Gestalt Manager Reference

Carbon > Resource Management

2007-10-31

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

Tables

Gestalt Manager Reference

Framework: CoreServices/CoreServices.h

Declared in Gestalt.h

Overview

You can use the Gestalt Manager and other system software facilities to investigate the operating environment. You need to know about the operating environment if your application takes advantage of hardware or software that is not available on all Macintosh computers. You can also use the Gestalt Manager to inform the operating system that your software is present and to find out about other software registered with the Gestalt Manager.

Carbon supports the Gestalt Manager. However, the results returned by Gestalt functions in Mac OS X are relevant only to your application's context. In general, the `Gestalt` function returns a different result when called from a Carbon application running in Mac OS X than it returns when called from a Classic application in Mac OS X, because these are different environments. For example, Carbon does not use a ROM, so calling `Gestalt` from a Carbon application on a beige G3 Macintosh computer and passing the `ROMVersion` selector returns a different result than `Gestalt` returns for a Classic application on the same computer. In fact, `Gestalt` could conceivably return different results for the same call by two Carbon applications.

Because `Gestalt` operates on a per-context basis in Mac OS X, you can't use it to share information (through pointers or any other means) among applications.

The `ROMVersion` and `machineType` selectors are not supported in Carbon.

In versions of the Mac OS prior to Mac OS X, the `NewGestalt` and `ReplaceGestalt` functions make use of the system heap, so that new or replaced selectors are available to any process. In Mac OS X, however, there is no system heap, and the selectors are available only on a per-context basis.

Functions by Task

Getting and Setting Gestalt Selector Codes and Values

`Gestalt` (page 13)
Obtains information about the operating environment.

`NewGestaltValue` (page 15)
Installs a new `Gestalt` selector code and a value that `Gestalt` returns for that selector.

`SetGestaltValue` (page 16)
: Sets the value the function `Gestalt` will return for a specified selector code, installing the selector if it was not already installed.

`ReplaceGestaltValue` (page 16)
: Replaces the value that the function `Gestalt` returns for a specified selector code with the value provided to the function.

`DeleteGestaltValue` (page 12)
: Deletes a `Gestalt` selector code so that it is no longer recognized by `Gestalt`.

`NewGestalt` (page 121) Deprecated in Mac OS X v10.3
: Adds a selector code to those already recognized by `Gestalt`. (Deprecated. Use `NewGestaltValue` (page 15) instead.)

`ReplaceGestalt` (page 122) Deprecated in Mac OS X v10.3
: Replaces the selector function associated with an existing selector code. (Deprecated. Use `NewGestaltValue` (page 15) instead.)

Working With Universal Procedure Pointers for Gestalt Selector Functions

`NewSelectorFunctionUPP` (page 15)
: Creates a universal procedure pointer (UPP) to a selector callback function.

`DisposeSelectorFunctionUPP` (page 13)
: Disposes of a universal procedure pointer to a selector callback function.

`InvokeSelectorFunctionUPP` (page 14)
: Invokes a selector callback function.

Functions

DeleteGestaltValue

Deletes a `Gestalt` selector code so that it is no longer recognized by `Gestalt`.

```
OSErr DeleteGestaltValue (
   OSType selector
);
```

Parameters

`selector`
: The selector code you want to delete. This should be a four-character sequence similar to those defined in "Gestalt Manager Constants" (page 18).

Return Value
A result code. See "Gestalt Manager Result Codes" (page 119).

Discussion
After calling this function, subsequent query or replacement calls for the selector code will fail as if the selector had never been installed.

In Mac OS X, the selector is on a per-context basis. You cannot use this function to affect another process.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Gestalt.h`

DisposeSelectorFunctionUPP

Disposes of a universal procedure pointer to a selector callback function.

```
void DisposeSelectorFunctionUPP (
   SelectorFunctionUPP userUPP
);
```

Parameters

userUPP

The universal procedure pointer you want to dispose of.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Gestalt.h`

Gestalt

Obtains information about the operating environment.

```
OSErr Gestalt (
   OSType selector,
   SInt32 *response
);
```

Parameters

selector

The selector code for the information you need. You can provide any of the four-character sequences defined in "Gestalt Manager Constants" (page 18).

response

On input, `Gestalt` interprets this parameter as an address at which it is to place the result returned by the selector function. `Gestalt` ignores any information already at this address.

On return, a pointer to the requested information whose format depends on the selector code specified in the selector parameter. Note that the `Gestalt` function returns the response from all selectors in a long word, which occupies 4 bytes. When not all 4 bytes are needed, the significant information appears in the low-order byte or bytes.

Return Value
A result code. See "Gestalt Manager Result Codes" (page 119).

Discussion
The Apple-defined selector codes fall into two categories: environmental selectors, which supply specific environmental information you can use to control the behavior of your application, and informational selectors, which can't supply information you can use to determine what hardware or software features are available. You can use one of the selector codes defined by Apple or a selector code defined by a third-party product.

Selectors with the suffix `Attr` return a 32-bit response value in which the individual bits represent specific attributes. The constants listed for these response values represent bit numbers.

Special Considerations

When passed one of the Apple-defined selector codes, the `Gestalt` function does not move or purge memory and therefore may be called even at interrupt time. However, selector functions associated with non-Apple selector codes might move or purge memory, and third-party software can alter the Apple-defined selector functions. Therefore, it is safest always to assume that `Gestalt` could move or purge memory.

Version Notes

The `ROMVersion` and `machineType` selectors are not supported in Carbon.

In general, the `Gestalt` function returns a different result when called from a Carbon application running in Mac OS X than it returns when called from a Classic application in Mac OS X, because these are different environments. For example, Carbon does not use a ROM, so calling `Gestalt` from a Carbon application on a beige G3 Macintosh computer and passing the `ROMVersion` selector returns a different result than Gestalt returns for a Classic application on the same computer.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

BSDLLCTest

QTCarbonShell

QTMetaData

SoftVDigX

WhackedTV

Declared In

`Gestalt.h`

InvokeSelectorFunctionUPP

Invokes a selector callback function.

```
OSErr InvokeSelectorFunctionUPP (
   OSType selector,
   long *response,
   SelectorFunctionUPP userUPP
);
```

Parameters

selector

The selector code for the function you want to invoke. You can provide any of the four-character sequences defined in "Gestalt Manager Constants" (page 18).

response

On ouput, the value associated with the selector code.

userUPP

A universal procedure pointer to the selector callback function you want to invoke.

Return Value

A result code. See "Gestalt Manager Result Codes" (page 119).

Discussion
You should not need to call this function, as the operating system invokes your selecor callback for you.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Gestalt.h`

NewGestaltValue

Installs a new `Gestalt` selector code and a value that `Gestalt` returns for that selector.

```
OSErr NewGestaltValue (
   OSType selector,
   SInt32 newValue
);
```

Parameters

`selector`

The selector code you want to add. This should be a four-character sequence similar to those defined in "Gestalt Manager Constants" (page 18).

`newValue`

The value to return for the new selector code.

Return Value
A result code. See "Gestalt Manager Result Codes" (page 119).

Discussion
You call the function `NewGestaltValue` when the specified selector is not already installed and you don't want to override an existing value.

In Mac OS X, the new selector and value are on a per-context basis. That means they are available only to the application or other code that installs them. You cannot use this function to make information available to another process.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Gestalt.h`

NewSelectorFunctionUPP

Creates a universal procedure pointer (UPP) to a selector callback function.

```
SelectorFunctionUPP NewSelectorFunctionUPP (
   SelectorFunctionProcPtr userRoutine
);
```

Parameters

`userRoutine`

The address of the selector callback function.

Return Value
On return, a universal procedure pointer to the selector callback function. See the description of the `SelectorFunctionUPP` data type.

Discussion
You use the `NewSelectorFunctionUPP` function to create a UPP to pass to the `NewGestalt` or `ReplaceGestalt` functions.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Gestalt.h`

ReplaceGestaltValue

Replaces the value that the function `Gestalt` returns for a specified selector code with the value provided to the function.

```
OSErr ReplaceGestaltValue (
   OSType selector,
   SInt32 replacementValue
);
```

Parameters

`selector`
The selector code you want to add. This should be a four-character sequence similar to those defined in "Gestalt Manager Constants" (page 18).

`replacementValue`
The replacement `Gestalt` value for the selector code.

Return Value
A result code. See "Gestalt Manager Result Codes" (page 119).

Discussion
You use the function `ReplaceGestaltValue` to replace an existing value. You should not call this function to introduce a value that doesn't already exist; instead call the function `NewGestaltValue` (page 15).

In Mac OS X, the selector and replacement value are on a per-context basis. That means they are available only to the application or other code that installs them. You cannot use this function to make information available to another process.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Gestalt.h`

SetGestaltValue

Sets the value the function `Gestalt` will return for a specified selector code, installing the selector if it was not already installed.

```
OSErr SetGestaltValue (
   OSType selector,
   SInt32 newValue
);
```

Parameters

selector

The selector code you want to set. This should be a four-character sequence similar to those defined in "Gestalt Manager Constants" (page 18).

newValue

The new `Gestalt` value for the selector code.

Return Value

A result code. See "Gestalt Manager Result Codes" (page 119).

Discussion

You use `SetGestaltValue` to establish a value for a selector, without regard to whether the selector was already installed.

In Mac OS X, the selector and new value are on a per-context basis. That means they are available only to the application or other code that installs them. You cannot use this function to make information available to another process.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Gestalt.h`

Callbacks

SelectorFunctionProcPtr

Defines a pointer to a selector callback function that returns information associated with your own selector code.

```
typedef OSErr (*SelectorFunctionProcPtr)
(
    OSType selector,
    long * response
);
```

If you name your function `MySelectorFunctionProc`, you would declare it like this:

```
OSErr SelectorFunctionProcPtr (
    OSType selector,
    long * response
);
```

Parameters

selector

The selector code that triggers the function. This should be a four-character sequence similar to those defined in "Gestalt Manager Constants" (page 18).

response

On output, the information associated with the selector code.

Return Value
A result code. See "Gestalt Manager Result Codes" (page 119).

Discussion
Your selector function places the requested information in the `response` parameter and returns a result code. If the information is not available, the selector function returns the appropriate error code, which the `Gestalt` function returns as its function result.

A selector function can call `Gestalt` or even other selector functions.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Gestalt.h`

Data Types

SelectorFunctionUPP

Defines a universal procedure pointer to a selector function callback.

```
typedef SelectorFunctionProcPtr SelectorFunctionUPP;
```

Discussion
You can obtain a `SelectorFunctionUPP` by calling the function `NewSelectorFunctionUPP` (page 15). For more information, see `SelectorFunctionProcPtr` (page 17).

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Gestalt.h`

Constants

Addressing Mode Attribute Selectors

Specify feature availability information for the addressing mode of the operating system.

```
enum {
    gestaltAddressingModeAttr = 'addr',
    gestalt32BitAddressing = 0,
    gestalt32BitSysZone = 1,
    gestalt32BitCapable = 2
};
```

Constants

`gestaltAddressingModeAttr`
The `Gestalt` selector you pass to determine the addressing mode attributes that are present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt32BitAddressing`
If `true`, the operating system is using 32-bit addressing mode.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt32BitSysZone`
If `true`, there is a 32-bit compatible system zone.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt32BitCapable`
If `true`, Machine is 32-bit capable.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion

Before calling any function dependent on memory, your application should pass the selector `gestaltAddressingModeAttr` to the Gestalt function to determine the addressing mode attributes that are present.

Admin Attribute Selectors

Specify feature availability for Macintosh Manager adminstration software.

```
enum {
    gestaltAdminFeaturesFlagsAttr = 'fred',
    gestaltFinderUsesSpecialOpenFoldersFile = 0
};
```

Constants

`gestaltAdminFeaturesFlagsAttr`
The `Gestalt` selector you pass to determine the admin features that are present. This selector is typically used by the system.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFinderUsesSpecialOpenFoldersFile`
Specifies that the Finder uses a special file to store the list of open folders.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

AFP Client Selectors

Specify version and feature availability information for the AFP client.

```
enum {
    gestaltAFPClient = 'afps',
    gestaltAFPClientVersionMask = 0x0000FFFF,
    gestaltAFPClient3_5 = 0x0001,
    gestaltAFPClient3_6 = 0x0002,
    gestaltAFPClient3_6_1 = 0x0003,
    gestaltAFPClient3_6_2 = 0x0004,
    gestaltAFPClient3_6_3 = 0x0005,
    gestaltAFPClient3_7 = 0x0006,
    gestaltAFPClient3_7_2 = 0x0007,
    gestaltAFPClient3_8 = 0x0008,
    gestaltAFPClient3_8_1 = 0x0009,
    gestaltAFPClient3_8_3 = 0x000A,
    gestaltAFPClient3_8_4 = 0x000B,
    gestaltAFPClientAttributeMask = 0xFFFF0000,
    gestaltAFPClientCfgRsrc = 16,
    gestaltAFPClientSupportsIP = 29,
    gestaltAFPClientVMUI = 30,
    gestaltAFPClientMultiReq = 31
};
```

Alias Manager Attribute Selectors

Specify feature availability information for the Alias Manager.

```
enum {
    gestaltAliasMgrAttr = 'alis',
    gestaltAliasMgrPresent = 0,
    gestaltAliasMgrSupportsRemoteAppletalk = 1,
    gestaltAliasMgrSupportsAOCEKeychain = 2,
    gestaltAliasMgrResolveAliasFileWithMountOptions = 3,
    gestaltAliasMgrFollowsAliasesWhenResolving = 4,
    gestaltAliasMgrSupportsExtendedCalls = 5,
    gestaltAliasMgrSupportsFSCalls = 6,
    gestaltAliasMgrPrefersPath = 7
};
```

Constants

`gestaltAliasMgrAttr`

The selector you pass to the `Gestalt` function to determine the Alias Manager attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Appearance Manager Attribute Selectors

Specify feature availability information for the Appearance Manager.

```
enum {
    gestaltAppearanceAttr = 'appr',
    gestaltAppearanceExists = 0,
    gestaltAppearanceCompatMode = 1
};
```

Constants

`gestaltAppearanceAttr`
The `Gestalt` selector passed to determine whether the Appearance Manager is present. Produces a 32-bit value whose bits you should test to determine which Appearance Manager features are available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAppearanceExists`
If this bit is set, Appearance Manager functions are available. To determine which version of the Appearance Manager is installed, check for the presence of the `Gestalt` selector `gestaltAppearanceVersion`. If this bit is not set, Appearance Manager functions are not available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAppearanceCompatMode`
If this bit is set, systemwide platinum appearance is off. When systemwide platinum appearance is off, the Appearance Manager does not auto-map standard System 7 definition functions to their Mac OS 8 equivalents (for those applications that have not called `RegisterAppearanceClient`). If this bit is not set, systemwide platinum appearance is on, and the Appearance Manager auto-maps standard System 7 definition functions to their Mac OS 8 equivalents for all applications.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion

Before calling any functions dependent upon the Appearance Manager's presence, your application should pass the selector `gestaltAppearanceAttr` to the `Gestalt` function to determine whether the Appearance Manager is present. To determine which version of the Appearance Manager is installed, your application should check for the presence of the `Gestalt` selector `gestaltAppearanceVersion`.

Appearance Manager Version Selector

Specifies version information for the Appearance Manager.

```
enum {
    gestaltAppearanceVersion = 'apvr'
};
```

Constants

`gestaltAppearanceVersion`
The `Gestalt` selector passed to determine which version of the Appearance Manager is installed. If this selector exists, Appearance Manager 1.0.1 (or later) is installed. The version number of the currently installed Appearance Manager is returned in the low-order word of the result in binary code decimal format (for example, version 1.0.1 would be 0x0101). If this selector does not exist but `gestaltAppearanceAttr` does, Appearance Manager 1.0 is installed.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Apple Event Manager Attribute Selectors

Specify feature availability information for the Apple Event Manager.

```
enum {
    gestaltAppleEventsAttr = 'evnt',
    gestaltAppleEventsPresent = 0,
    gestaltScriptingSupport = 1,
    gestaltOSLInSystem = 2,
    gestaltSupportsApplicationURL = 4
};
```

Constants

`gestaltAppleEventsAttr`

A selector you pass to the `Gestalt` function. If the Apple Event Manager is not present, the `Gestalt` function returns an error value; otherwise, it returns `noErr` and supplies, in the `response` parameter, a 32-bit value whose bits specify which features of the Apple Event Manager are available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAppleEventsPresent`

A `Gestalt` attribute constant. If the bit specified by this constant is set in the `response` parameter value supplied by `Gestalt` for the `gestaltAppleEventsAttr` selector, the Apple Event Manager is present and installed in the system.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltScriptingSupport`

A `Gestalt` attribute constant. If the bit specified by this constant is set in the `response` parameter value supplied by `Gestalt` for the `gestaltAppleEventsAttr` selector, the Open Scripting Architecture (OSA) is available to provide scripting support. The OSA is described in "Scripting Components".

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltOSLInSystem`

A `Gestalt` attribute constant. If the bit specified by this constant is set in the `response` parameter value supplied by `Gestalt` for the `gestaltAppleEventsAttr` selector, the Object Support Library (OSL) is part of the system.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSupportsApplicationURL`

Available in Mac OS X v10.1 and later.

Declared in `Gestalt.h`.

AppleScript Attribute Selectors

Specify feature availability information for AppleScript.

```
enum {
    gestaltAppleScriptAttr = 'ascr',
    gestaltAppleScriptPresent = 0,
    gestaltAppleScriptPowerPCSupport = 1
};
```

Constants

`gestaltAppleScriptAttr`

A selector you pass to the `Gestalt` function. If AppleScript is not present, the `Gestalt` function returns an error value; otherwise, it returns `noErr` and supplies, in the `response` parameter, a 32-bit value whose bits specify which AppleScript features are available.

