

Gestalt Manager

This chapter describes how 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 (such as a floating-point unit) or software (such as Color QuickDraw) 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.

The Gestalt Manager is available in system software versions 6.0.4 and later. The MPW software development system and some other development environments supply code that allows you to use the Gestalt Manager on earlier system software versions; check the documentation provided with your development system.

In system software versions earlier than 6.0.4, you can retrieve a limited description of the operating environment with the `SysEnviRons` function, also described in this chapter.

You need to read this chapter if you take advantage of specific hardware or software features that may not be present on all versions of the Macintosh, or if you wish to inform other software that your software is present in the operating environment.

This chapter describes how the Gestalt Manager works and then explains how you can

- determine whether the Gestalt Manager is available
- call the `Gestalt` function to investigate the operating environment
- make information about your own hardware or software available to other applications
- retrieve a limited description of the operating environment even if the Gestalt Manager is not available

About the Gestalt Manager

The Macintosh family of computers includes models that use a number of different processors, some accompanied by a floating-point unit (FPU) or memory management unit (MMU). Also, a single hardware configuration can have various versions of system software, drivers, and QuickDraw routines.

In general, applications should communicate with the system software and hardware through the available managers and device drivers. However, if your application takes advantage of hardware or software components that may not be present on all Macintosh computers, then you need some mechanism to determine whether those components are available.

The `Gestalt` function provides a simple, efficient way to determine the hardware and software configurations so your application can exploit as fully as possible whatever environment it is running in. When your application calls the `Gestalt` function, your application passes a *selector code* (or *selector*) as a parameter to specify the information it needs. Your application can call the `Gestalt` function to determine

Gestalt Manager

- the version and features of QuickDraw
- the versions and features of various other managers and drivers
- the type of floating-point unit (FPU), if any
- the type of memory management unit (MMU), if any
- the amount of available RAM
- the amount of available virtual memory
- the version of the A/UX operating system, if it's running
- the type of keyboard
- the model of computer
- the version number of the System file
- the type of central processing unit (CPU)

Your application can use the information returned by `Gestalt` in various ways. It might branch to alternate code, for example, depending on the version of QuickDraw, or cancel an operation and present an alert box if a critical but optional hardware component is unavailable.

Associated with the `Gestalt` function are two other functions—one that allows an application to register new features with the Gestalt Manager and another that allows an application to change the function used by `Gestalt` to retrieve a particular piece of information. These two functions make it easy for your software to announce its presence to other applications. A debugger, for example, can register itself with the Gestalt Manager during system initialization; afterward, debugging code in an application under development can call `Gestalt` to verify that the special routines provided by the debugger are available on the local machine. In this way, the Gestalt Manager can act as a central clearinghouse for information on the available software and hardware features of the operating environment and enhance cooperation and awareness among third-party products.

Although the `Gestalt` function can provide much of the information your application needs, you might still need to call some special-purpose routines supplied by various parts of the system software. To determine the resolution of the main Macintosh screen, for example, you call the `ScreenRes` procedure, described in the book *Inside Macintosh: Imaging with QuickDraw*.

The `Gestalt` function has replaced both the `SysEnviron`s function and the `Environ`s procedure. The `Gestalt` function is simpler to use and provides more information than either of those routines. Applications that use `SysEnviron`s execute correctly in system software versions 7.0 and later, in which `SysEnviron`s calls `Gestalt`.

The `SysEnviron`s function, introduced with the Macintosh SE and Macintosh II computers, fills in and returns a pointer to a *system environment record*, a data structure that describes some features of the operating environment. The `SysEnviron`s function cannot provide the detailed information supplied by `Gestalt`.

Like the `SysEnviron`s function, `Gestalt` can provide objective configuration information such as ROM version and size, but you should not infer the presence or

absence of particular hardware or software features from that information. When you need to know whether a feature is present, you should request information about it directly by using the appropriate selector code. (“Getting Information About the Operating Environment” beginning on page 1-6, lists the Apple-defined selector codes for Gestalt.)

Using the Gestalt Manager

The Gestalt Manager includes three functions—`Gestalt`, `NewGestalt`, and `ReplaceGestalt`. You can use the `Gestalt` function to get information about hardware or software components available on the current machine. You can use `NewGestalt` to register new software modules (such as drivers and patches) with the Gestalt Manager. You can use `ReplaceGestalt` to replace the function associated with a particular selector code.

Note

Most applications do not need to use either `NewGestalt` or `ReplaceGestalt`. ♦

If the Gestalt Manager is not present, you can get a brief description of the operating environment by calling the `SysEnviron`s function.

