Optional)	`typeWindowRef`

Toolbar Item View Events

Specify constants for toolbar item view events.

```
enum {
   kEventToolbarItemViewConfigForMode = 3,
   kEventToolbarItemViewConfigForSize = 4,
   kEventToolbarItemViewEnterConfigMode = 5,
   kEventToolbarItemViewExitConfigMode = 6
};
```

Constants

`kEventToolbarItemViewConfigForMode`

This event notifies a toolbar item view that the toolbar's display mode has changed. A custom toolbar item view can respond in any way it sees fit. Usually, responding to this is not necessary; for example, when the toolbar goes from icon to text only the view is automatically hidden, so there is not much to do. This event is for informational purposes only. (Available in Mac OS X v10.3 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemViewConfigForSize`

This event notifies a toolbar item view that the toolbar's display size has changed. A custom toolbar item view can respond to this in any way it sees fit. Usually, responding to this is not necessary. Some custom views might need to flush metrics caches when the display size changes. (Available in Mac OS X v10.3 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemViewEnterConfigMode`

This event notifies a toolbar item view that configure mode has been entered. A custom toolbar item view can respond to this in any way it sees fit. For example, the space and flexible space mark themselves to draw a rectangle and merely invalidate so they get redrawn so they can be seen during configuration. (Available in Mac OS X v10.3 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventToolbarItemViewExitConfigMode`

This event notifies a toolbar item view that configure mode has been exited. A custom toolbar item view can respond to this event in any way it sees fit. (Available in Mac OS X v10.3 and later.)

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

Discussion

Table 3 lists the parameters and types for toolbar item view events.

Table 3 Parameter names and types for toolbar item view events

Event kind	Parameter name	Parameter type
`kEventToolbarItem-ViewConfigForMode`	`kEventParamToolbar-DisplayMode`	`typeHIToolbar-DisplayMode`
`kEventToolbarItem-ViewConfigForSize`	`kEventParamToolbar-DisplaySize`	`typeHIToolbar-DisplaySize`
`kEventToolbarItem-ViewEnterConfigMode`	*None*	
`kEventToolbarItem-ViewExitConfigMode`	*None*	

Toolbar View Display Event Parameters and Types

Specify constants for toolbar display event parameters and types.

```
enum {
   kEventParamToolbarDisplayMode = 'tbdm',
   kEventParamToolbarDisplaySize = 'tbds',
   typeHIToolbarDisplayMode = 'tbdm',
   typeHIToolbarDisplaySize = 'tbds'
};
```

Constants

`kEventParamToolbarDisplayMode`

Indicates that the display mode changed.

Available in OS X v10.3 and later.

Declared in `HIToolbar.h`.

`kEventParamToolbarDisplaySize`
: Indicates that the display size changed.
: Available in OS X v10.3 and later.
: Declared in `HIToolbar.h`.

`typeHIToolbarDisplayMode`
: HIToolbarDisplayMode.
: Available in OS X v10.3 and later.
: Declared in `HIToolbar.h`.

`typeHIToolbarDisplaySize`
: HIToolbarDisplaySize.
: Available in OS X v10.3 and later.
: Declared in `HIToolbar.h`.

Document Revision History

This table describes the changes to *HIToolbar Reference* .

Date	Notes
2005-09-08	Added updates and corrections.
2005-06-04	Updated for Mac OS X v10.4.
2003-05-15	Converted from latest Jaguar HeaderDoc.