The only bit currently in use specifies whether AppleScript is present. You can test this bit with the constant `gestaltAppleScriptPresent`.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAppleScriptPresent`

A `Gestalt` attribute constant. If the bit specified by this constant is set in the `response` parameter value supplied by `Gestalt` for the `gestaltAppleScriptAttr` selector, AppleScript is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAppleScriptPowerPCSupport`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

AppleScript Version Selector

Specifies version information for AppleScript.

```
enum {
    gestaltAppleScriptVersion = 'ascv'
};
```

Constants

`gestaltAppleScriptVersion`

A selector you pass to the `Gestalt` function. If AppleScript is not present, the `Gestalt` function returns an error value; otherwise, it returns `noErr` and supplies, in the `response` parameter, a 32-bit AppleScript version number.

The low word of the 32-bit AppleScript version number specifies the current AppleScript version, while the high word specifies a compatibility version. For example, for AppleScript 1.3.7, which shipped with Mac OS 8.6, the value returned in the `response` parameter, viewed as a hex number, is 0x01100137. The low word, 0x0137, refers to the current AppleScript version. The high word, 0x0110, refers to the compatibility version number—scripts written for AppleScript versions 1.1.0 and later will run with AppleScript version 1.3.7.

The Version Notes section provides additional information about AppleScript versions and features.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Version Notes
For System 7.0 and 7.1, AppleScript and the Apple Event Manager were optional installs. Starting with System 7.5, they are part of a standard install, so if you've already checked for the presence of System 7.5 or later, you'll know that the Apple Event Manager is available, though AppleScript could be disabled using the Extensions Manager.

If you need features that are only available starting with a specific version of AppleScript, call `Gestalt` with the `gestaltAppleScriptVersion` selector to obtain the version number, then determine whether it is greater than or equal to the version your application requires.

AppleTalk Driver Version Selector

Specifies version information for the AppleTalk driver.

```
enum {
    gestaltATalkVersion = 'atkv'
};
```

Constants

`gestaltATalkVersion`

The version number of the AppleTalk driver, in the format introduced with AppleTalk version 56. The version is stored in the high 3 bytes of the return value.

Byte 3 contains the major revision number, byte 2 contains the minor revision number, and byte 1 contains a constant that represents the release stage.

For example, if you call the `Gestalt` function with the 'atkv' selector when AppleTalk version 57 is loaded, you receive the long integer response value $39008000.

Byte 0 always contains 0.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

AppleTalk Version Selector

Specifies version information for AppleTalk.

```
enum {
    gestaltAppleTalkVersion = 'atlk'
};
```

Constants

`gestaltAppleTalkVersion`

The version number of the AppleTalk driver (in particular, the .MPP driver) currently installed. The version number is placed into the low-order byte of the result; ignore the three high-order bytes. If an AppleTalk driver is not currently open, the `response` parameter is 0.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

ATSUI Attribute Selectors

Specify feature availability for Apple Type Services for Unicode Imaging.

```
enum {
    gestaltATSUFeatures = 'uisf',
    gestaltATSUTrackingFeature = 0x00000001,
    gestaltATSUMemoryFeature = 0x00000001,
    gestaltATSUFallbacksFeature = 0x00000001,
    gestaltATSUGlyphBoundsFeature = 0x00000001,
    gestaltATSULineControlFeature = 0x00000001,
    gestaltATSULayoutCreateAndCopyFeature = 0x00000001,
    gestaltATSULayoutCacheClearFeature = 0x00000001,
    gestaltATSUTextLocatorUsageFeature = 0x00000002,
    gestaltATSULowLevelOrigFeatures = 0x00000004,
    gestaltATSUFallbacksObjFeatures = 0x00000008,
    gestaltATSUIgnoreLeadingFeature = 0x00000008,
    gestaltATSUByCharacterClusterFeature = 0x00000010,
    gestaltATSUAscentDescentControlsFeature = 0x00000010,
    gestaltATSUHighlightInactiveTextFeature = 0x00000010,
    gestaltATSUPositionToCursorFeature = 0x00000010,
    gestaltATSUBatchBreakLinesFeature = 0x00000010,
    gestaltATSUTabSupportFeature = 0x00000010,
    gestaltATSUDirectAccess = 0x00000010,
    gestaltATSUDecimalTabFeature  = 0x00000020,
    gestaltATSUBiDiCursorPositionFeature = 0x00000020,
    gestaltATSUNearestCharLineBreakFeature = 0x00000020,
    gestaltATSUHighlightColorControlFeature = 0x00000020,
    gestaltATSUUnderlineOptionsStyleFeature = 0x00000020,
    gestaltATSUStrikeThroughStyleFeature = 0x00000020,
    gestaltATSUDropShadowStyleFeature = 0x00000020
};
```

Constants

`gestaltATSUFeatures`

Specifies the ATSUI features available on the user's system. You pass this selector to the `Gestalt` function. On return, the `Gestalt` function passes back a value that represents the features available in the version of ATSUI installed on the user's system.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltATSUTrackingFeature`

If the bit specified by this mask constant is set, the functions `ATSUCountFontTracking` and `ATSUGetIndFontTracking` are available.

Available beginning with ATSUI 1.1.

Declared in `Gestalt.h`.

`gestaltATSUMemoryFeature`

If the bit specified by this mask is set, the functions `ATSUCreateMemorySetting`, `ATSUSetCurrentMemorySetting`, `ATSUGetCurrentMemorySetting`, and `ATSUDisposeMemorySetting` are available.

Available beginning with ATSUI 1.1.

Declared in `Gestalt.h`.

`gestaltATSUFallbacksFeature`

If the bit specified by this mask is set, the functions `ATSUSetFontFallbacks` and `ATSUGetFontFallbacks` are available.

Available beginning with ATSUI 1.1.

Declared in `Gestalt.h`.

`gestaltATSUGlyphBoundsFeature`
If the bit specified by this mask is set, the function `ATSUGetGlyphBounds` is available.

Available beginning with ATSU 1.1.

Declared in `Gestalt.h`.

`gestaltATSULineControlFeature`
If the bit specified by this mask is set, the functions `ATSUCopyLineControls`, `ATSUSetLineControls`, `ATSUGetLineControl`, `ATSUGetAllLineControls`, and `ATSUClearLineControls` are available.

Available beginning with ATSUI 1.1.

Declared in `Gestalt.h`.

`gestaltATSULayoutCreateAndCopyFeature`
If the bit specified by this mask is set, the function `ATSUCreateAndCopyTextLayout` is available.

Available beginning with ATSUI 1.1.

Declared in `Gestalt.h`.

`gestaltATSULayoutCacheClearFeature`
If the bit specified by this mask is set, the function `ATSUClearLayoutCache` is available.

Available beginning with ATSUI 1.1.

Declared in `Gestalt.h`.

`gestaltATSUTextLocatorUsageFeature`
If the bit specified by this mask is set, the text-break locator attribute is available for both style and text layout objects.

Available beginning with ATSUI 1.2.

Declared in `Gestalt.h`.

`gestaltATSULowLevelOrigFeatures`
If the bit specified by this mask is set, the low-level features introduced in ATSUI version 2.0 are available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltATSUFallbacksObjFeatures`
If the bit specified by this mask is set, `ATSUFontFallbacks` objects are available.

Available beginning with ATSUI version 2.3.

Declared in `Gestalt.h`.

`gestaltATSUIgnoreLeadingFeature`
If the bit specified by this mask is set, the line layout option (`kATSIgnoreFontLeadingTag`) to ignore the font leading value is available.

Available beginning with ATSUI version 2.3.

Declared in `Gestalt.h`.

`gestaltATSUByCharacterClusterFeature`
If the bit specified by this mask is set, ATSUI cursor movement types are available.

Available beginning with ATSUI version 2.4.

Declared in `Gestalt.h`.

`gestaltATSUAscentDescentControlsFeature`
If the bit specified by this mask is set, ascent and descent controls (`kATSUDescentTag` and `kATSUAscentTag`) are available.

Available beginning with ATSUI version 2.4.

Declared in `Gestalt.h`.

`gestaltATSUHighlightInactiveTextFeature`
If the bit specified by this mask is set, the highlight inactive text feature is available.

Available beginning with ATSUI version 2.4.

Declared in `Gestalt.h`.

`gestaltATSUPositionToCursorFeature`
If the bit specified by this mask is set, the position-to-cursor feature is available.

Available beginning with ATSUI version 2.4.

Declared in `Gestalt.h`.

`gestaltATSUBatchBreakLinesFeature`
If the bit specified by this mask is set, the `ATSUBatchBreakLines` function is available.

Available beginning with ATSUI version 2.4.

Declared in `Gestalt.h`.

`gestaltATSUTabSupportFeature`
If the bit specified by this mask is set, support for tabs is available.

Available beginning with ATSUI version 2.4.

Declared in `Gestalt.h`.

`gestaltATSUDirectAccess`
If the bit specified by this mask is set, ATSU direct-access functions are available. These functions let you access glyph information directly.

Available beginning with ATSUI version 2.4.

Declared in `Gestalt.h`.

`gestaltATSUDecimalTabFeature`
If the bit specified by this mask is set, your application can set a decimal tab character.

Available beginning with ATSUI version 2.5.

Declared in `Gestalt.h`.

`gestaltATSUBiDiCursorPositionFeature`
If the bit specified by this mask is set, support for bidirectional cursor positioning is available.

Available beginning with ATSUI version 2.5.

Declared in `Gestalt.h`.

`gestaltATSUNearestCharLineBreakFeature`
If the bit specified by this mask is set, the nearest character line break feature is available.

Available beginning with ATSUI version 2.5.

Declared in `Gestalt.h`.

`gestaltATSUHighlightColorControlFeature`
If the bit specified by this mask is set, your application can control highlight color.

Available beginning with ATSUI version 2.5.

Declared in `Gestalt.h`.

`gestaltATSUUnderlineOptionsStyleFeature`
If the bit specified by this mask is set, underline options are available.

Available beginning with ATSUI version 2.5.

Declared in `Gestalt.h`.

`gestaltATSUStrikeThroughStyleFeature`
If the bit specified by this mask is set, strike through styles are available.

Available beginning with ATSUI version 2.5.

Declared in `Gestalt.h`.

`gestaltATSUDropShadowStyleFeature`
If the bit specified by this mask is set, drop shadow features are available.

Available beginning with ATSUI version 2.5.

Declared in `Gestalt.h`.

Discussion
You can pass the `gestaltATSUFeature` selector to the `Gestalt` function to obtain a value that specifies which ATSUI features are available on the user's system.

You can pass the `gestaltATSUVersion` selector to the `Gestalt` function to determine which version of ATSUI is installed on the user's system. See "ATSUI Version Selectors" (page 28) for more information

ATSUI Version Selectors

Specify version information for Apple Type Service for Unicode Imaging.

```
enum {
    gestaltATSUVersion = 'uisv',
    gestaltOriginalATSUVersion = (1 << 16),
    gestaltATSUUpdate1 = (2 << 16),
    gestaltATSUUpdate2 = (3 << 16),
    gestaltATSUUpdate3 = (4 << 16),
    gestaltATSUUpdate4 = (5 << 16),
    gestaltATSUUpdate5 = (6 << 16),
    gestaltATSUUpdate6 = (7 << 16),
    gestaltATSUUpdate7 = (8 << 16)
};
```

Constants
`gestaltATSUVersion`
Specifies the version of ATSUI installed on the user's system. You pass this selector to the `Gestalt` function. On return, the `Gestalt` function passes back a value that represents the version of ATSUI installed on the user's system.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltOriginalATSUVersion`
Indicates that version 1.0 of ATSUI is installed on the user's system.

Available beginning with ATSUI 1.0.

Declared in `Gestalt.h`.

`gestaltATSUUpdate1`
Indicates that version 1.1 of ATSUI is installed on the user's system.

Available beginning with ATSUI 1.1.

Declared in `Gestalt.h`.

`gestaltATSUUpdate2`
Indicates that version 1.2 of ATSUI is installed on the user's system.

Available beginning with ATSUI 1.2.

Declared in `Gestalt.h`.

`gestaltATSUUpdate3`
Indicates that version 2.0 of ATSUI is installed on the user's system.

Available beginning with ATSUI 2.0.

Declared in `Gestalt.h`.

`gestaltATSUUpdate4`
Indicates that ATSUI for a version of Mac OS X from 10.0.1 through 10.0.4 is installed on the user's system.

Available beginning with Mac OS X version 10.0.1.

Declared in `Gestalt.h`.

`gestaltATSUUpdate5`
Indicates that version 2.3 of ATSUI is installed on the user's system. Available beginning with ATSUI 2.3, in Mac OS X version 10.1.

Available in Mac OS X v10.1 and later.

Declared in `Gestalt.h`.

`gestaltATSUUpdate6`
Indicates that version 2.4 of ATSUI is installed on the user's system. Available beginning with ATSUI 2.4, in Mac OS X version 10.2.

Available in Mac OS X v10.2 and later.

Declared in `Gestalt.h`.

`gestaltATSUUpdate7`
Indicates that version 2.5 of ATSUI is installed on the user's system. Available beginning with ATSUI 2.5, in Mac OS X version 10.3.

Available in Mac OS X v10.3 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any functions dependent upon ATSUI, you should pass the `gestaltATSUVersion` selector to the `Gestalt` function to determine which version of ATSUI is available.

You can pass the `gestaltATSUFeatures` selector to the `Gestalt` function to determine which features of ATSUI are available. See "ATSUI Attribute Selectors" (page 24) for more information.

ATA Manager Attribute Selectors

Specify feature availability information for the ATA Manager.

```
enum {
    gestaltATAAttr = 'ata ',
    gestaltATAPresent = 0
};
```

AUX Version Selector

Specifies version information for A/UX.

```
enum {
    gestaltAUXVersion = 'a/ux'
};
```

Constants

`gestaltAUXVersion`

The version of A/UX if it is currently executing. The result is placed into the low-order word of the response parameter. If A/UX is not executing, the `Gestalt` function returns `gestaltUnknownErr`.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

AVL Tree Attribute Selectors

Specify feature availability information for AVL tree routines.

```
enum {
    gestaltAVLTreeAttr = 'tree',
    gestaltAVLTreePresentBit = 0,
    gestaltAVLTreeSupportsHandleBasedTreeBit = 1,
    gestaltAVLTreeSupportsTreeLockingBit = 2
};
```

Bus Clock Version Selector

Specifies version information for the bus clock speed.

```
enum {
    gestaltBusClkSpeed = 'bclk'
};
```

Carbon Version Selector

Specifies version information for Carbon.

```
enum {
    gestaltCarbonVersion = 'cbon'
};
```

Classic Compatibility Attribute Selectors

Specify feature availability for the Classic environment.

```
enum {
    gestaltMacOSCompatibilityBoxAttr = 'bbox',
    gestaltMacOSCompatibilityBoxPresent = 0,
    gestaltMacOSCompatibilityBoxHasSerial = 1,
    gestaltMacOSCompatibilityBoxless = 2
};
```

CloseView Attribute Selectors

Specify feature availability information for CloseView.

```
enum {
    gestaltCloseViewAttr = 'BSDa',
    gestaltCloseViewEnabled = 0,
    gestaltCloseViewDisplayMgrFriendly = 1
};
```

Code Fragment Manager Attribute Selectors

Specify feature availability information for the Code Fragment Manager.

```
enum {
    gestaltCFMAttr = 'cfrg',
    gestaltCFMPresent = 0,
    gestaltCFMPresentMask = 0x0001,
    gestaltCFM99Present = 2,
    gestaltCFM99PresentMask = 0x0004
};
```

Collection Manager Version Selector

Specify version information for the Collection manager.

```
enum {
    gestaltCollectionMgrVersion = 'cltn'
};
```

Constants

`gestaltCollectionMgrVersion`

Collection Manager version.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Color Picker Version Selectors

Specify version information for the Color Picker.

```
enum {
    gestaltColorPickerVersion = 'cpkr',
    gestaltColorPicker = 'cpkr'
};
```

Discussion
To test for the availability and version of the Color Picker Manager, use the `Gestalt` function with the selector defined by this enumerator.

If the `Gestalt` function returns a value of 00000200, version 2.0 of the Color Picker Manager is available. If the `Gestalt` function returns a value of 00000100, version 1.0 (that is, the original Color Picker Package) is available.

ColorSync Manager Attribute Selectors

Specify feature availability information for the ColorSync Manager.

```
enum {
    gestaltColorMatchingAttr = 'cmta',
    gestaltHighLevelMatching = 0,
    gestaltColorMatchingLibLoaded = 1
};
```

Constants

`gestaltColorMatchingAttr`
: The selector for obtaining version information. Use when calling the `Gestalt` function to check for particular ColorSync Manager features.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltHighLevelMatching`
: This constant is provided for backward compatibility only. Bit 0 of the `Gestalt` response value is always set if ColorSync is present.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltColorMatchingLibLoaded`
: This constant is provided for backward compatibility only. Bit 1 of the `Gestalt` response value is always set on a Power Macintosh machine if ColorSync is present. It is always cleared on a 68K machine if ColorSync is present.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

Discussion
These constants were added to ColorSync version 2.0 to aid in the transition from 68K to PowerPC systems. They are not recommended for new applications and are not guaranteed to be carried forward in future versions of ColorSync. However, they are still supported as of version 2.5 for backward compatibility. If you call the `Gestalt` function passing the selector `gestaltColorMatchingAttr`, you can test the bit fields of

the returned value with the `gestaltColorMatchingLibLoaded` constant to determine if the ColorSync Manager shared libraries are loaded, or with the `gestaltHighLevelMatching` constant to determine if the ColorSync QuickDraw-specific functions are present.