Determining Whether the Gestalt Manager Is Available

Versions 3.2 and later of MPW provide glue routines that allow you to call the Gestalt Manager functions even if they’re not in ROM or in the System file (that is, if your application is running under a system software version earlier than 6.0.4). In assembly language, however, and possibly in other development environments, you must verify that the Gestalt Manager is available before you use it.

You can verify that the `Gestalt` function is available by calling the function `NGetTrapAddress`, specifying the trap number of `Gestalt`, and comparing the result with the address of the code that is executed when you invoke an unimplemented instruction. If `Gestalt` is available, you can safely assume that `NewGestalt` and `ReplaceGestalt` are also available. For efficiency, you might want to define a global Boolean variable that you can set at the beginning of your program. Listing 1-1 illustrates a test that sets the variable `gHasGestalt`.

Listing 1-1 Determining whether Gestalt is available

```
gHasGestalt := MySWRoutineAvailable(_Gestalt);
```

For a sample definition of the application-defined function `MySWRoutineAvailable`, see the chapter “Trap Manager” later in this book.

Getting Information About the Operating Environment

When your application needs information about a software or hardware feature, it calls the `Gestalt` function, which has this interface:

```
FUNCTION Gestalt (selector: OSType; VAR response: LongInt): OSErr;
```

The first parameter is a selector code, which specifies the kind of information your application needs. You can use any of the Apple-defined selector codes listed later in this section and described in more detail in the section “Constants” beginning on page 1-14. You can also define and register your own selector codes using the `NewGestalt` function (as described in “Adding a New Selector Code” beginning on page 1-10), and you can use selector codes defined and registered by other applications.

If `Gestalt` can determine the requested information, it returns that information in the `response` parameter and returns a result code of `noErr`. If `Gestalt` cannot obtain the information, it returns a result code indicating the cause of the error; in that case, the value of the `response` parameter is undefined. You should *always* check the result code returned by `Gestalt` to make sure that the `response` parameter contains meaningful information.

Listing 1-2 illustrates an application-defined function that retrieves the sound attributes of the current operating environment. The application-defined `MyGetSoundAttr` function checks the function result returned by `Gestalt` and passes any calls with a nonzero result code to an error-handling routine.

Listing 1-2 Calling `Gestalt` and checking its result code

```
FUNCTION MyGetSoundAttr: LongInt;
VAR
    myErr: OSErr;
    myAttr: LongInt;
BEGIN
    IF gHasGestalt THEN
        BEGIN
            myErr := Gestalt(gestaltSoundAttr, myAttr);
            IF myErr <> noErr THEN {Gestalt failed}
                DoError(myErr)
            END
        ELSE
            myAttr := 0; {Gestalt not available}
        MyGetSoundAttr := myAttr;
    END;
```

You get different kinds of information from `Gestalt` by passing selectors from two kinds of Apple-defined selector codes:

Gestalt Manager

- *environmental selectors*, which return information your application can use to guide its actions
- *informational selectors*, which return information that cannot be used to determine whether a feature is available

It is particularly important that you understand the difference between environmental and informational selectors. The response returned by `Gestalt` when it is passed an informational selector is for your (or the user's) edification only; it should *never* be used by your application to determine whether a specific hardware or software feature is available. For example, you can use `Gestalt` to test for the version of the ROM installed on a particular machine. You can display this information to the user, but you should not infer from it anything about the actual software available. Routines you expect to be in ROM may actually be in RAM; hence, you cannot know that a routine usually found in ROM is not present simply because the ROM version predates the routine. Also, routines contained in ROM may have been patched by the system at startup time, in which case the system might not have the features you think it has on the basis of the reported ROM version. A Macintosh Plus with an old ROM, for example, could be running System 7. Similar remarks apply to other informational selectors, including ROM size, machine type, and System file version number.

To retrieve specific information about the hardware and software features available, you can use the following environmental selectors:

CONST

```

gestaltAddressingModeAttr = 'addr'; {addressing-mode attributes}
gestaltAliasMgrAttr      = 'alis'; {Alias Manager attributes}
gestaltAppleEventsAttr  = 'evnt'; {Apple events attributes}
gestaltAppleTalkVersion = 'atlk'; {old format AppleTalk version}
gestaltATalkVersion     = 'atkv'; {new format AppleTalk version}
gestaltAUXVersion       = 'a/ux'; {A/UX version, if present}
gestaltCFMAttr          = 'cfrg'; {Code Fragment Manager attributes}
gestaltCloseViewAttr   = 'BSDa'; {CloseView attributes}
gestaltComponentMgr    = 'cpnt'; {Component Manager version}
gestaltCompressionMgr  = 'icmp'; {Image Compression Manager version}
gestaltConnMgrAttr     = 'conn'; {Connection Manager attributes}
gestaltCRMAttr         = 'crm '; {Communication Resource Manager }
                        { attributes}
gestaltCTBVersion      = 'ctbv'; {Communication Toolbox version}
gestaltDBAccessMgrAttr = 'dbac'; {Data Access Manager attributes}
gestaltDictionaryMgrAttr = 'dict'; {Dictionary Manager attributes}
gestaltDisplayMgrAttr  = 'dply'; {Display Manager attributes}
gestaltDisplayMgrVers  = 'dplv'; {Display Manager version}
gestaltDITLExtAttr    = 'ditl'; {Dialog Manager extensions}
gestaltDragMgrAttr    = 'drag'; {Drag Manager attributes}
gestaltEasyAccessAttr  = 'easy'; {Easy Access attributes}
gestaltEditionMgrAttr  = 'edtn'; {Edition Manager attributes}

```

Gestalt Manager

```

gestaltExtToolboxTable = 'xttt'; {Toolbox trap dispatch table info}
gestaltFinderAttr     = 'fndr'; {Finder attributes}
gestaltFindFolderAttr = 'fold'; {FindFolder attributes}
gestaltFirstSlotNumber = 'slt1'; {first physical slot}
gestaltFontMgrAttr    = 'font'; {Font Manager attributes}
gestaltFPUType        = 'fpu '; {floating-point unit (FPU) type}
gestaltFSAttr         = 'fs  '; {file system attributes}
gestaltFXfrMgrAttr    = 'fxfr'; {File Transfer Manager attributes}
gestaltHelpMgrAttr    = 'help'; {Help Manager attributes}
gestaltIconUtilitiesAttr = 'icon'; {Icon Utilities attributes}
gestaltKeyboardType   = 'kbd '; {keyboard type code}
gestaltLogicalPageSize = 'pgsz'; {logical page size}
gestaltLogicalRAMSize = 'lram'; {logical RAM size}
gestaltLowMemorySize  = 'lmem'; {size of low memory}
gestaltMiscAttr       = 'misc'; {miscellaneous attributes}
gestaltMixedModeVersion = 'mixd'; {MixedMode version}
gestaltMMUType        = 'mmu '; {MMU type}
gestaltNativeCPUtype  = 'cput'; {native CPU type}
gestaltNotificationMgrAttr = 'nmgr'; {Notification Manager attributes}
gestaltNuBusConnectors = 'sltc'; {NuBus connector bitmap}
gestaltNuBusSlotCount = 'nubs'; {number of logical NuBus slots}
gestaltOSAttr         = 'os  '; {Operating System attributes}
gestaltOSTable        = 'ostt'; {base address of Operating System }
                        { trap dispatch table}

gestaltParityAttr     = 'prty'; {parity attributes}
gestaltPCXAttr        = 'pcxg'; {PC exchange attributes}
gestaltPhysicalRAMSize = 'ram '; {physical RAM size}
gestaltPopupAttr      = 'pop!'; {pop-up 'CDEF' attributes}
gestaltPowerMgrAttr   = 'powr'; {Power Manager attributes}
gestaltPPCToolboxAttr = 'ppc '; {Program-to-Program Communications }
                        { (PPC) Toolbox attributes}

gestaltProcessorType  = 'proc'; {microprocessor type code}
gestaltQuickdrawFeatures = 'qdrw'; {QuickDraw features}
gestaltQuickdrawVersion = 'qd  '; {QuickDraw version}
gestaltQuickTimeVersion = 'qtim'; {QuickTime version}
gestaltRealTimeMgrAttr = 'rtmr'; {Realtime Manager attributes}
gestaltResourceMgrAttr = 'rsrc'; {Resource Manager attributes}
gestaltScrapMgrAttr   = 'scra'; {Scrap Manager attributes}
gestaltScriptCount    = 'scr#'; {number of active script systems}
gestaltScriptMgrVersion = 'scri'; {Script Manager version}
gestaltSerialAttr     = 'ser  '; {serial hardware attributes}
gestaltSlotAttr       = 'slot'; {slot attributes}
gestaltSoundAttr      = 'snd  '; {sound attributes}

```

Gestalt Manager

```

gestaltSpeechAttr      = 'ttsc'; {Speech Manager attributes}
gestaltStandardFileAttr = 'stdf'; {Standard File attributes}
gestaltStdNBPAttr      = 'nlup'; {StandardNBP attributes}
gestaltSysArchitecture = 'sysa'; {Native System Architecture}
gestaltTEAttr          = 'teat'; {TextEdit attributes}
gestaltTermMgrAttr     = 'term'; {Terminal Manager attributes}
gestaltTextEditVersion = 'te  '; {TextEdit version code}
gestaltThreadMgrAttr   = 'thds'; {Thread Manager attributes}
gestaltTimeMgrVersion  = 'tmgr'; {Time Manager version code}
gestaltToolboxTable    = 'tbtt'; {base address of Toolbox trap }
                        { dispatch table}
gestaltTranslationAttr = 'xlat'; {Translation Manager attributes}
gestaltTSMgrVersion    = 'tsmv'; {Text Services Manager version}
gestaltVersion         = 'vers'; {Gestalt version}
gestaltVMAttr          = 'vm  '; {virtual memory attributes}

```

The informational selectors are provided for your or the user's information only. You can display the information returned from these selectors, but you should never use this information as an indication of what hardware or software features may be available. You can use the following informational selectors:

CONST

```

gestaltHardwareAttr    = 'hdwr'; {hardware attributes}
gestaltMachineIcon     = 'micn'; {machine 'ICON'/'cicn' resource ID}
gestaltMachineType     = 'mach'; {Macintosh model code}
gestaltROMSize         = 'rom  '; {ROM size}
gestaltROMVersion      = 'romv'; {ROM version}
gestaltSystemVersion   = 'sysv'; {System file version number}

```

For a description of the return values for these environmental and informational selectors, see the next section, "Interpreting Gestalt Responses," and the list of constants beginning on page 1-14.

Interpreting Gestalt Responses

The meaning of the value that Gestalt returns in the response parameter depends on the selector code with which it was called. For example, if you call Gestalt using the `gestaltTimeMgrVersion` selector, it returns a version code in the response parameter. In this case, a returned value of 3 indicates that the extended Time Manager is available.

In most cases, the last few characters in the selector's symbolic name form a suffix that indicates what type of value you can expect Gestalt to place in the response parameter. For example, if the suffix in a Gestalt selector is `Size`, then Gestalt returns a size in the response parameter. Table 1-1 lists the meaningful suffixes.

Gestalt Manager

Table 1-1 Gestalt selector suffixes and their meanings

Suffix	Returned value
Attr	A range of 32 bits, the meanings of which are defined by a list of constants. Bit 0 is the least significant bit of the long word.
Count	A number indicating how many of the indicated type of item exist.
Size	A size, usually in bytes.
Table	The base address of a table.
Type	An index to a list of feature descriptions.
Version	A version number, which can be either a constant with a defined meaning or an actual version number, usually stored as four hexadecimal digits in the low-order word of the return value. Implied decimal points may separate digits. The value \$0701, for example, returned in response to the <code>gestaltSystemVersion</code> selector, represents system software version 7.0.1.

Selectors that have the suffix `Attr` deserve special attention. They cause `Gestalt` to return a bit field that your application must interpret to determine whether a desired feature is present. For example, the application-defined sample function `MyGetSoundAttr`, defined in Listing 1-2 on page 1-6, returns a `LongInt` that contains the Sound Manager attributes field retrieved from `Gestalt`. To determine whether a particular feature is available, you need to look at the designated bit. The application-defined sample function `MyIsStereoMixing` in Listing 1-3, for example, determines whether stereo mixing is available.

Listing 1-3 Interpreting a Gestalt attributes response

```
FUNCTION MyIsStereoMixing: Boolean;
BEGIN
    MyIsStereoMixing := BTst(MyGetSoundAttr, gestaltStereoMixing);
END;
```

The `MyIsStereoMixing` function uses the MPW Pascal function `BTst` and the application-defined `MyGetSoundAttr` function to determine whether the stereo-mixing bit is set in the response value returned by `Gestalt` when it's called with the `gestaltSoundAttr` selector. The constant `gestaltStereoMixing` is defined in the header files.

Adding a New Selector Code

You can add your own selector code to those already understood by `Gestalt` by calling the `NewGestalt` function. Typically, a system extension registers itself with the Gestalt Manager so that applications that might use its services can find out whether it's there. A debugger, for example, could register its presence. Programmers working on an application could then embed instructions for the debugger in code under

development and call `Gestalt` to make sure the debugger is available before invoking those instructions.

The `NewGestalt` function requires two parameters: the new selector to be registered and the address of the associated *selector function*. `Gestalt` executes the selector function to determine what value to pass back when it's called with the new selector code.