ColorSync Manager Version Selectors

Specify version information for the ColorSync Manager.

```
enum {
    gestaltColorMatchingVersion = 'cmtc',
    gestaltColorSync10 = 0x0100,
    gestaltColorSync11 = 0x0110,
    gestaltColorSync104 = 0x0104,
    gestaltColorSync105 = 0x0105,
    gestaltColorSync20 = 0x0200,
    gestaltColorSync21 = 0x0210,
    gestaltColorSync211 = 0x0211,
    gestaltColorSync212 = 0x0212,
    gestaltColorSync213 = 0x0213,
    gestaltColorSync25 = 0x0250,
    gestaltColorSync26 = 0x0260,
    gestaltColorSync261 = 0x0261,
    gestaltColorSync30 = 0x0300
};
```

Constants

`gestaltColorMatchingVersion`

The selector for obtaining version information. Use when calling the `Gestalt` function to determine whether the ColorSync Manager is available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync10`

A `Gestalt` response value of `gestaltColorSync10` indicates version 1.0 of the ColorSync Manager is present. This version supports general purpose color matching only and does not provide QuickDraw-specific matching functions.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync11`

A `Gestalt` response value of `gestaltColorSync11` indicates version 1.0.3 of the ColorSync Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync104`

A `Gestalt` response value of `gestaltColorSync104` indicates version 1.4 of the ColorSync Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync105`
A `Gestalt` response value of `gestaltColorSync105` indicates version 1.5 of the ColorSync Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync20`
A `Gestalt` response value of `gestaltColorSync20` indicates version 2.0 of the ColorSync Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync21`
A `Gestalt` response value of `gestaltColorSync21` indicates version 2.1 of the ColorSync Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync211`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync212`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync213`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync25`
A `Gestalt` response value of `gestaltColorSync25` indicates version 2.5 of the ColorSync Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync26`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync261`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltColorSync30`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
These constants were added to ColorSync version 2.0 to aid in the transition from 68K to PowerPC systems. They are not recommended for new applications and are not guaranteed to be carried forward in future versions of ColorSync. However, they are still supported as of version 2.5 for backward compatibility. If you call the `Gestalt` function passing the selector `gestaltColorMatchingAttr`, you can test the bit fields of

the returned value with the `gestaltColorMatchingLibLoaded` constant to determine if the ColorSync Manager shared libraries are loaded, or with the `gestaltHighLevelMatching` constant to determine if the ColorSync QuickDraw-specific functions are present.

Communications Toolbox Version Selector

Specifies version information for the Communications Toolbox.

```
enum {
    gestaltCTBVersion = 'ctbv'
};
```

Constants

`gestaltCTBVersion`

The version number of the Communications Toolbox (in the low-order word of the return value).

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Communication Resource Manager Attribute Selectors

Specify version and feature availability information for the Communications Resource Manager.

```
enum {
    gestaltCRMAttr = 'crm ',
    gestaltCRMPresent = 0,
    gestaltCRMPersistentFix = 1,
    gestaltCRMToolRsrcCalls = 2
};
```

Component Manager Version Selectors

Specify version information for the Coomponent Manager.

```
enum {
    gestaltComponentMgr = 'cpnt',
    gestaltComponentPlatform = 'copl'
};
```

Constants

`gestaltComponentMgr`

The `Gestalt` selector you pass to determine what version of the Component Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion

To determine the version of the current Component Manager, your application should pass the selector `gestaltComponentMgr` to the `Gestalt` function.

Computer Model Selectors

Specify computer models.

```
enum {
    gestaltMachineType = 'mach',
    gestaltClassic = 1,
    gestaltMacXL = 2,
    gestaltMac512KE = 3,
    gestaltMacPlus = 4,
    gestaltMacSE = 5,
    gestaltMacII = 6,
    gestaltMacIIx = 7,
    gestaltMacIIcx = 8,
    gestaltMacSE030 = 9,
    gestaltPortable = 10,
    gestaltMacIIci = 11,
    gestaltPowerMac8100_120 = 12,
    gestaltMacIIfx = 13,
    gestaltMacClassic = 17,
    gestaltMacIIsi = 18,
    gestaltMacLC = 19,
    gestaltMacQuadra900 = 20,
    gestaltPowerBook170 = 21,
    gestaltMacQuadra700 = 22,
    gestaltClassicII = 23,
    gestaltPowerBook100 = 24,
    gestaltPowerBook140 = 25,
    gestaltMacQuadra950 = 26,
    gestaltMacLCIII = 27,
    gestaltPerforma450 = gestaltMacLCIII,
    gestaltPowerBookDuo210 = 29,
    gestaltMacCentris650 = 30,
    gestaltPowerBookDuo230 = 32,
    gestaltPowerBook180 = 33,
    gestaltPowerBook160 = 34,
    gestaltMacQuadra800 = 35,
    gestaltMacQuadra650 = 36,
    gestaltMacLCII = 37,
    gestaltPowerBookDuo250 = 38,
    gestaltAWS9150_80 = 39,
    gestaltPowerMac8100_110 = 40,
    gestaltAWS8150_110 = gestaltPowerMac8100_110,
    gestaltPowerMac5200 = 41,
    gestaltPowerMac5260 = gestaltPowerMac5200,
    gestaltPerforma5300 = gestaltPowerMac5200,
    gestaltPowerMac6200 = 42,
    gestaltPerforma6300 = gestaltPowerMac6200,
    gestaltMacIIvi = 44,
    gestaltMacIIvm = 45,
    gestaltPerforma600 = gestaltMacIIvm,
    gestaltPowerMac7100_80 = 47,
    gestaltMacIIvx = 48,
    gestaltMacColorClassic = 49,
    gestaltPerforma250 = gestaltMacColorClassic,
    gestaltPowerBook165c = 50,
    gestaltMacCentris610 = 52,
    gestaltMacQuadra610 = 53,
    gestaltPowerBook145 = 54,
    gestaltPowerMac8100_100 = 55,
    gestaltMacLC520 = 56,
    gestaltAWS9150_120 = 57,
```

```
    gestaltPowerMac6400 = 58,
    gestaltPerforma6400 = gestaltPowerMac6400,
    gestaltPerforma6360 = gestaltPerforma6400,
    gestaltMacCentris660AV = 60,
    gestaltMacQuadra660AV = gestaltMacCentris660AV,
    gestaltPerforma46x = 62,
    gestaltPowerMac8100_80 = 65,
    gestaltAWS8150_80 = gestaltPowerMac8100_80,
    gestaltPowerMac9500 = 67,
    gestaltPowerMac9600 = gestaltPowerMac9500,
    gestaltPowerMac7500 = 68,
    gestaltPowerMac7600 = gestaltPowerMac7500,
    gestaltPowerMac8500 = 69,
    gestaltPowerMac8600 = gestaltPowerMac8500,
    gestaltAWS8550 = gestaltPowerMac7500,
    gestaltPowerBook180c = 71,
    gestaltPowerBook520 = 72,
    gestaltPowerBook520c = gestaltPowerBook520,
    gestaltPowerBook540 = gestaltPowerBook520,
    gestaltPowerBook540c = gestaltPowerBook520,
    gestaltPowerMac5400 = 74,
    gestaltPowerMac6100_60 = 75,
    gestaltAWS6150_60 = gestaltPowerMac6100_60,
    gestaltPowerBookDuo270c = 77,
    gestaltMacQuadra840AV = 78,
    gestaltPerforma550 = 80,
    gestaltPowerBook165 = 84,
    gestaltPowerBook190 = 85,
    gestaltMacTV = 88,
    gestaltMacLC475 = 89,
    gestaltPerforma47x = gestaltMacLC475,
    gestaltMacLC575 = 92,
    gestaltMacQuadra605 = 94,
    gestaltMacQuadra630 = 98,
    gestaltMacLC580 = 99,
    gestaltPerforma580 = gestaltMacLC580,
    gestaltPowerMac6100_66 = 100,
    gestaltAWS6150_66 = gestaltPowerMac6100_66,
    gestaltPowerBookDuo280 = 102,
    gestaltPowerBookDuo280c = 103,
    gestaltPowerMacLC475 = 104,
    gestaltPowerMacPerforma47x = gestaltPowerMacLC475,
    gestaltPowerMacLC575 = 105,
    gestaltPowerMacPerforma57x = gestaltPowerMacLC575,
    gestaltPowerMacQuadra630 = 106,
    gestaltPowerMacLC630 = gestaltPowerMacQuadra630,
    gestaltPowerMacPerforma63x = gestaltPowerMacQuadra630,
    gestaltPowerMac7200 = 108,
    gestaltPowerMac7300 = 109,
    gestaltPowerMac7100_66 = 112,
    gestaltPowerBook150 = 115,
    gestaltPowerMacQuadra700 = 116,
    gestaltPowerMacQuadra900 = 117,
    gestaltPowerMacQuadra950 = 118,
    gestaltPowerMacCentris610 = 119,
    gestaltPowerMacCentris650 = 120,
    gestaltPowerMacQuadra610 = 121,
    gestaltPowerMacQuadra650 = 122,
```

```
    gestaltPowerMacQuadra800 = 123,
    gestaltPowerBookDuo2300 = 124,
    gestaltPowerBook500PPCUpgrade = 126,
    gestaltPowerBook5300 = 128,
    gestaltPowerBook1400 = 310,
    gestaltPowerBook3400 = 306,
    gestaltPowerBook2400 = 307,
    gestaltPowerBookG3Series = 312,
    gestaltPowerBookG3 = 313,
    gestaltPowerBookG3Series2 = 314,
    gestaltPowerMacNewWorld = 406,
    gestaltPowerMacG3 = 510,
    gestaltPowerMac5500 = 512,
    gestalt20thAnniversary = gestaltPowerMac5500,
    gestaltPowerMac6500 = 513,
    gestaltPowerMac4400_160 = 514,
    gestaltPowerMac4400 = 515,
    gestaltMacOSCompatibility = 1206
};
```

Discussion
To obtain a string containing the machine's name, you can pass the returned value to the `GetIndString` procedure as an index into the resource of type `'STR#'` in the System file having the resource ID defined by the constant `kMachineNameStrID`.

Computer Name Selector

Specifes user-visiblity information for the computer name.

```
enum {
    gestaltUserVisibleMachineName = 'mnam'
};
```

Connection Manager Attribute Selectors

Specify feature availability information for the Connection Manager.

```
enum {
    gestaltConnMgrAttr = 'conn',
    gestaltConnMgrPresent = 0,
    gestaltConnMgrCMSearchFix = 1,
    gestaltConnMgrErrorString = 2,
    gestaltConnMgrMultiAsyncIO = 3
};
```

Constants

`gestaltConnMgrAttr`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltConnMgrPresent`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltConnMgrCMSearchFix`
The `gestaltConnMgrCMSearchFix` bit flag indicates that the fix is present that allows the `CMAddSearch` function to work over the `mAttn` channel.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltConnMgrErrorString`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltConnMgrMultiAsyncIO`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Control Manager Attribute Selectors

Specify feature availability information for the Control Manager.

```
enum {
    gestaltControlMgrAttr = 'cntl',
    gestaltControlMgrPresent = (1L << 0),
    gestaltControlMgrPresentBit = 0,
    gestaltControlMsgPresentMask = (1L << gestaltControlMgrPresentBit)
};
```

Constants

`gestaltControlMgrAttr`
The `Gestalt` selector passed to determine what features of the Control Manager are present. This selector is available with Mac OS 8.5 and later. The `Gestalt` function produces a 32-bit value whose bits you should test to determine what Control Manager functionality is available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltControlMgrPresent`
If the bit specified by this mask is set, the Control Manager functionality for Appearance Manager 1.1 is available. This bit is set for Mac OS 8.5 and later.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltControlMgrPresentBit`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltControlMsgPresentMask`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion

Before calling any functions dependent upon the Control Manager, your application should pass the selector `gestaltControlMgrAttr` to the `Gestalt` function to determine which Control Manager functions are available.

Control Manager Version Selector

Specifies version information for the Control Manager.

```
enum {
    gestaltControlMgrVersion = 'cmvr'
};
```

Constants

`gestaltControlMgrVersion`
Available in Mac OS X v10.1 and later.
Declared in `Gestalt.h`.

Control Strip Attribute Selectors

Specify feature availability for the Control Strip.

```
enum {
    gestaltControlStripAttr = 'sdev',
    gestaltControlStripExists = 0,
    gestaltControlStripVersionFixed = 1,
    gestaltControlStripUserFont = 2,
    gestaltControlStripUserHotKey = 3
};
```

Control Strip Version Selector

Specifies version information for the Control Strip.

```
enum {
    gestaltControlStripVersion = 'csvr'
};
```

Constants

`gestaltControlStripVersion`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

CPU Selectors for Apollo

Specify version information for Apollo CPUs.

```
enum {
    gestaltCPUApollo = 0x0111,
    gestaltCPU750FX = 0x0120
};
```

CPU Selectors for Intel and Pentium

Specify version information for Intel and Pentium CPUs.

```
enum {
    gestaltCPU486 = 'i486',
    gestaltCPUPentium = 'i586',
    gestaltCPUPentiumPro = 'i5pr',
    gestaltCPUPentiumII = 'i5ii',
    gestaltCPUX86 = 'ixxx'
};
```

Data Access Manager Attribute Selectors

Specify feature availability information for the Data Access Manager.

```
enum {
    gestaltDBAccessMgrAttr = 'dbac',
    gestaltDBAccessMgrPresent = 0
};
```

Desktop Pictures Attribute Selectors

Specify feature availability information for Desktop Pictures.

```
enum {
    gestaltDesktopPicturesAttr = 'dkpx',
    gestaltDesktopPicturesInstalled = 0,
    gestaltDesktopPicturesDisplayed = 1
};
```

Desktop Printing Attribute Selector

Specify feature availablity information for all desktop printer.

```
enum {
    gestaltDTPInfo = 'dtpx'
};
```

Desktop Printing Driver Attribute Selectors

Specify feature availability for third-party desktop printing drivers.

```
enum {
    gestaltDTPFeatures = 'dtpf',
    kDTPThirdPartySupported = 0x00000004
};
```

Dialog Manager Attribute Selectors

Specify feature availability for the Dialog Manager.

```
enum {
    gestaltDITLExtAttr = 'ditl',
    gestaltDITLExtPresent = 0,
    gestaltDITLExtSupportsIctb = 1
};
```

Constants

`gestaltDITLExtAttr`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltDITLExtPresent`
If this flag bit is `TRUE`, then the Dialog Manager extensions included in System 7 are available.
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltDITLExtSupportsIctb`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

Dictionary Manager Attribute Selectors

Specify feature availability information for the Dictionary Manager.

```
enum {
    gestaltDictionaryMgrAttr = 'dict',
    gestaltDictionaryMgrPresent = 0
};
```

Dialog Manager Selectors for Mac OS 8.5

Specify version and feature availability information for the Dialog Manager in Mac OS 8.5.

```
enum {
    gestaltDialogMgrAttr = 'dlog',
    gestaltDialogMgrPresent = (1L << 0),
    gestaltDialogMgrPresentBit = 0,
    gestaltDialogMgrHasAquaAlertBit = 2,
    gestaltDialogMgrPresentMask = (1L << gestaltDialogMgrPresentBit),
    gestaltDialogMgrHasAquaAlertMask = (1L << gestaltDialogMgrHasAquaAlertBit),
    gestaltDialogMsgPresentMask = gestaltDialogMgrPresentMask
};
```

Constants

`gestaltDialogMgrAttr`
The `Gestalt` selector passed to determine what features of the Dialog Manager are present. This selector is available with Mac OS 8.5 and later. Passing `gestaltDialogManagerAttr` produces a 32-bit value whose bits you should test to determine what Dialog Manager functionality is available.
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltDialogMgrPresent`
If the bit specified by this mask is set, the Dialog Manager functionality for Appearance Manager 1.1 is available. This bit is set for Mac OS 8.5 and later.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDialogMgrPresentBit`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDialogMgrHasAquaAlertBit`
Available in Mac OS X v10.1 and later.

Declared in `Gestalt.h`.

`gestaltDialogMgrPresentMask`
Available in Mac OS X v10.1 and later.

Declared in `Gestalt.h`.

`gestaltDialogMgrHasAquaAlertMask`
Available in Mac OS X v10.1 and later.

Declared in `Gestalt.h`.

`gestaltDialogMsgPresentMask`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any Dialog Manager functions, your application should pass the selector `gestaltDialogManagerAttr` to the `Gestalt` function to determine which Dialog Manager functions are available.

Digital Signature Version Selector

Specifies version information for digital signatures.

```
enum {
    gestaltDigitalSignatureVersion = 'dsig'
};
```

Direct IO Attribute Selector

Specifies availability of direct input/ouput support by the file system.

```
enum {
    gestaltFSSupportsDirectIO = 11
};
```

Disk Cache Size Selector

Specifies size information for the disk cache buffer.

```
enum {
    gestaltDiskCacheSize = 'dcsz'
};
```

Constants

`gestaltDiskCacheSize`

A selector that you pass to the `Gestalt` function. If the function returns `noErr`, the `response` parameter contains the size of the disk cache's buffer. See the Gestalt Manager Reference for more information on the `Gestalt` function.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Display Manager Attribute Selectors

Specify feature availability for the Display Manager.

```
enum {
    gestaltDisplayMgrAttr = 'dply',
    gestaltDisplayMgrPresent = 0,
    gestaltDisplayMgrCanSwitchMirrored = 2,
    gestaltDisplayMgrSetDepthNotifies = 3,
    gestaltDisplayMgrCanConfirm = 4,
    gestaltDisplayMgrColorSyncAware = 5,
    gestaltDisplayMgrGeneratesProfiles = 6,
    gestaltDisplayMgrSleepNotifies = 7
};
```

Constants

`gestaltDisplayMgrAttr`

The `Gestalt` selector you pass to determine which Display Manager attributes are present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDisplayMgrPresent`

If `true`, the Display Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDisplayMgrCanSwitchMirrored`

If `true`, the Display Manager can switch modes on mirrored displays.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDisplayMgrSetDepthNotifies`

If `true`, and you have registered for notification and you will be notified of depth mode changes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDisplayMgrCanConfirm`
Not yet supported. Most commonly comes up for display modes that are not marked `kModeSafe`. There is currently no system support for trying an unsafe mode and then restoring if the user does not confirm. When this is supported, this bit will be set.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDisplayMgrColorSyncAware`
If `true`, Display Manager supports profiles for displays.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDisplayMgrGeneratesProfiles`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDisplayMgrSleepNotifies`
Available in Mac OS X v10.2 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any function dependent upon the Display Manager, your application should pass the selector `gestaltDisplayMgrAttr` to the `Gestalt` function to determine the Display Manager attributes that are present.

Display Manager Version Selector

Specifies version information for the Display Manager.

```
enum {
    gestaltDisplayMgrVers = 'dplv'
};
```

Constants
`gestaltDisplayMgrVers`
The `Gestalt` selector you pass to determine what version of the Display Manager is present. For example, a Gestalt result may be 0x00020500, which means that the Display Manager version 2.5 is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
To determine the version of the current Display Manager, your application should pass the selector `gestaltDisplayMgrVers` to the `Gestalt` function.

Drag Manager Attribute Selectors

Specify feature availability information for the Drag Manager.

```
enum {
    gestaltDragMgrAttr = 'drag',
    gestaltDragMgrPresent = 0,
    gestaltDragMgrFloatingWind = 1,
    gestaltPPCDragLibPresent = 2,
    gestaltDragMgrHasImageSupport = 3,
    gestaltCanStartDragInFloatWindow = 4,
    gestaltSetDragImageUpdates = 5
};
```

Constants

`gestaltDragMgrAttr`
The Gestalt selector passed to determine what features of the Drag Manager are present. Passing the `gestaltDragMgrAttr` constant produces a 32-bit value whose bits you should test to determine what Drag Manager functionality is available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDragMgrPresent`
If the bit specified by this mask is set, the Drag Manager functions are available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDragMgrFloatingWind`
If the bit specified by this mask is set, the Drag Manager floating window support functions are available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPPCDragLibPresent`
If the bit specified by this mask is set, the Drag Manager PPC Drag Library functions are available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDragMgrHasImageSupport`
If the bit specified by this mask is set, the Drag Manager image support functions are available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCanStartDragInFloatWindow`
If the bit specified by this mask is set, the Drag Manager can start a drag in a floating window.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSetDragImageUpdates`
Available in Mac OS X v10.1 and later.

Declared in `Gestalt.h`.

Draw Sprocket Version Selectors

Specifies version information for Draw Sprocket.

```
enum {
    gestaltDrawSprocketVersion = 'dspv'
};
```

Easy Access Selectors

Specify version and feature availability information for Easy Access.

```
enum {
    gestaltEasyAccessAttr = 'easy',
    gestaltEasyAccessOff = 0,
    gestaltEasyAccessOn = 1,
    gestaltEasyAccessSticky = 2,
    gestaltEasyAccessLocked = 3
};
```

Edition Manager Attribute Selectors

Specify feature availability for the Edition Manager.

```
enum {
    gestaltEditionMgrAttr = 'edtn',
    gestaltEditionMgrPresent = 0,
    gestaltEditionMgrTranslationAware = 1
};
```

Extension Table Version Selector

Specifies version information for the extension table.

```
enum {
    gestaltExtensionTableVersion = 'etbl'
};
```

File Mapping Attribute Selectors

Specify feature availability for file mapping.

```
enum {
    gestaltFileMappingAttr = 'flmp',
    gestaltFileMappingPresent = 0,
    gestaltFileMappingMultipleFilesFix = 1
};
```

File System Attribute Selectors

Specify feature availability for the file system.

```
enum {
    gestaltFSAttr = 'fs  ',
    gestaltFullExtFSDispatching = 0,
    gestaltHasFSSpecCalls = 1,
    gestaltHasFileSystemManager = 2,
    gestaltFSMDoesDynamicLoad = 3,
    gestaltFSSupports4GBVols = 4,
    gestaltFSSupports2TBVols = 5,
    gestaltHasExtendedDiskInit = 6,
    gestaltDTMgrSupportsFSM = 7,
    gestaltFSNoMFSVols = 8,
    gestaltFSSupportsHFSPlusVols = 9,
    gestaltFSIncompatibleDFA82 = 10
};
```

Constants

`gestaltFSAttr`

A selector you pass to the `Gestalt` function. If the Gestalt function returns `noErr`, the `response` parameter contains a 32-bit value specifying the features of the file system.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFullExtFSDispatching`

If this bit is set in the `response` parameter, all of the functions selected through the `_HFSDispatch` trap are available to external file systems. If this bit is clear, the File Manager checks the selector passed to `_HFSDispatch` and ensures that it is valid; if the selector is invalid, the result code `paramErr` is returned to the caller. If this bit is set, no such validity checking is performed. See the *Guide to the File System Manager* for more information on external file systems.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasFSSpecCalls`

If this bit is set in the `response` parameter, the operating environment provides the file system specification (`FSSpec`) versions of the basic file-manipulation functions, as well as the `FSMakeFSSpec` function.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasFileSystemManager`

If this bit is set in the `response` parameter, the File System Manager is present. See the *Guide to the File System Manager* for more information about the File System Manager.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFSMDoesDynamicLoad`

If this bit is set in the `response` parameter, the File System Manager supports dynamic loading of external file system code resources.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFSSupports4GBVols`

If this bit is set in the `response` parameter, the file system supports 4 gigabyte volumes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFSSupports2TBVols`
If this bit is set in the `response` parameter, the file system supports 2 terabyte volumes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasExtendedDiskInit`
If this bit is set in the `response` parameter, the extended Disk Initialization Package functions are present. These are the `DIXFormat`, `DIXZero`, or `DIReformat` functions. See the *Guide to the File System Manager* for more information about the Disk Initialization Package interfaces.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDTMgrSupportsFSM`
If this bit is set in the `response` parameter, the desktop database supports File System Manager-based foreign file systems.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFSNoMFSVols`
If this bit is set in the `response` parameter, the file system does not support MFS volumes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFSSupportsHFSPlusVols`
If this bit is set in the `response` parameter, the file system supports HFS Plus volumes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFSIncompatibleDFA82`
If this bit is set in the `response` parameter, VCB and FCB structures are changed; DFA 8.2 is incompatible.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

File System Attribute Selectors for Mac OS 9

Specify feature availability for the file system for features introduced in Mac OS 9.

```
enum {
    gestaltHasHFSPlusAPIs = 12,
    gestaltMustUseFCBAccessors = 13,
    gestaltFSUsesPOSIXPathsForConversion = 14,
    gestaltFSSupportsExclusiveLocks = 15,
    gestaltFSSupportsHardLinkDetection = 16
};
```

Constants

`gestaltHasHFSPlusAPIs`
If this bit is set in the `response` parameter, the File Manager supports the HFS Plus APIs. Individual file systems may or may not implement the HFS Plus APIs. However, if this bit is set, the File Manager will emulate the HFS Plus APIs for file systems that do not implement them. Call the functions `PBHGetVolParmsSync` or `PBHGetVolParmsAsync` to determine whether the HFS Plus APIs are directly supported on a given volume.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMustUseFCBAccessors`
If this bit is set in the `response` parameter, the File Manager no longer supports the low memory globals `FCBSPtr` and `FSFCBLen`. All access to file or fork control blocks must use the File System Manager utility functions instead.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFSUsesPOSIXPathsForConversion`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFSSupportsExclusiveLocks`
Available in Mac OS X v10.2 and later.

Declared in `Gestalt.h`.

`gestaltFSSupportsHardLinkDetection`
Available in Mac OS X v10.2 and later.

Declared in `Gestalt.h`.

Discussion

Use these constants with the gestalt selector `gestaltFSAttr`, described in "File System Attribute Selectors" (page 48).

File System Manager Version Selector

Specifies version information for the File System Manager.

```
enum {
    gestaltFSMVersion = 'fsm '
};
```

Constants

`gestaltFSMVersion`

Pass this selector to the `Gestalt` function to determine the version of the HFS External File Systems Manager (FSM).

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

File System Transport Manager Attribute Selectors

Specify feature availability for the File System Transport Manager

```
enum {
    gestaltFXfrMgrAttr = 'fxfr',
    gestaltFXfrMgrPresent = 0,
    gestaltFXfrMgrMultiFile = 1,
    gestaltFXfrMgrErrorString = 2,
    gestaltFXfrMgrAsync = 3
};
```

Constants

`gestaltFXfrMgrAttr`

The selector you pass to the `Gestalt` function to determine the File Transfer Manager attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Find By Content State Selectors

Specify state information for Find By Content.

```
enum {
    gestaltFBCIndexingState = 'fbci',
    gestaltFBCindexingSafe = 0,
    gestaltFBCindexingCritical = 1
};
```

Find By Content Version Selectors

Specify version information for Find By Content.

```
enum {
    gestaltFBCVersion = 'fbcv',
    gestaltFBCCurrentVersion = 0x0011,
    gestaltOSXFBCCurrentVersion = 0x0100
};
```

Find Folder Redirection Attribute Selector

Specifies feature availability information for Find Folder.

```
enum {
    gestaltFindFolderRedirectionAttr = 'fole'
};
```

Finder Attribute Selectors

Specify feature availability for the Finder.

```
enum {
    gestaltFinderAttr = 'fndr',
    gestaltFinderDropEvent = 0,
    gestaltFinderMagicPlacement = 1,
    gestaltFinderCallsAEProcess = 2,
    gestaltOSLCompliantFinder = 3,
    gestaltFinderSupports4GBVolumes = 4,
    gestaltFinderHasClippings = 6,
    gestaltFinderFullDragManagerSupport = 7,
    gestaltFinderFloppyRootComments = 8,
    gestaltFinderLargeAndNotSavedFlavorsOK = 9,
    gestaltFinderUsesExtensibleFolderManager = 10,
    gestaltFinderUnderstandsRedirectedDesktopFolder = 11
};
```

Floppy Driver Attribute Selectors

Specify feature availability information for the floppy disk drive.

```
enum {
    gestaltFloppyAttr = 'flpy',
    gestaltFloppyIsMFMOnly = 0,
    gestaltFloppyIsManualEject = 1,
    gestaltFloppyUsesDiskInPlace = 2
};
```

Font Manager Attribute Selectors

Specify feature availability information for the Font Manager.

```
enum {
    gestaltFontMgrAttr = 'font',
    gestaltOutlineFonts = 0
};
```

Constants

`gestaltFontMgrAttr`

The `Gestalt` selector you pass to determine which Font Manager attributes are present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltOutlineFonts`

If `true`, outline fonts are supported.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion

Before calling any function dependent upon the Font Manager, your application should pass the selector `gestaltDisplayMgrAttr` to the `Gestalt` function to determine the Font Manager attributes that are present.

Folder Manager Attribute Selectors

Specify feature availability information for the Folder Manager.

```
enum {
    gestaltFindFolderAttr = 'fold',
    gestaltFindFolderPresent = 0,
    gestaltFolderDescSupport = 1,
    gestaltFolderMgrFollowsAliasesWhenResolving = 2,
    gestaltFolderMgrSupportsExtendedCalls = 3,
    gestaltFolderMgrSupportsDomains = 4,
    gestaltFolderMgrSupportsFSCalls = 5
};
```

Constants

`gestaltFindFolderAttr`

The selector you pass to the `Gestalt` function to determine the `FindFolder` function attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFindFolderPresent`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFolderDescSupport`

If this bit is set, the extended Folder Manager functionality supporting folder descriptors and routings is available. This bit is set for versions of the Mac OS starting with Mac OS 8.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFolderMgrFollowsAliasesWhenResolving`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFolderMgrSupportsExtendedCalls`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFolderMgrSupportsDomains`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFolderMgrSupportsFSCalls`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any Folder Manager functions, your application should pass the selector `gestaltFindFolderAttr` to the `Gestalt` function to determine which Folder Manager functions are available.

FPU Type Selectors

Specify version and availability information for the type of floating-point unit installed.

```
enum {
    gestaltFPUType = 'fpu ',
    gestaltNoFPU = 0,
    gestalt68881 = 1,
    gestalt68882 = 2,
    gestalt68040FPU = 3
};
```

Constants
`gestaltFPUType`
A constant that represents the type of floating-point unit currently installed, if any.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltNoFPU`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68881`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68882`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68040FPU`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Gestalt Manager Version Selectors

Specify Gestalt Manager version information.

```
enum {
    gestaltVersion = 'vers',
    gestaltValueImplementedVers = 5
};
```

Constants

`gestaltVersion`

The selector you pass to the function `Gestalt` (page 13) to determine the version of the Gestalt Manager. The function passes back the version in the low-order word of the response.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltValueImplementedVers`

The first version of the Gestalt Manager that implements this selector.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Hardware Attribute Attribute Selectors

Specify feature availability information for hardware.

```
enum {
    gestaltHardwareAttr = 'hdwr',
    gestaltHasVIA1 = 0,
    gestaltHasVIA2 = 1,
    gestaltHasASC = 3,
    gestaltHasSCC = 4,
    gestaltHasSCSI = 7,
    gestaltHasSoftPowerOff = 19,
    gestaltHasSCSI961 = 21,
    gestaltHasSCSI962 = 22,
    gestaltHasUniversalROM = 24,
    gestaltHasEnhancedLtalk = 30
};
```

Constants

`gestaltHardwareAttr`

The selector you pass to the Gestalt function to determine low-level hardware configuration attributes.

Never infer the existence of certain hardware or software features from the responses that `Gestalt` returns when you pass it this selector.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasVIA1`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasVIA2`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasASC`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasSCC`
The `gestaltHasSCC` bit is normally returned as 0 on the Macintosh IIfx and Macintosh Quadra 900 computers, which have intelligent I/O processors that isolate the hardware and make direct access to the SCC impossible. However, if the user has used the Compatibility Switch control panel to enable compatibility mode, `gestaltHasSCC` is set.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasSCSI`
The `gestaltHasSCSI` bit means the machine is equipped with a SCSI implementation based on the 53C80 chip, which was introduced in the Macintosh Plus. This bit is 0 on computers with a different SCSI implementation.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasSoftPowerOff`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasSCSI961`
This bit is set if the machine has a SCSI implementation based on the 53C96 chip installed on an internal bus.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasSCSI962`
This bit is set if the machine has a SCSI implementation based on the 53C96 chip installed on an external bus.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasUniversalROM`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasEnhancedLtalk`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Hardware Icon Selector

Specifies icon family resource ID information for the computer hardware.

```
enum {
    gestaltMachineIcon = 'micn'
};
```

Constants

`gestaltMachineIcon`
The selector you pass to the `Gestalt` function to determine the icon family resource ID for the current type of Macintosh.

Never infer the existence of certain hardware or software features from the responses that `Gestalt` returns when you pass it this selector.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Hardware Vendor Selectors

Specify hardware vendor information.

```
enum {
    gestaltHardwareVendorCode = 'hrad',
    gestaltHardwareVendorApple = 'Appl'
};
```

Help Manager Attribute Selectors

Specify feature availability for the Apple Help Manager.

```
enum {
    gestaltHelpMgrAttr = 'help',
    gestaltHelpMgrPresent = 0,
    gestaltHelpMgrExtensions = 1,
    gestaltAppleGuideIsDebug = 30,
    gestaltAppleGuidePresent = 31
};
```

Constants

`gestaltHelpMgrAttr`
The selector you pass to the `Gestalt` function to determine the Help Manager attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHelpMgrPresent`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHelpMgrExtensions`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAppleGuideIsDebug`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAppleGuidePresent`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Icon Services Attribute Selectors

Specify feature availability for Icon Services.

```
enum {
    gestaltIconUtilitiesAttr = 'icon',
    gestaltIconUtilitiesPresent = 0,
    gestaltIconUtilitiesHas48PixelIcons = 1,
    gestaltIconUtilitiesHas32BitIcons = 2,
    gestaltIconUtilitiesHas8BitDeepMasks = 3,
    gestaltIconUtilitiesHasIconServices = 4
};
```

Constants

`gestaltIconUtilitiesAttr`
The `Gestalt` selector passed to determine which features of Icon Services are present. The `Gestalt` function produces a 32-bit value whose bits you should test to determine which Icon Services features are available.

Note: available in System 7.0, despite gestalt.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltIconUtilitiesPresent`
True if icon utilities are present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltIconUtilitiesHas48PixelIcons`
True if 48x48 icons are supported by IconUtilities.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltIconUtilitiesHas32BitIcons`
True if 32-bit deep icons are supported.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltIconUtilitiesHas8BitDeepMasks`
True if 8-bit deep masks are supported.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltIconUtilitiesHasIconServices`
True if IconServices is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any Icon Services functions, your application should pass the selector `gestaltIconUtilitiesAttr` to the `Gestalt` function.

Image Compression Manager Version Selector

Specifies the version of the Image Compression Manager.

```
enum {
    gestaltCompressionMgr = 'icmp'
};
```

Constants
`gestaltCompressionMgr`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

Intel Architecture Selector

Specifies the Intel architecture.

```
enum {
    gestaltIntel = 10
};
```

Internal Display Location Selector

Specifies the slot number information for the internal display location.

```
enum {
    gestaltInternalDisplay = 'idsp'
};
```

Keyboard Selectors

Specify keyboard information.

```
enum {
    gestaltKeyboardType = 'kbd ',
    gestaltMacKbd = 1,
    gestaltMacAndPad = 2,
    gestaltMacPlusKbd = 3,
    gestaltExtADBKbd = 4,
    gestaltStdADBKbd = 5,
    gestaltPrtblADBKbd = 6,
    gestaltPrtblISOKbd = 7,
    gestaltStdISOADBKbd = 8,
    gestaltExtISOADBKbd = 9,
    gestaltADBKbdII = 10,
    gestaltADBISOKbdII = 11,
    gestaltPwrBookADBKbd = 12,
    gestaltPwrBookISOADBKbd = 13,
    gestaltAppleAdjustKeypad = 14,
    gestaltAppleAdjustADBKbd = 15,
    gestaltAppleAdjustISOKbd = 16,
    gestaltJapanAdjustADBKbd = 17,
    gestaltPwrBkExtISOKbd = 20,
    gestaltPwrBkExtJISKbd = 21,
    gestaltPwrBkExtADBKbd = 24,
    gestaltPS2Keyboard = 27,
    gestaltPwrBkSubDomKbd = 28,
    gestaltPwrBkSubISOKbd = 29,
    gestaltPwrBkSubJISKbd = 30,
    gestaltPwrBkEKDomKbd = 195,
    gestaltPwrBkEKISOKbd = 196,
    gestaltPwrBkEKJISKbd = 197,
    gestaltUSBCosmoANSIKbd = 198,
    gestaltUSBCosmoISOKbd = 199,
    gestaltUSBCosmoJISKbd = 200,
    gestaltPwrBk99JISKbd = 201,
    gestaltUSBAndyANSIKbd = 204,
    gestaltUSBAndyISOKbd = 205,
    gestaltUSBAndyJISKbd = 206
};
```

Constants

`gestaltKeyboardType`

The selector you pass to the `Gestalt` function to determine the type of the keyboard.

If the Apple Desktop Bus (ADB) is in use, there may be multiple keyboards or other ADB devices attached to the machine. The `gestaltKeyboardType` selector identifies only the type of the keyboard on which the last keystroke occurred.

You cannot use this selector to find out what ADB devices are connected. For that, you can use the Apple Desktop Bus Manager. Note that the ADB keyboard types described by `Gestalt` do not necessarily map directly to ADB device handler IDs.

Future support for the `gestaltKeyboardType` selector is not guaranteed. To determine the type of the keyboard last touched without using `Gestalt`, check the system global variable `KbdType`.

If the Gestalt Manager does not recognize the keyboard type, it returns an error.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Keyboard Selectors for Laptops

Specify laptop keyboard information.

```
enum {
    gestaltPortable2001ANSIKbd = 202,
    gestaltPortable2001ISOKbd = 203,
    gestaltPortable2001JISKbd = 207
};
```

Logical Page Size Selector

Specifies logical page size information.

```
enum {
    gestaltLogicalPageSize = 'pgsz'
};
```

Constants

`gestaltLogicalPageSize`

The logical page size. This value is defined only on machines with the MC68010, MC68020, MC68030, or MC68040 microprocessors. On a machine with the MC68000, the `Gestalt` function returns an error when called with this selector.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Logical RAM Size Selector

Specifies logical randon-access memory size information.

```
enum {
    gestaltLogicalRAMSize = 'lram'
};
```

Constants

`gestaltLogicalRAMSize`

The amount of logical memory available. This value is the same as that returned by `gestaltPhysicalRAMSize` when virtual memory is not installed. On some machines, however, this value might be less than the value returned by `gestaltPhysicalRAMSize` because some RAM may be used by the video display and the Operating System.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Low Memory Size Selector

Specifies information about the size of the low-memory area.

```
enum {
    gestaltLowMemorySize = 'lmem'
};
```

Constants

`gestaltLowMemorySize`

The size (in bytes) of the low-memory area. The low-memory area is used for vectors, global variables, and dispatch tables

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Machine Name String ID

Defines a machine name string ID.

```
enum {
    kMachineNameStrID = -16395
};
```

Mailer Version Selector

Specifies version information for the OCE standard mailer.

```
enum {
    gestaltSMPMailerVersion = 'malr'
};
```

Mailer Send LetterVersion Selector

Specifies version information for the OCE standard mailer's send letter.

```
enum {
    gestaltSMPSPSendLetterVersion = 'spsl'
};
```

Media Bay Selectors

Specify information about media bay availability.

```
enum {
    gestaltMediaBay = 'mbeh',
    gestaltMBLegacy = 0,
    gestaltMBSingleBay = 1,
    gestaltMBMultipleBays = 2
};
```

Memory Attribute Selectors

Specify feature availability information for memory.

```
enum {
    gestaltOSAttr = 'os  ',
    gestaltSysZoneGrowable = 0,
    gestaltLaunchCanReturn = 1,
    gestaltLaunchFullFileSpec = 2,
    gestaltLaunchControl = 3,
    gestaltTempMemSupport = 4,
    gestaltRealTempMemory = 5,
    gestaltTempMemTracked = 6,
    gestaltIPCSupport = 7,
    gestaltSysDebuggerSupport = 8,
    gestaltNativeProcessMgrBit = 19,
    gestaltAltivecRegistersSwappedCorrectlyBit = 20
};
```

Constants

`gestaltOSAttr`

The `Gestalt` selector you pass to determine general Operating System attributes, such as whether temporary memory handles are real handles. The low-order bits of the response parameter are interpreted as bit flags. A flag is set to 1 to indicate that the corresponding feature is available. Currently, the following bits are significant.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSysZoneGrowable`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltLaunchCanReturn`

If this bit is set, the `_Launch` trap macro can return to the caller. The `_Launch` trap macro in system software version 7.0 (and in earlier versions running MultiFinder) gives your application the option to continue running after it launches another application. In earlier versions of system software not running MultiFinder, the `_Launch` trap macro forces the launching application to quit.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltLaunchFullFileSpec`

If this bit is set, the `launchControlFlags` field supports control flags in addition to the `launchContinue` flag, and if the `_Launch` trap can process the `launchAppSpec`, `launchProcessSN`, `launchPreferredSize`, `launchMinimumSize`, `launchAvailableSize`, and `launchAppParameters` fields in the launch parameter block.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltLaunchControl`

If this bit is set, the Process Manager is available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTempMemSupport`

If `true`, there is temporary memory support.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltRealTempMemory`
If `true`, temporary memory handles are real.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTempMemTracked`
If `true`, temporary memory handles are tracked.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltIPCSupport`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSysDebuggerSupport`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltNativeProcessMgrBit`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAltivecRegistersSwappedCorrectlyBit`
Available in Mac OS X v10.2 and later.

Declared in `Gestalt.h`.

Memory Mapping Attribute Selectors

Specify feature availibility information for memory mapping.

```
enum {
    gestaltMemoryMapAttr = 'mmap',
    gestaltMemoryMapSparse = 0
};
```

Menu Manager Selectors in Mac OS 8.5

Specify version and feature availability information for the Menu Manager in Mac OS 8.5

```
enum {
    gestaltMenuMgrAttr = 'menu',
    gestaltMenuMgrPresent = (1L << 0),
    gestaltMenuMgrPresentBit = 0,
    gestaltMenuMgrAquaLayoutBit = 1,
    gestaltMenuMgrMultipleItemsWithCommandIDBit = 2,
    gestaltMenuMgrRetainsIconRefBit = 3,
    gestaltMenuMgrSendsMenuBoundsToDefProcBit = 4,
    gestaltMenuMgrMoreThanFiveMenusDeepBit = 5,
    gestaltMenuMgrPresentMask = (1L << gestaltMenuMgrPresentBit),
    gestaltMenuMgrAquaLayoutMask = (1L << gestaltMenuMgrAquaLayoutBit),
    gestaltMenuMgrMultipleItemsWithCommandIDMask = (1L <<
gestaltMenuMgrMultipleItemsWithCommandIDBit),
    gestaltMenuMgrRetainsIconRefMask = (1L << gestaltMenuMgrRetainsIconRefBit),
    gestaltMenuMgrSendsMenuBoundsToDefProcMask = (1L <<
gestaltMenuMgrSendsMenuBoundsToDefProcBit),
    gestaltMenuMgrMoreThanFiveMenusDeepMask = (1L <<
gestaltMenuMgrMoreThanFiveMenusDeepBit)
};
```

Constants

`gestaltMenuMgrAttr`
: The `Gestalt` selector passed to determine what features of the Menu Manager are present. This selector is available with Mac OS 8.5 and later. Passing `gestaltMenuMgrAttr` produces a 32-bit value whose bits you should test to determine what Menu Manager functionality is available.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltMenuMgrPresent`
: If the bit specified by this mask is set, the Menu Manager functionality for Appearance Manager 1.1 is available. This bit is set for Mac OS 8.5 and later.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltMenuMgrPresentBit`
: Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltMenuMgrAquaLayoutBit`
: Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltMenuMgrMultipleItemsWithCommandIDBit`
: Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltMenuMgrRetainsIconRefBit`
: Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltMenuMgrSendsMenuBoundsToDefProcBit`
: Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltMenuMgrMoreThanFiveMenusDeepBit`
Available in Mac OS X v10.2 and later.

Declared in `Gestalt.h`.

`gestaltMenuMgrPresentMask`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMenuMgrAquaLayoutMask`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMenuMgrMultipleItemsWithCommandIDMask`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMenuMgrRetainsIconRefMask`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMenuMgrSendsMenuBoundsToDefProcMask`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMenuMgrMoreThanFiveMenusDeepMask`
Available in Mac OS X v10.2 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any Menu Manager functions, your application should pass the selector `gestaltMenuMgrAttr` to the `Gestalt` function to determine which Menu Manager functions are available.

Message Manager Version Selector

Specify version information for the Message Manager.

```
enum {
    gestaltMessageMgrVersion = 'mess'
};
```

Miscellaneous Attribute Selectors

Specify feature availability information for miscellaneous pieces of the operating system or the hardware configuration.

```
enum {
    gestaltMiscAttr = 'misc',
    gestaltScrollingThrottle = 0,
    gestaltSquareMenuBar = 2
};
```

Constants

`gestaltMiscAttr`
The selector you pass to the `Gestalt` function to determine information about miscellaneous pieces of the Operating System or hardware configuration.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltScrollingThrottle`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSquareMenuBar`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Mixed Mode Manager Selectors

Specify version and feature availability information for the Mixed Mode Manager.

```
enum {
    gestaltMixedModeAttr = 'mixd',
    gestaltMixedModePowerPC = 0,
    gestaltPowerPCAware = 0,
    gestaltMixedModeCFM68K = 1,
    gestaltMixedModeCFM68KHasTrap = 2,
    gestaltMixedModeCFM68KHasState = 3
};
```

Constants

`gestaltMixedModeAttr`
The Gestalt selector you pass to determine what version of Mixed Mode Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMixedModePowerPC`
True if Mixed Mode supports PowerPC ABI calling conventions

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPowerPCAware`
Old name for `gestaltMixedModePowerPC`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMixedModeCFM68K`
: True if Mixed Mode supports CFM-68K calling conventions

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMixedModeCFM68KHasTrap`
: True if CFM-68K Mixed Mode implements `_MixedModeDispatch` (versions 1.0.1 and prior did not)

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMixedModeCFM68KHasState`
: True if CFM-68K Mixed Mode exports Save/RestoreMixedModeState

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any function dependent upon Mixed Mode Manager, your application should pass the selector `gestaltMixedModeAttr` to the `Gestalt` function to determine the Mixed Mode Manager attributes that are present.

Mixed Mode Manager Version Selector

Specifies version information for the Mixed Mode Manager.

```
enum {
    gestaltMixedModeVersion = 'mixd'
};
```

Constants
`gestaltMixedModeVersion`
: The selector you pass to the `Gestalt` function to determine the version of Mixed Mode Manager.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

MMU Type Selectors

Specify information about the type of MMU installed.

```
enum {
    gestaltMMUType = 'mmu ',
    gestaltNoMMU = 0,
    gestaltAMU = 1,
    gestalt68851 = 2,
    gestalt68030MMU = 3,
    gestalt68040MMU = 4,
    gestaltEMMU1 = 5
};
```

Constants

`gestaltMMUType`
The selector you pass to the `Gestalt` function to determine the type of MMU currently installed.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltNoMMU`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAMU`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68851`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68030MMU`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68040MMU`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltEMMU1`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Multiple Users State Selector

Specifies information about the multiple user state.

```
enum {
    gestaltMultipleUsersState = 'mfdr'
};
```

Name-Binding Protocol Attribute Selectors

Specify feature availiability information for the standard name-binding protocol.

```
enum {
    gestaltStdNBPAttr = 'nlup',
    gestaltStdNBPPresent = 0,
    gestaltStdNBPSupportsAutoPosition = 1
};
```

Constants

`gestaltStdNBPAttr`
The selector you pass to the `Gestalt` function to determine information about the StandardNBP (Name-Binding Protocol) function.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltStdNBPPresent`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltStdNBPSupportsAutoPosition`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Name Registry Version Selector

Specifies the version of the name registry.

```
enum {
    gestaltNameRegistryVersion = 'nreg'
};
```

Constants

`gestaltNameRegistryVersion`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Native CPU Selectors

Specify the native CPU type or family.

```
enum {
    gestaltNativeCPUtype = 'cput',
    gestaltNativeCPUfamily = 'cpuf',
    gestaltCPU68000 = 0,
    gestaltCPU68010 = 1,
    gestaltCPU68020 = 2,
    gestaltCPU68030 = 3,
    gestaltCPU68040 = 4,
    gestaltCPU601 = 0x0101,
    gestaltCPU603 = 0x0103,
    gestaltCPU604 = 0x0104,
    gestaltCPU603e = 0x0106,
    gestaltCPU603ev = 0x0107,
    gestaltCPU750 = 0x0108,
    gestaltCPU604e = 0x0109,
    gestaltCPU604ev = 0x010A,
    gestaltCPUG4 = 0x010C,
    gestaltCPUG47450 = 0x0110
};
```

Constants

`gestaltNativeCPUtype`
The selector you pass to the `Gestalt` function to determine the native CPU type.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltNativeCPUfamily`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU68000`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU68010`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU68020`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU68030`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU68040`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU601`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU603`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU604`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU603e`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU603ev`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU750`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU604e`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPU604ev`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPUG4`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCPUG47450`
Available in Mac OS X v10.2 and later.

Declared in `Gestalt.h`.

Discussion
The use of these selectors is no longer recommended. You can use the `gestaltSysArchitecture` (page 104) selector to determine whether your application is running on a PowerPC or Intel-based Macintosh. If you are trying to determine whether you can use a particular processor feature, you should check directly for that feature using a BSD library function such as `sysctl` or `sysctlbyname`. For more information, see *Mac OS X Man Pages*.

Notification Manager Attribute Selectors

Specify feature availability information for the Notification Manager.

```
enum {
    gestaltNotificationMgrAttr = 'nmgr',
    gestaltNotificationPresent = 0
};
```

Constants
`gestaltNotificationMgrAttr`
.The Gestalt selector which you pass to the `Gestalt` function to determine Notification Manager attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltNotificationPresent`
: True if the Notification Manager exists.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

NuBus Location Selector

Specifies information about the NuBus slot connector locations.

```
enum {
    gestaltNuBusConnectors = 'sltc'
};
```

Constants

`gestaltNuBusConnectors`
: A bitmap that describes the NuBus slot connector locations. On a Macintosh II, for example, the return value would have bits 9 through 14 set, indicating that 6 NuBus slots are present, at locations 9 through 14.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

NuBus Slot Count Selector

Specifies information about the number of NuBus slots.

```
enum {
    gestaltNuBusSlotCount = 'nubs'
};
```

OCE Toolbox Attribute Selectors

Specify feature availability for the OCE Toolbox.

```
enum {
    gestaltOCEToolboxAttr = 'oceu',
    gestaltOCETBPresent = 0x01,
    gestaltOCETBAvailable = 0x02,
    gestaltOCESFServerAvailable = 0x04,
    gestaltOCETBNativeGlueAvailable = 0x10
};
```

OCE Toolbox Version Selectors

Specify version information for the OCE Toolbox.

```
enum {
    gestaltOCEToolboxVersion = 'ocet',
    gestaltOCETB = 0x0102,
    gestaltSFServer = 0x0100
};
```

Open Firmware Selector

Specifies version information for Open Firmware.

```
enum {
    gestaltOpenFirmwareInfo = 'opfw'
};
```

Open Firmware Safe Selectors

Specify feature availability for Open Firmware safe features.

```
enum {
    gestaltSafeOFAttr = 'safe',
    gestaltVMZerosPagesBit = 0,
    gestaltInitHeapZerosOutHeapsBit = 1,
    gestaltNewHandleReturnsZeroedMemoryBit = 2,
    gestaltNewPtrReturnsZeroedMemoryBit = 3,
    gestaltFileAllocationZeroedBlocksBit = 4
};
```

Open Transport Selectors

Specify version and feature availability information for Open Transport.

```
enum {
    gestaltOpenTpt = 'otan',
    gestaltOpenTptPresentMask = 0x00000001,
    gestaltOpenTptLoadedMask = 0x00000002,
    gestaltOpenTptAppleTalkPresentMask = 0x00000004,
    gestaltOpenTptAppleTalkLoadedMask = 0x00000008,
    gestaltOpenTptTCPPresentMask = 0x00000010,
    gestaltOpenTptTCPLoadedMask = 0x00000020,
    gestaltOpenTptIPXSPXPresentMask = 0x00000040,
    gestaltOpenTptIPXSPXLoadedMask = 0x00000080,
    gestaltOpenTptPresentBit = 0,
    gestaltOpenTptLoadedBit = 1,
    gestaltOpenTptAppleTalkPresentBit = 2,
    gestaltOpenTptAppleTalkLoadedBit = 3,
    gestaltOpenTptTCPPresentBit = 4,
    gestaltOpenTptTCPLoadedBit = 5,
    gestaltOpenTptIPXSPXPresentBit = 6,
    gestaltOpenTptIPXSPXLoadedBit = 7
};
```

Open Transport Network Setup Selectors

Specify feature availability and setup information for Open Transport networking.

```
enum {
    gestaltOpenTptNetworkSetup = 'otcf',
    gestaltOpenTptNetworkSetupLegacyImport = 0,
    gestaltOpenTptNetworkSetupLegacyExport = 1,
    gestaltOpenTptNetworkSetupSupportsMultihoming = 2
};
```

Open Transport Network Version Selector

Specifies the version of the Open Transport network setup.

```
enum {
    gestaltOpenTptNetworkSetupVersion = 'otcv'
};
```

Open Transport Remote Access Selectors

Specify feature availabiliy for Open Transport remote access.

```
enum {
    gestaltOpenTptRemoteAccess = 'otra',
    gestaltOpenTptRemoteAccessPresent = 0,
    gestaltOpenTptRemoteAccessLoaded = 1,
    gestaltOpenTptRemoteAccessClientOnly = 2,
    gestaltOpenTptRemoteAccessPServer = 3,
    gestaltOpenTptRemoteAccessMPServer = 4,
    gestaltOpenTptPPPPresent = 5,
    gestaltOpenTptARAPPresent = 6
};
```

Opent Transport Remote Access Version Selector

Specifies version information for Open Transport remote access.

```
enum {
    gestaltOpenTptRemoteAccessVersion = 'otrv'
};
```

Open Transport Version Selector

Specifies version information for Open Transport.

```
enum {
    gestaltOpenTptVersions = 'otvr'
};
```

OS Trap Table Selector

Specifies base address information for the operating system trap dispatch table.

```
enum {
    gestaltOSTable = 'ostt'
};
```

Constants

`gestaltOSTable`

The selector you pass to the `Gestalt` function to determine the base address of the operating system trap dispatch table.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Parity Checking Attribute Selectors

Specify feature availability for parity checking.

```
enum {
    gestaltParityAttr = 'prty',
    gestaltHasParityCapability = 0,
    gestaltParityEnabled = 1
};
```

Constants

`gestaltParityAttr`
The selector you pass to the `Gestalt` function to determine information about the machine's parity-checking features.

Note that parity is not considered to be enabled unless all installed memory is parity RAM.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasParityCapability`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltParityEnabled`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

PC Compatibility Card Selectors

Specify version and feature availability information for a PC-compatibility card.

```
enum {
    gestaltPCCard = 'pccd',
    gestaltCardServicesPresent = 0,
    gestaltPCCardFamilyPresent = 1,
    gestaltPCCardHasPowerControl = 2,
    gestaltPCCardSupportsCardBus = 3
};
```

PC Exchange Attribute Selectors

Specify feature availability information for PC Exchange.

```
enum {
    gestaltPCXAttr = 'pcxg',
    gestaltPCXHas8and16BitFAT = 0,
    gestaltPCXHasProDOS = 1,
    gestaltPCXNewUI = 2,
    gestaltPCXUseICMapping = 3
};
```

Constants

`gestaltPCXAttr`
The selector you pass to the `Gestalt` function to determine the PC Exchange attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPCXHas8and16BitFAT`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPCXHasProDOS`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPCXNewUI`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPCXUseICMapping`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Physical RAM Size Selector

Specifies information about the size of the physical RAM.

```
enum {
    gestaltPhysicalRAMSize = 'ram '
};
```

Constants

`gestaltPhysicalRAMSize`
The selector you pass to the `Gestalt` function to determine the number of bytes of physical RAM currently installed.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Pop-up Control Selector

Specify feature availability for pop-up controls.

```
enum {
    gestaltPopupAttr = 'pop!',
    gestaltPopupPresent = 0
};
```

Constants

`gestaltPopupAttr`
The selector you pass to the `Gestalt` function to determine the attribute of the pop-up control definition.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPopupPresent`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Power Manager Attribute Selectors

Specify feature availability for the Power Manager.

```
enum {
    gestaltPowerMgrAttr = 'powr',
    gestaltPMgrExists = 0,
    gestaltPMgrCPUIdle = 1,
    gestaltPMgrSCC = 2,
    gestaltPMgrSound = 3,
    gestaltPMgrDispatchExists = 4,
    gestaltPMgrSupportsAVPowerStateAtSleepWake = 5
};
```

Constants

`gestaltPowerMgrAttr`
The `Gestalt` selector you pass to determine which Power Manager capabilities are available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPMgrExists`
If true, the Power Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPMgrCPUIdle`
If true the CPU is capable of going into a low–power-consumption state.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPMgrSCC`
If true, it is possible to stop the SCC clock, thus effectively turning off the serial ports.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPMgrSound`
If true, it is possible to turn off power to the sound circuits.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPMgrDispatchExists`
If true, Dispatch is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPMgrSupportsAVPowerStateAtSleepWake`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Power Manager Version Selector

Specifies version information for the Power Manager.

```
enum {
    gestaltPowerMgrVers = 'pwrv'
};
```

PowerPC Attribute Selectors

Specify feature availability for PowerPC processors.

```
enum {
    gestaltPowerPCProcessorFeatures = 'ppcf',
    gestaltPowerPCHasGraphicsInstructions = 0,
    gestaltPowerPCHasSTFIWXInstruction = 1,
    gestaltPowerPCHasSquareRootInstructions = 2,
    gestaltPowerPCHasDCBAInstruction = 3,
    gestaltPowerPCHasVectorInstructions = 4,
    gestaltPowerPCHasDataStreams = 5
};
```

PowerPC Toolbox Attribute Selectors

Specify feature availability for the PowerPC Toolbox.

```
enum {
    gestaltPPCToolboxAttr = 'ppc ',
    gestaltPPCToolboxPresent = 0x0000,
    gestaltPPCSupportsRealTime = 0x1000,
    gestaltPPCSupportsIncoming = 0x0001,
    gestaltPPCSupportsOutGoing = 0x0002,
    gestaltPPCSupportsTCP_IP = 0x0004,
    gestaltPPCSupportsIncomingAppleTalk = 0x0010,
    gestaltPPCSupportsIncomingTCP_IP = 0x0020,
    gestaltPPCSupportsOutgoingAppleTalk = 0x0100,
    gestaltPPCSupportsOutgoingTCP_IP = 0x0200
};
```

Constants

`gestaltPPCToolboxAttr`
: The selector you pass to the Gestalt function to determine the Program-to-Program Communication (PPC) Toolbox attributes. Note that these constants are defined as masks, not bit numbers.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltPPCToolboxPresent`
: Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltPPCSupportsRealTime`
: Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltPPCSupportsIncoming`
: Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltPPCSupportsOutGoing`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPPCSupportsTCP_IP`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPPCSupportsIncomingAppleTalk`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPPCSupportsIncomingTCP_IP`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPPCSupportsOutgoingAppleTalk`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPPCSupportsOutgoingTCP_IP`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Preemptive Function Atrribute Selectors

Specify feature availability information for preemptive system software functions.

```
enum {
    gestaltMPCallableAPIsAttr = 'mpsc',
    gestaltMPFileManager = 0,
    gestaltMPDeviceManager = 1,
    gestaltMPTrapCalls = 2
};
```

Constants

`gestaltMPCallableAPIsAttr`
The Gestalt selector passed to determine the availability of preemptive system software functions. The `Gestalt` function produces a 32-bit value that you should test to determine which what type of preemptive calls are allowed.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMPFileManager`
If this bit is set, you can call preemptively safe File Manager functions.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMPDeviceManager`
If this bit is set, you can call preemptively safe Device Manager function.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMPTrapCalls`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any Mac OS system software functions from a preemptive task , you should call the `Gestalt` function with the `gestaltMPCallableAPIsAttr` selector set to determine which preemptively safe system calls are allowed.

Note that for functions that are shared between managers (for example, `PBCloseSync`), you should check the bit that is appropriate for the manager you want to call.

Version Notes
Introduced with Multiprocessing Services 2.1

Processor Clock Speed Selector

Specifies information about processor clock speed.

```
enum {
    gestaltProcClkSpeed = 'pclk'
};
```

Processor Type Selector

Specifies information about the type of microprocessor.

```
enum {
    gestaltProcessorType = 'proc',
    gestalt68000 = 1,
    gestalt68010 = 2,
    gestalt68020 = 3,
    gestalt68030 = 4,
    gestalt68040 = 5
};
```

Constants
`gestaltProcessorType`
The selector you pass to the `Gestalt` function to determine the type of microprocessor currently running.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68000`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68010`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68020`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68030`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68040`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Quadra Redefinitions

Specifies alternate names for MacQuadra constants.

```
enum {
    gestaltQuadra605 = gestaltMacQuadra605,
    gestaltQuadra610 = gestaltMacQuadra610,
    gestaltQuadra630 = gestaltMacQuadra630,
    gestaltQuadra650 = gestaltMacQuadra650,
    gestaltQuadra660AV = gestaltMacQuadra660AV,
    gestaltQuadra700 = gestaltMacQuadra700,
    gestaltQuadra800 = gestaltMacQuadra800,
    gestaltQuadra840AV = gestaltMacQuadra840AV,
    gestaltQuadra900 = gestaltMacQuadra900,
    gestaltQuadra950 = gestaltMacQuadra950
};
```

QuickDraw 3D Attribute Selectors

Specify feature availability information for QuickDraw 3D.

```
enum {
    gestaltQD3D = 'qd3d',
    gestaltQD3DPresent = 0
};
```

Quick Draw 3D Old Attribute Selectors

Specify old feature availability information for QuickDraw 3D.

```
enum {
    gestaltQD3DNotPresent = (0 << gestaltQD3DPresent),
    gestaltQD3DAvailable = (1 << gestaltQD3DPresent)
};
```

Quick Draw 3D Version Selector

Specifies version information for QuickDraw 3D.

```
enum {
    gestaltQD3DVersion = 'q3v '
};
```

QuickDraw 3D Viewer Attribute Selectors

Specify feature availablity information for QuickDraw 3D Viewer.

```
enum {
    gestaltQD3DViewer = 'q3vc',
    gestaltQD3DViewerPresent = 0
};
```

QuickDraw Attribute Selectors

Specify feature availability information for QuickDraw.

```
enum {
    gestaltQuickdrawFeatures = 'qdrw',
    gestaltHasColor = 0,
    gestaltHasDeepGWorlds = 1,
    gestaltHasDirectPixMaps = 2,
    gestaltHasGrayishTextOr = 3,
    gestaltSupportsMirroring = 4,
    gestaltQDHasLongRowBytes = 5
};
```

Constants

`gestaltQuickdrawFeatures`
The selector you pass to the `Gestalt` function to determine the QuickDraw features.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasColor`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasDeepGWorlds`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasDirectPixMaps`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasGrayishTextOr`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSupportsMirroring`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltQDHasLongRowBytes`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

QuickDraw Version Selectors

Specify version information for QuickDraw.

```
enum {
    gestaltQuickdrawVersion = 'qd  ',
    gestaltOriginalQD = 0x0000,
    gestalt8BitQD = 0x0100,
    gestalt32BitQD = 0x0200,
    gestalt32BitQD11 = 0x0201,
    gestalt32BitQD12 = 0x0220,
    gestalt32BitQD13 = 0x0230,
    gestaltAllegroQD = 0x0250,
    gestaltMacOSXQD = 0x0300
};
```

Constants

`gestaltQuickdrawVersion`
The Gestalt selector you pass to determine what version of QuickDraw is present. For QuickDraw Text, the Gestalt selector you pass to determine what version of QuickDraw Text is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltOriginalQD`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt8BitQD`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt32BitQD`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt32BitQD11`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt32BitQD12`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt32BitQD13`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAllegroQD`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMacOSXQD`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
The version of QuickDraw is encoded as a revision number in the low-order word of the return value. The high-order byte represents the major revision number, and the low-order byte represents the minor revision number. For example, version 1.3 of 32-Bit QuickDraw represents QuickDraw revision 2.3; its response value is $0230.

Values having a major revision number of 1 or 2 indicate that Color QuickDraw is available, in either the 8-bit or 32-bit version. These results do not, however, indicate whether a color monitor is attached to the system. You must use high-level QuickDraw functions to obtain that information.

QuickDraw GX Overall Version Selector

Specifies version information for the overall version of QuickDraw GX.

```
enum {
    gestaltGXVersion = 'qdgx'
};
```

QuickDraw GX Printing Version Selector

Specifies version information for QuickDraw GX printing.

```
enum {
    gestaltGXPrintingMgrVersion = 'pmgr'
};
```

QuickDraw GX Version Selectors

Specify version information for QuickDraw GX.

```
enum {
    gestaltGraphicsVersion = 'grfx',
    gestaltCurrentGraphicsVersion = 0x00010200
};
```

Constants
`gestaltGraphicsVersion`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCurrentGraphicsVersion`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

QuickDraw GX Attribute Selectors

Specify feature availability information for QuickDraw GX.

```
enum {
    gestaltGraphicsAttr = 'gfxa',
    gestaltGraphicsIsDebugging = 0x00000001,
    gestaltGraphicsIsLoaded = 0x00000002,
    gestaltGraphicsIsPowerPC = 0x00000004
};
```

QuickDraw 3D Viewer Old Selectors

Specify old feature availability information for QuickDraw 3D.

```
enum {
    gestaltQD3DViewerNotPresent = (0 << gestaltQD3DViewerPresent),
    gestaltQD3DViewerAvailable = (1 << gestaltQD3DViewerPresent)
};
```

QuickDraw Text Attribute Selectors

Specify feature availability information for QuickDraw Text.

```
enum {
    gestaltQDTextFeatures = 'qdtf',
    gestaltWSIISupport = 0,
    gestaltSbitFontSupport = 1,
    gestaltAntiAliasedTextAvailable = 2,
    gestaltOFA2available = 3,
    gestaltCreatesAliasFontRsrc = 4,
    gestaltNativeType1FontSupport = 5,
    gestaltCanUseCGTextRendering = 6
};
```

Constants

`gestaltQDTextFeatures`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltWSIISupport`

WSII support is included.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSbitFontSupport`

sbit-only fonts are supported.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAntiAliasedTextAvailable`

Capable of antialiased text.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltOFA2available`
OFA2 is available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCreatesAliasFontRsrc`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltNativeType1FontSupport`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltCanUseCGTextRendering`
Available in Mac OS X v10.2 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any function dependent upon QuickDraw Text, your application should pass the selector `gestaltQDTextFeatures` to the Gestalt function to determine the QuickDraw Text attributes that are present.

QuickDraw Text Version Selectors

Specify version information for QuickDraw Text.

```
enum {
    gestaltQDTextVersion = 'qdtx',
    gestaltOriginalQDText = 0x0000,
    gestaltAllegroQDText = 0x0100,
    gestaltMacOSXQDText = 0x0200
};
```

Constants
`gestaltQDTextVersion`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltOriginalQDText`
This is the original version of QuickDraw Text, used through Mac OS 8.1.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltAllegroQDText`
This is the version of QuickDraw Text used with Mac OS 8.2 and up.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMacOSXQDText`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
To determine the version of the current QuickDraw Text, your application should pass the selector `gestaltQuickdrawVersion` to the Gestalt function.

QuickTime VR Feature Selectors

Specify feature availability information for QuickTime VR.

```
enum {
    gestaltQTVRMgrAttr = 'qtvr',
    gestaltQTVRMgrPresent = 0,
    gestaltQTVRObjMoviesPresent = 1,
    gestaltQTVRCylinderPanosPresent = 2,
    gestaltQTVRCubicPanosPresent = 3
};
```

QuickTime VR Version Selector

Specifies version information for QuickTime VR.

```
enum {
    gestaltQTVRMgrVers = 'qtvv'
};
```

QuickTime Attribute Selectors

Specify feature availability information for QuickTime.

```
enum {
    gestaltQuickTimeFeatures = 'qtrs',
    gestaltPPCQuickTimeLibPresent = 0
};
```

QuickTime Version Selectors

Specify version information for QuickTime.

```
enum {
    gestaltQuickTimeVersion = 'qtim',
    gestaltQuickTime = 'qtim'
};
```

Constants

`gestaltQuickTimeVersion`

The selector you pass to the `Gestalt` function to determine the QuickTime version.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltQuickTime`

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

QuickTime Conferencing Information Selector

Specifies information about QuickTime conferencing.

```
enum {
    gestaltQuickTimeConferencingInfo = 'qtci'
};
```

Constants

`gestaltQuickTimeConferencingInfo`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

QuickTime Conferencing Selector

Specifies availability information for QuickTime conferencing.

```
enum {
    gestaltQuickTimeConferencing = 'mtlk'
};
```

QuickTime Streaming Attribute Selector

Specify feature availability information for QuickTime streaming.

```
enum {
    gestaltQuickTimeStreamingFeatures = 'qtsf'
};
```

QuickTime Streaming Version Selector

Specifies version information for QuickTime streaming.

```
enum {
    gestaltQuickTimeStreamingVersion = 'qtst'
};
```

RBV Address Selector

Specifies information about the RBV base address.

```
enum {
    gestaltRBVAddr = 'rbv '
};
```

Realtime Manager Attribute Selectors

Specify feature availability information for the Realtime Manager.

```
enum {
    gestaltRealtimeMgrAttr = 'rtmr',
    gestaltRealtimeMgrPresent = 0
};
```

Constants

`gestaltRealtimeMgrAttr`
The selector you pass to the `Gestalt` function to determine the Realtime Manager attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltRealtimeMgrPresent`
(description forthcoming)

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Resource Manager Bug Fixes Attribute Selectors

Specify feature availability information for Resource Manager bug fixes.

```
enum {
    gestaltResourceMgrBugFixesAttrs = 'rmbg',
    gestaltRMForceSysHeapRolledIn = 0,
    gestaltRMFakeAppleMenuItemsRolledIn = 1,
    gestaltSanityCheckResourceFiles = 2,
    gestaltSupportsFSpResourceFileAlreadyOpenBit = 3,
    gestaltRMSupportsFSCalls = 4,
    gestaltRMTypeIndexOrderingReverse = 8
};
```

Resource Manager Attribute Selectors

Specify feature availability information for the Resource Manager.

```
enum {
    gestaltResourceMgrAttr = 'rsrc',
    gestaltPartialRsrcs = 0,
    gestaltHasResourceOverrides = 1
};
```

Constants

`gestaltResourceMgrAttr`
The `Gestalt` selector you pass to determine which Resource Manager attributes are present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPartialRsrcs`
If `true`, partial resources exist.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasResourceOverrides`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any function dependent upon the Resource Manager, your application should pass the selector `gestaltResourceMgrAttr` to the `Gestalt` function to determine the Resource Manager attributes that are present.

ROM Size Selector

Specifies information about ROM size information.

```
enum {
    gestaltROMSize = 'rom '
};
```

Constants
`gestaltROMSize`
The selector you pass to the `Gestalt` function to determine the size of the installed ROM, in bytes. The value is returned in only one word.

You should not infer the existence of certain hardware or software features from the responses that Gestalt returns when you pass it this selector.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

ROM Version Selector

Specifies ROM version information.

```
enum {
    gestaltROMVersion = 'romv'
};
```

Constants
`gestaltROMVersion`
This selector is NOT supported in Carbon.

The selector you pass to the `Gestalt` function to determine the version number of the installed ROM (in the low-order word of the return value).

Never infer the existence of certain hardware or software features from the responses that `Gestalt` returns when you pass it this selector.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

SCC Read Address Selector

Specifies information about the base address for reading SCC.

```
enum {
    gestaltSCCReadAddr = 'sccr'
};
```

SCC Write Address Selector

Specifies information about the base address for writing SCC.

```
enum {
    gestaltSCCWriteAddr = 'sccw'
};
```

SCSI Manager Attribute Selectors

Specify feature availability information for the SCSI Manager.

```
enum {
    gestaltSCSI = 'scsi',
    gestaltAsyncSCSI = 0,
    gestaltAsyncSCSIINROM = 1,
    gestaltSCSISlotBoot = 2,
    gestaltSCSIPollSIH = 3
};
```

Scrap Manager Selectors

Specify version and feature availability information for the Scrap Manager.

```
enum {
    gestaltScrapMgrAttr = 'scra',
    gestaltScrapMgrTranslationAware = 0
};
```

Constants

`gestaltScrapMgrAttr`

The `Gestalt` selector you pass to determine which Scrap Manager attributes are present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltScrapMgrTranslationAware`

If `true`, the Scrap Manager supports Translation Manager.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion

Before calling any function dependent upon the Scrap Manager, your application should pass the selector `gestaltScrapMgrAttr` to the `Gestalt` function to determine the Scrap Manager attributes that are present.

Screen Capture Selectors

Specifies location information for screen capture.

```
enum {
    gestaltScreenCaptureMain = 'pic1',
    gestaltScreenCaptureDir = 'pic2'
};
```

Script Manager Version Selector

Specifies version information for the Script Manager.

```
enum {
    gestaltScriptMgrVersion = 'scri'
};
```

Constants

`gestaltScriptMgrVersion`

The selector you pass to the `Gestalt` function to determine the version number of the Script Manager (in the low-order word of the return value).

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Script Systems Count Selector

Specifies information about the number of active script systems.

```
enum {
    gestaltScriptCount = 'scr#'
};
```

Constants

`gestaltScriptCount`

The selector you pass to the Gestalt function to determine the number of script systems currently active.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Serial Hardware Attribute Selectors

Specify serial hardware attributes.

```
enum {
    gestaltSerialAttr = 'ser ',
    gestaltHasGPIaToDCDa = 0,
    gestaltHasGPIaToRTxCa = 1,
    gestaltHasGPIbToDCDb = 2,
    gestaltHidePortA = 3,
    gestaltHidePortB = 4,
    gestaltPortADisabled = 5,
    gestaltPortBDisabled = 6
};
```

Constants

`gestaltSerialAttr`
The selector you pass to the `Gestalt` function to determine the serial hardware attributes of the machine, such as whether or not the GPIa line is connected and can be used for external clocking.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasGPIaToDCDa`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasGPIaToRTxCa`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasGPIbToDCDb`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHidePortA`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHidePortB`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPortADisabled`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPortBDisabled`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Serial Port Arbitrator Attribute Selectors

Specify feature availability information for serial port arbitration.

```
enum {
    gestaltArbitorAttr = 'arb ',
    gestaltSerialArbitrationExists = 0
};
```

Settings Manager Attribute Selectors

Specify feature availability information for the Settings Manager.

```
enum {
    gestaltALMAttr = 'trip',
    gestaltALMPresent = 0,
    gestaltALMHasSFGroup = 1,
    gestaltALMHasCFMSupport = 2,
    gestaltALMHasRescanNotifiers = 3
};
```

Discussion
See also "Settings Manager Version Selector" (page 97).

Settings Manager Location Selector

Specifies location information for the Settings Manager.

```
enum {
    gestaltALMHasSFLocation = gestaltALMHasSFGroup
};
```

Settings Manager Version Selector

Specifies version information for the Settings Manager.

```
enum {
    gestaltALMVers = 'walk'
};
```

Shutdown Attribute Selectors

Specify shutdown attributes.

```
enum {
    gestaltShutdownAttributes = 'shut',
    gestaltShutdownHassdOnBootVolUnmount = 0
};
```

Single Window Mode Selectors

Specify single-window modes.

```
enum {
    gestaltHasSingleWindowModeBit = 8,
    gestaltHasSingleWindowModeMask = (1L << gestaltHasSingleWindowModeBit)
};
```

Slot Attribute Selectors

Specify feature availablity for slots.

```
enum {
    gestaltSlotAttr = 'slot',
    gestaltSlotMgrExists = 0,
    gestaltNuBusPresent = 1,
    gestaltSESlotPresent = 2,
    gestaltSE30SlotPresent = 3,
    gestaltPortableSlotPresent = 4
};
```

Constants

`gestaltSlotAttr`
The selector you pass to the `Gestalt` function to determine the Slot Manager attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSlotMgrExists`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltNuBusPresent`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSESlotPresent`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSE30SlotPresent`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPortableSlotPresent`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Slot Number Selector

Specifies information about the first physical slot in the computer.

```
enum {
    gestaltFirstSlotNumber = 'slt1'
};
```

Constants

`gestaltFirstSlotNumber`
The first physical slot.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Software Vendor Codes

Specify codes for software vendors.

```
enum {
    gestaltSoftwareVendorCode = 'srad',
    gestaltSoftwareVendorApple = 'Appl',
    gestaltSoftwareVendorLicensee = 'Lcns'
};
```

Sound Manager Attribute Selectors

Specify feature availability information for the Sound Manager.

```
enum {
    gestaltSoundAttr = 'snd ',
    gestaltStereoCapability = 0,
    gestaltStereoMixing = 1,
    gestaltSoundIOMgrPresent = 3,
    gestaltBuiltInSoundInput = 4,
    gestaltHasSoundInputDevice = 5,
    gestaltPlayAndRecord = 6,
    gestalt16BitSoundIO = 7,
    gestaltStereoInput = 8,
    gestaltLineLevelInput = 9,
    gestaltSndPlayDoubleBuffer = 10,
    gestaltMultiChannels = 11,
    gestalt16BitAudioSupport = 12
};
```

Constants

`gestaltSoundAttr`
The Gestalt selector which you pass to the `Gestalt` function.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltStereoCapability`
Set if the built-in sound hardware is able to produce stereo sounds.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltStereoMixing`
Set if the built-in sound hardware mixes both left and right channels of stereo sound into a single audio signal for the internal speaker.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSoundIOMgrPresent`
Set if the Sound Input Manager is available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltBuiltInSoundInput`
Set if a built-in sound input device is available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasSoundInputDevice`
Set if a sound input device is available. This device can be either built-in or external.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPlayAndRecord`
Set if the built-in sound hardware is able to play and record sounds simultaneously. If this bit is clear, the built-in sound hardware can either play or record, but not do both at once. This bit is valid only if the `gestaltBuiltInSoundInput` bit is set, and it applies only to any built-in sound input and output hardware.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt16BitSoundIO`
Set if the built-in sound hardware is able to play and record 16-bit samples. This indicates that built-in hardware necessary to handle 16-bit data is available.

This bit is not defined for versions of the Sound Manager prior to version 3.0.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltStereoInput`
Set if the built-in sound hardware can record stereo sounds.

This bit is not defined for versions of the Sound Manager prior to version 3.0.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltLineLevelInput`
Set if the built-in sound input port requires line level input.

This bit is not defined for versions of the Sound Manager prior to version 3.0.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSndPlayDoubleBuffer`
Set if the Sound Manager supports the play-from-disk functions.

This bit is not defined for versions of the Sound Manager prior to version 3.0.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltMultiChannels`
Set if the Sound Manager supports multiple channels of sound.

This bit is not defined for versions of the Sound Manager prior to version 3.0.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt16BitAudioSupport`
Set if the Sound Manager can handle 16-bit audio data. This indicates that software necessary to handle 16-bit data is available.

This bit is not defined for versions of the Sound Manager prior to version 3.0.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
You can pass the `gestaltSoundAttr` selector to the `Gestalt` function to determine information about the sound input capabilities of a Macintosh computer.

The `Gestalt` function returns information by setting or clearing bits in the `response` parameter. The bits relevant to the Sound Input Manager are defined by constants.

Speech Manager Attribute Selectors

Specify feature availability information for the Speech Manager.

```
enum {
    gestaltSpeechAttr = 'ttsc',
    gestaltSpeechMgrPresent = 0,
    gestaltSpeechHasPPCGlue = 1
};
```

Constants
`gestaltSpeechAttr`
The selector you pass to the `Gestalt` function to determine the Speech Manager attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSpeechMgrPresent`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSpeechHasPPCGlue`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Speech Recognition Version Selector

Specifies version information for the Speech Recognition Manager.

```
enum {
    gestaltSpeechRecognitionVersion = 'srtb'
};
```

Speech Recognition Manager Attribute Selectors

Specify feature availability information for the Speech Recognition Manager.

```
enum {
    gestaltSpeechRecognitionAttr = 'srta',
    gestaltDesktopSpeechRecognition = 1,
    gestaltTelephoneSpeechRecognition = 2
};
```

Constants

`gestaltSpeechRecognitionAttr`
The selector which you pass to the `Gestalt` function to determine the Speech Recognition Manager attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltDesktopSpeechRecognition`
If this bit is set, the Speech Recognition Manager supports the desktop microphone.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTelephoneSpeechRecognition`
If this bit is set, the Speech Recognition Manager supports telephone input. In versions 1.5 and earlier, this bit is always 0.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion

You can pass the `gestaltSpeechRecognitionAttr` selector to the `Gestalt` function to get the attributes of the Speech Recognition Manager. `Gestalt` returns information to you by returning a long word in the `response` parameter. The returned values are defined by these constants.

Standard Directory Find Panel Selector

Specifies version information for the standard directory find panel.

```
enum {
    gestaltSDPFindVersion = 'dfnd'
};
```

Standard Directory Prompt Panel Selector

Specifies version information for the standard directory prompt panel.

```
enum {
    gestaltSDPPromptVersion = 'prpv'
};
```

Standard Directory Version Selector

Specifies version information for the standard directory.

```
enum {
    gestaltSDPStandardDirectoryVersion = 'sdvr'
};
```

Startup Disk Attribute Selectors

Specify feature availability information for the startup disk.

```
enum {
    gestaltSplitOSAttr = 'spos',
    gestaltSplitOSBootDriveIsNetworkVolume = 0,
    gestaltSplitOSAware = 1,
    gestaltSplitOSEnablerVolumeIsDifferentFromBootVolume = 2,
    gestaltSplitOSMachineNameSetToNetworkNameTemp = 3,
    gestaltSplitOSMachineNameStartupDiskIsNonPersistent = 5
};
```

Standard File Attribute Selectors

Specify feature availabity information for Standard File.

```
enum {
    gestaltStandardFileAttr = 'stdf',
    gestaltStandardFile58 = 0,
    gestaltStandardFileTranslationAware = 1,
    gestaltStandardFileHasColorIcons = 2,
    gestaltStandardFileUseGenericIcons = 3,
    gestaltStandardFileHasDynamicVolumeAllocation = 4
};
```

Constants

`gestaltStandardFileAttr`

The selector you pass to the Gestalt function to determine the Standard File Package attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltStandardFile58`

If the `gestaltStandardFile58` flag bit is set, you can call the four new procedures—`StandardPutFile`, `StandardGetFile`, `CustomPutFile`, and `CustomGetFile`—introduced with System 7. (The name of the constant reflects the enabling of selectors 5 through 8 on the trap macro that handles the Standard File Package.)

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltStandardFileTranslationAware`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltStandardFileHasColorIcons`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltStandardFileUseGenericIcons`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltStandardFileHasDynamicVolumeAllocation`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

System Architecture Selectors

Specify the native system architecture.

```
enum {
    gestaltSysArchitecture = 'sysa',
    gestalt68k = 1,
    gestaltPowerPC = 2,
    gestaltIntel = 10
};
```

Constants

`gestaltSysArchitecture`
The selector you pass to the `Gestalt` function to determine the native system architecture.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestalt68k`
If the `Gestalt` function returns `gestalt68k`, the system is a MC680x0 Macintosh.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltPowerPC`
If the `Gestalt` function returns `gestaltPowerPC`, the system is a PowerPC Macintosh.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltIntel`
If the `Gestalt` function returns `gestaltIntel`, the system is is an Intel-based Macintosh.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

System Update Version Selector

Specifies version information for system updates.

```
enum {
    gestaltSystemUpdateVersion = 'sysu'
};
```

System Version Selectors

Specifies version information for the operating system.

```
enum {
    gestaltSystemVersion = 'sysv'
    gestaltSystemVersionMajor = 'sys1',
    gestaltSystemVersionMinor = 'sys2',
    gestaltSystemVersionBugFix = 'sys3'
};
```

Constants

`gestaltSystemVersion`

The selector you pass to the `Gestalt` function to determine the version number of the currently active System file. For systems prior to Mac OS X, the version is represented as four hexadecimal digits in the low-order word of the return value. For example, if your application is running in version 7.0.1, then `Gestalt` returns the value 0x0701. Ignore the high-order word of the returned value. For Mac OS X versions, the representation is as shown in Table 1.

Table 1 The representation of Mac OS X versions by the Gestalt Manager

Mac OS X Version	Representation
10.0	0x1000
10.1	0x1010
10.2	0x1020
10.3	0x1030
10.4	0x1040

If the values of the minor or bug fix revision are larger than 9, then `gestaltSystemVersion` will substitute the value 9 for them. For example, Mac OS X 10.3.15 will be returned as 0x1039, and Mac OS X 10.10.5 will return 0x1095.

Never infer the existence of certain hardware or software features from the responses that `Gestalt` returns when you pass it this selector.

In Mac OS X v10.4 and later, a better way to get system version information is to use the selectors `gestaltSystemVersionMajor`, `gestaltSystemVersionMinor`, and `gestaltSystemVersionBugFix`, which are listed below. These selectors don't have arbitrary limits on the values returned.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSystemVersionMajor`
The major system version number. For example, in 10.4.12, this would be the decimal value 10.

Available in Mac OS X v10.3 and later.

Declared in `Gestalt.h`.

`gestaltSystemVersionMinor`
The minor system version number. For example, in 10.4.12, this would be the decimal value 4.

Available in Mac OS X v10.3 and later.

Declared in `Gestalt.h`.

`gestaltSystemVersionBugFix`
The bug fix version number. For example, in 10.4.12, this would be the decimal value 12.

Available in Mac OS X v10.3 and later.

Declared in `Gestalt.h`.

Telephone Manager Attribute Selectors

Specify feature availability information for the Telephone Manager.

```
enum {
    gestaltTeleMgrAttr = 'tele',
    gestaltTeleMgrPresent = 0,
    gestaltTeleMgrPowerPCSupport = 1,
    gestaltTeleMgrSoundStreams = 2,
    gestaltTeleMgrAutoAnswer = 3,
    gestaltTeleMgrIndHandset = 4,
    gestaltTeleMgrSilenceDetect = 5,
    gestaltTeleMgrNewTELNewSupport = 6
};
```

Terminal Manager Attribute Selectors

Specify feature availability information for the Terminal Manager.

```
enum {
    gestaltTermMgrAttr = 'term',
    gestaltTermMgrPresent = 0,
    gestaltTermMgrErrorString = 2
};
```

Constants

`gestaltTermMgrAttr`
The selector you pass to the `Gestalt` function to determine the Terminal Manager attributes.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTermMgrPresent`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTermMgrErrorString`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

TextEdit Attribute Selectors

Specify feature availability information for TextEdit.

```
enum {
    gestaltTEAttr = 'teat',
    gestaltTEHasGetHiliteRgn = 0,
    gestaltTESupportsInlineInput = 1,
    gestaltTESupportsTextObjects = 2,
    gestaltTEHasWhiteBackground = 3
};
```

Constants

`gestaltTEAttr`
The `Gestalt` selector you pass to determine whichTextEdit attributes are present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTEHasGetHiliteRgn`
If true, TextEdit has `TEGetHiliteRgn`.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTESupportsInlineInput`
If `true`, TextEdit does Inline Input.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTESupportsTextObjects`
If `true`, TextEdit does Text Objects.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTEHasWhiteBackground`
If `true`, TextEdit supports overriding the `TERec`' data structure background field to white.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any function dependent upon TextEdit, your application should pass the selector `gestaltTEAttr` to the `Gestalt` function to determine the TextEdit attributes that are present.

TextEdit Version Selectors

Specify version information for TextEdit.

```
enum {
    gestaltTextEditVersion = 'te  ',
    gestaltTE1 = 1,
    gestaltTE2 = 2,
    gestaltTE3 = 3,
    gestaltTE4 = 4,
    gestaltTE5 = 5
};
```

Constants

`gestaltTextEditVersion`
The `Gestalt` selector you pass to determine what version of TextEdit is present.
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltTE1`
The version of TextEdit found in Mac IIci ROM.
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltTE2`
The version of TextEdit shipped with 6.0.4 Script Systems on Mac IIci (Script bug fixes for Mac IIci).
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltTE3`
The version of TextEdit shipped with 6.0.4 Script Systems (all but Mac IIci).
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltTE4`
The version of TextEdit shipped in System 7.0.
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltTE5`
`TextWidthHook` is available in TextEdit.
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

Discussion

To determine the version of the current TextEdit, your application should pass the selector `gestalt` TextEditVersion to the `Gestalt` function.

Text Services Manager Attribute Selectors

Specify feature availability information for the Text Services Manager.

```
enum {
    gestaltTSMgrAttr = 'tsma',
    gestaltTSMDisplayMgrAwareBit = 0,
    gestaltTSMdoesTSMTEBit = 1
};
```

Text Services Manager Version Selectors

Specifies version information for the Text Services Manager.

```
enum {
    gestaltTSMgrVersion = 'tsmv',
    gestaltTSMgr15 = 0x0150,
    gestaltTSMgr20 = 0x0200
};
```

Constants

`gestaltTSMgrVersion`
The selector you pass to the `Gestalt` function to determine the version of the Text Services Manager.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTSMgr15`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTSMgr20`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Thread Manager Attribute Selectors

Specify feature availability information for the Thread Manager.

```
enum {
    gestaltThreadMgrAttr = 'thds',
    gestaltThreadMgrPresent = 0,
    gestaltSpecificMatchSupport = 1,
    gestaltThreadsLibraryPresent = 2
};
```

Constants

`gestaltThreadMgrAttr`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltThreadMgrPresent`
This bit is set if the Thread Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSpecificMatchSupport`
: This bit is set if the Thread Manager supports the allocation of threads based on an exact match with the requested stack size. If this bit is not set, the Thread Manager allocates threads based on the closest match to the requested stack size.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltThreadsLibraryPresent`
: This bit is set if the native version of the threads library has been loaded.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

Discussion
Before calling any function dependent upon the Thread Manager, your application should pass the selector `gestaltThreadMgrAttr` to the `Gestalt` function to determine the Thread Manager attributes that are present.

Time Manager Version Selectors

Specify version information for the Time Manager.

```
enum {
    gestaltTimeMgrVersion = 'tmgr',
    gestaltStandardTimeMgr = 1,
    gestaltRevisedTimeMgr = 2,
    gestaltExtendedTimeMgr = 3,
    gestaltNativeTimeMgr = 4
};
```

Constants
`gestaltTimeMgrVersion`
: The Gestalt selector you pass to determine what version of the Time Manager is present.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltStandardTimeMgr`
: If this bit is set, the original Time Manager is present.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltRevisedTimeMgr`
: If this bit is set, the revised Time Manager is present.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltExtendedTimeMgr`
: If this bit is set, the extended Time Manager is present.

  Available in Mac OS X v10.0 and later.

  Declared in `Gestalt.h`.

`gestaltNativeTimeMgr`
If this bit is set, the native Time Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion
To determine the version of the current Time Manager, your application should pass the selector `gestaltTimeMgrVersion` to the `Gestalt` function.

Toolbox Trap Table Selector

Specifes base address information for the Toolbox trap dispatch table.

```
enum {
    gestaltToolboxTable = 'tbtt'
};
```

Constants
`gestaltToolboxTable`
The selector you pass to the `Gestalt` function to determine the base address of the Toolbox trap dispatch table.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Toolbox Trap Table (Second Half) Selector

Specifies address information for the second half of the Toolbox trap table.

```
enum {
    gestaltExtToolboxTable = 'xttt'
};
```

Constants
`gestaltExtToolboxTable`
The base address of the second half of the Toolbox trap table if the table is discontiguous. If the table is contiguous, this selector returns 0.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Translation Manager Attribute Selectors

Specify feature availability information for the Translation Manager.

```
enum {
    gestaltTranslationAttr = 'xlat',
    gestaltTranslationMgrExists = 0,
    gestaltTranslationMgrHintOrder = 1,
    gestaltTranslationPPCAvail = 2,
    gestaltTranslationGetPathAPIAvail = 3
};
```

Constants

`gestaltTranslationAttr`
The `Gestalt` selector you pass to determine which Translation Manager attributes are present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTranslationMgrExists`
If `true`, the Translation Manager is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTranslationMgrHintOrder`
In earlier versions of the Translation Manager, the scrap hints in the `DoTranslateScrapProcPtr` function were reversed. In later versions, this was fixed. If this bit is true, this bug fix is in effect.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTranslationPPCAvail`
If `true`, the PowerPC Translation Library is available, and you can call the Translation Manager from native PowerPC code.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltTranslationGetPathAPIAvail`
If true, the functions `GetFileTranslationPath` and `GetPathTranslationDialog` are available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion

Before calling any function dependent upon the Translation Manager your application should pass the selector `gestaltTranslationAttr` to the `Gestalt` function to determine the Translation Manager attributes that are present.

TSME Version Selector

Specifies version information for the Text Services Manager integrated with TextEdit.

```
enum {
    gestaltTE6 = 6
};
```

TSMTE Attribute Selectors

Specify feature availability information for TSMTE.

```
enum {
    gestaltTSMTEAttr = 'tmTE',
    gestaltTSMTEPresent = 0,
    gestaltTSMTE = 0
};
```

TSMTE Version Selectors

Specify version information for TSMTE.

```
enum {
    gestaltTSMTEVersion = 'tmTV',
    gestaltTSMTE1 = 0x0100,
    gestaltTSMTE15 = 0x0150,
    gestaltTSMTE152 = 0x0152
};
```

TV Tuner Attribute Selectors

Specifies feature availabilty information for the TV tuner.

```
enum {
    gestaltTVAttr = 'tv  ',
    gestaltHasTVTuner = 0,
    gestaltHasSoundFader = 1,
    gestaltHasHWClosedCaptioning = 2,
    gestaltHasIRRemote = 3,
    gestaltHasVidDecoderScaler = 4,
    gestaltHasStereoDecoder = 5,
    gestaltHasSerialFader = 6,
    gestaltHasFMTuner = 7,
    gestaltHasSystemIRFunction = 8,
    gestaltIRDisabled = 9,
    gestaltINeedIRPowerOffConfirm = 10,
    gestaltHasZoomedVideo = 11
};
```

UDF Selector

Specifes support information for communication between implementations of UDF .

```
enum {
    gestaltUDFSupport = 'kudf'
};
```

USB Attribute Selectors

Specifies feature availability information for USB.

```
enum {
    gestaltUSBAttr = 'usb ',
    gestaltUSBPresent = 0,
    gestaltUSBHasIsoch = 1
};
```

USB Printer Sharing Version Selectors

Specify version information for USB printer sharing.

```
enum {
    gestaltUSBPrinterSharingVersion = 'zak ',
    gestaltUSBPrinterSharingVersionMask = 0x0000FFFF,
    gestaltUSBPrinterSharingAttr = 'zak ',
    gestaltUSBPrinterSharingAttrMask = 0xFFFF0000,
    gestaltUSBPrinterSharingAttrRunning = 0x80000000,
    gestaltUSBPrinterSharingAttrBooted = 0x40000000
};
```

USB Version Selector

Specifies version information for USB.

```
enum {
    gestaltUSBVersion = 'usbv'
};
```

VIA1 Base Address Selector

Specifies base address information for VIA 1.

```
enum {
    gestaltVIA1Addr = 'via1'
};
```

VIA2 Base Address Selector

Specifies base address information for VIA 2.

```
enum {
    gestaltVIA2Addr = 'via2'
};
```

Virtual Memory Manager Attribute Selectors

Specify feature availability information for the Virtual Memory Manager.

```
enum {
    gestaltVMAttr = 'vm  ',
    gestaltVMPresent = 0,
    gestaltVMHasLockMemoryForOutput = 1,
    gestaltVMFilemappingOn = 3,
    gestaltVMHasPagingControl = 4
};
```

Constants

`gestaltVMAttr`
The `Gestalt` selector you pass to determine the virtual memory attributes that are present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltVMPresent`
If `true`, virtual memory is present.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltVMHasLockMemoryForOutput`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltVMFilemappingOn`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltVMHasPagingControl`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

Discussion

Before calling any function dependent on memory, your application should pass the selector `gestaltVMAttr` to the `Gestalt` function to determine the virtual memory attributes that are present.

Virtual Memory Backing Store Selector

Specifes file reference number information for the VM backing store.

```
enum {
    gestaltVMBackingStoreFileRefNum = 'vmbs'
};
```

Virtual Memory Information Type Selectors

Specifes information about the VM type.

```
enum {
    gestaltVMInfoType = 'vmin',
    gestaltVMInfoSizeStorageType = 0,
    gestaltVMInfoSizeType = 1,
    gestaltVMInfoSimpleType = 2,
    gestaltVMInfoNoneType = 3
};
```

Win32 Attribute Selectors

Specify feature availability information for Win32.

```
enum {
    gestaltX86Features = 'x86f',
    gestaltX86HasFPU = 0,
    gestaltX86HasVME = 1,
    gestaltX86HasDE = 2,
    gestaltX86HasPSE = 3,
    gestaltX86HasTSC = 4,
    gestaltX86HasMSR = 5,
    gestaltX86HasPAE = 6,
    gestaltX86HasMCE = 7,
    gestaltX86HasCX8 = 8,
    gestaltX86HasAPIC = 9,
    gestaltX86Reserved10 = 10,
    gestaltX86HasSEP = 11,
    gestaltX86HasMTRR = 12,
    gestaltX86HasPGE = 13,
    gestaltX86HasMCA = 14,
    gestaltX86HasCMOV = 15,
    gestaltX86HasPAT = 16,
    gestaltX86HasPSE36 = 17,
    gestaltX86HasMMX = 23,
    gestaltX86HasFXSR = 24
};
```

Window Manager Attribute Selectors

Specify feature availability information for the Window Manager.

```
enum {
    gestaltWindowMgrAttr = 'wind',
    gestaltWindowMgrPresent = (1L << 0),
    gestaltWindowMgrPresentBit = 0,
    gestaltExtendedWindowAttributes = 1,
    gestaltExtendedWindowAttributesBit = 1,
    gestaltHasFloatingWindows = 2,
    gestaltHasFloatingWindowsBit = 2,
    gestaltHasWindowBuffering = 3,
    gestaltHasWindowBufferingBit = 3,
    gestaltWindowLiveResizeBit = 4,
    gestaltWindowMinimizeToDockBit = 5,
    gestaltHasWindowShadowsBit = 6,
    gestaltSheetsAreWindowModalBit = 7,
    gestaltFrontWindowMayBeHiddenBit = 8,
    gestaltWindowMgrPresentMask = (1L << gestaltWindowMgrPresentBit),
   gestaltExtendedWindowAttributesMask = (1L << gestaltExtendedWindowAttributesBit),
    gestaltHasFloatingWindowsMask = (1L << gestaltHasFloatingWindowsBit),
    gestaltHasWindowBufferingMask = (1L << gestaltHasWindowBufferingBit),
    gestaltWindowLiveResizeMask = (1L << gestaltWindowLiveResizeBit),
    gestaltWindowMinimizeToDockMask = (1L << gestaltWindowMinimizeToDockBit),
    gestaltHasWindowShadowsMask = (1L << gestaltHasWindowShadowsBit),
    gestaltSheetsAreWindowModalMask = (1L << gestaltSheetsAreWindowModalBit),
    gestaltFrontWindowMayBeHiddenMask = (1L << gestaltFrontWindowMayBeHiddenBit)
};
```

Constants

`gestaltWindowMgrAttr`
The `Gestalt` selector passed to determine what features of the Window Manager are present. This selector is available with Mac OS 8.5 and later. The `Gestalt` function produces a 32-bit value whose bits you should test to determine which Window Manager features are available.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltWindowMgrPresent`
If the bit specified by this mask is set, the Window Manager functionality for Appearance Manager 1.1 is available. This bit is set for Mac OS 8.5 and later.

Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltWindowMgrPresentBit`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltExtendedWindowAttributes`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltExtendedWindowAttributesBit`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasFloatingWindows`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltHasFloatingWindowsBit`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltHasWindowBuffering`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltHasWindowBufferingBit`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltWindowLiveResizeBit`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltWindowMinimizeToDockBit`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltHasWindowShadowsBit`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltSheetsAreWindowModalBit`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltFrontWindowMayBeHiddenBit`
Available in Mac OS X v10.2 and later.
Declared in `Gestalt.h`.

`gestaltWindowMgrPresentMask`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltExtendedWindowAttributesMask`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltHasFloatingWindowsMask`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltHasWindowBufferingMask`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltWindowLiveResizeMask`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltWindowMinimizeToDockMask`
Available in Mac OS X v10.0 and later.
Declared in `Gestalt.h`.

`gestaltHasWindowShadowsMask`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltSheetsAreWindowModalMask`
Available in Mac OS X v10.0 and later.

Declared in `Gestalt.h`.

`gestaltFrontWindowMayBeHiddenMask`
Available in Mac OS X v10.2 and later.

Declared in `Gestalt.h`.

Discussion
Before calling any functions dependent on the Window Manager, your application should pass the selector `gestaltWindowMgrAttr` to the `Gestalt` function to determine which Window Manager functions are available.

WorldScriptII Version Selectors

Specify version information for WorldScript II.

```
enum {
    gestaltWorldScriptIIVersion = 'doub',
    gestaltWorldScriptIIAttr = 'wsat',
    gestaltWSIICanPrintWithoutPrGeneralBit = 0
};
```

Result Codes

The most common result codes returned by the Gestalt Manager are listed below.

Result Code	Value	Description
`gestaltUnknownErr`	-5550	Specifies an unknown error. Available in Mac OS X v10.0 and later.
`gestaltUndefSelectorErr`	-5551	Specifies an undefined selector was passed to the Gestalt Manager. Available in Mac OS X v10.0 and later.
`gestaltDupSelectorErr`	-5552	Specifies you tried to add an entry that already existed. Available in Mac OS X v10.0 and later.
`gestaltLocationErr`	-5553	Specifies the gestalt function ptr was not in the system heap. Available in Mac OS X v10.0 and later.

Deprecated Gestalt Manager Functions

A function identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.3

NewGestalt

Adds a selector code to those already recognized by `Gestalt`. (Deprecated in Mac OS X v10.3. Use `NewGestaltValue` (page 15) instead.)

```
OSErr NewGestalt (
   OSType selector,
   SelectorFunctionUPP gestaltFunction
);
```

Parameters

selector

The selector code you want to add. This should be a four-character sequence similar to those defined in "Gestalt Manager Constants" (page 18).

gestaltFunction

A universal procedure pointer (UPP) to the selector callback function that `Gestalt` executes when it receives the new selector code. See `SelectorFunctionProcPtr` (page 17) for more information on the callback you need to provide.

Return Value

A result code. See "Gestalt Manager Result Codes" (page 119).

Discussion

The `NewGestalt` function registers a specified selector code with the Gestalt Manager so that when the `Gestalt` function is called with that selector code, the specified selector function is executed. Before calling `NewGestalt`, you must define a selector function callback. See `SelectorFunctionProcPtr` (page 17) for a description of how to define your selector function.

Registering with the Gestalt Manager is a way for software such as system extensions to make their presence known to potential users of their services.

Special Considerations

You should avoid using the `NewGestalt` function to add a selector code, which requires moving your selector function into the system heap. Applications do not have access to the system heap in Mac OS X.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.3.

Not available to 64-bit applications.

Declared In
`Gestalt.h`

ReplaceGestalt

Replaces the selector function associated with an existing selector code. (Deprecated in Mac OS X v10.3. Use `NewGestaltValue` (page 15) instead.)

```
OSErr ReplaceGestalt (
   OSType selector,
   SelectorFunctionUPP gestaltFunction,
   SelectorFunctionUPP *oldGestaltFunction
);
```

Parameters

`selector`

The selector code for the function you want to replace. You must provide the four-character sequence you provided previously for the function you are replacing.

`gestaltFunction`

A universal procedure pointer to the replacement selector function. You must obtain the value for this argument by calling the `NewGestaltSelectorFunctionUPP` function.

`oldGestaltFunction`

On output, a universal procedure pointer to the callback function previously associated with the specified selector. If the function `ReplaceGestalt` returns an error of any type, then the value of `oldGestaltFunction` is undefined.

Return Value
A result code. See "Gestalt Manager Result Codes" (page 119).

Special Considerations

You should avoid using the `ReplaceGestalt` function to replace an existing selector callback function, which also requires your replacement function to reside in the system heap. Applications do not have access to the system heap in Mac OS X.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.3.
Not available to 64-bit applications.

Declared In
`Gestalt.h`

Document Revision History

This table describes the changes to *Gestalt Manager Reference*.

Date	Notes
2007-10-31	Updated the description of the `gestaltSystemVersion` selector. See "System Version Selectors" (page 105).
2006-09-05	Added a constant that indicates the application is running on an Intel-based Macintosh. See "System Architecture Selectors" (page 104).
2006-07-24	Added system version selectors.
2006-03-08	Added deprecation information.
2005-08-11	Updated description of Gestalt Manager version selector.
2005-07-07	Updated the System Version Selector with the values used for Mac OS X.
2005-04-29	Fixed several minor bugs.
2003-10-15	Added an ATSUI version constant and several ATSUI selector constants. See "ATSUI Version Selectors" (page 28) and "ATSUI Attribute Selectors" (page 24).
2002-12-10	Consolidated all Gestalt Manager constants into this document. Formerly, some of these constants were documented in the API that typically used the constants. Fixed formatting and edited for style.

Document Revision History

Index

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W