The selector code is a four-character sequence of type `OSType`. If you have registered a creator string with Apple Computer, Inc., you are strongly encouraged to use that sequence as your selector code. The Pipeline debugger, for example, with a creator string of 'PIPE', would use a `Gestalt` selector code of 'PIPE'.

Note

Apple reserves for its own use all four-character sequences consisting solely of lowercase letters and nonalphanumeric ASCII characters. ♦

When you register your own selector code with the Gestalt Manager, you supply the address of the selector function to be executed when an application calls `Gestalt` with that code. Your selector function must reside in the system heap and must have the following interface:

```
FUNCTION MySelectorFunction (selector: OSType;
                             VAR response: LongInt): OSErr;
```

The `Gestalt` function passes its input parameters on to your selector function. Your function places the requested information in the `LongInt` pointed to by the `response` parameter and returns an error code, which `Gestalt` returns to its caller.

Your selector function should be as simple as possible. If your function needs to use global variables from the A5 world—that of your own software or that of some other software—it must explicitly set up A5 and then restore it upon exit. (See *Inside Macintosh: Memory* for an explanation of setting up and restoring the A5 world.)

Your selector function can, if necessary, call `Gestalt` and pass it other selector codes. Note that the `response` parameter is merely the address into which your function places the information requested. You cannot use that parameter to pass information to your selector function.

Listing 1-4 illustrates a minimal selector function that sets the `response` parameter and returns an error code of `noErr`. The application-defined sample function, `MyGestaltPipe`, is isolated in a `UNIT` element for separate compilation and placement in a resource.

Listing 1-4 Defining a simple `Gestalt` selector function

```
UNIT GestaltFunc;
INTERFACE
    USES OSIntf;
    FUNCTION MyGestaltPipe (gestaltSelector: OSType;
```

Gestalt Manager

```

                                VAR gestaltReply: LongInt): OSErr;
IMPLEMENTATION
    FUNCTION MyGestaltPipe;
    BEGIN
        gestaltReply := $ACE;      {reply defined by Pipeline}
        MyGestaltPipe := noErr;    {too simple for errors}
    END;
END.

```

This sample linking command places the compiled code in resource ID 128 of a type arbitrarily named 'GDEF'.

```
Link GestaltFunc.p.o -rn -rt GDEF=128 -o Pipeline
```

To add a Gestalt selector code, you first move the selector function into the system heap and then call the `NewGestalt` function, which adds the selector code and its function to the Gestalt repertoire.

▲ **WARNING**

Take special care when accessing memory in the system heap; it persists even after your application terminates. ▲

Listing 1-5 illustrates the installation of a new Gestalt selector.

Listing 1-5 Installing a new Gestalt selector

```

PROCEDURE MyInstallGestFunc;
VAR
    gestFuncHandle:   Handle;
    gestFuncSize:     Size;
    gestSysPtr:       Ptr;
    myErr:            OSErr;
BEGIN
    gestFuncHandle := GetResource('GDEF', 128);
    IF ResError = noErr THEN
    BEGIN
        gestFuncSize := SizeResource(gestFuncHandle);
        gestSysPtr := NewPtrSys(gestFuncSize);
        IF MemError = noErr THEN
        BEGIN
            BlockMove(gestFuncHandle^, gestSysPtr, gestFuncSize);
            FlushInstructionCache;
            myErr := NewGestalt('PIPE',
                               SelectorFunctionUUP(gestSysPtr));
        END;
    END;
END;

```

Gestalt Manager

```

        ReleaseResource (gestFuncHandle);
    END;
END;

```

The application-defined sample procedure `MyInstallGestFunc` loads the resource and then gets its size so it can allocate a pointer in the system heap. It then copies the resource to the pointer and releases the resource.

▲ **WARNING**

Be sure to call the `FlushInstructionCache` procedure every time you modify code in RAM. See the chapter “Memory Management Utilities” in *Inside Macintosh: Memory* for details about `FlushInstructionCache`. ▲

Finally, `MyInstallGestFunc` calls `NewGestalt` to register the selector code 'PIPE' and its selector function with the Gestalt Manager.

Because the new selector function resides in the system heap, Gestalt recognizes and responds to the new selector until the machine restarts, even if your software terminates before that time. You might therefore want your selector function to determine whether your software is still running before filling in the response value. The simplest way to report that your application is not available is to return an error code.

If you attempt to add a selector code that Gestalt already recognizes, `NewGestalt` returns the error code `gestaltDupSelectorErr`.

Modifying a Selector Function

You can use the `ReplaceGestalt` function to modify the function that Gestalt executes when passed a particular selector code. Your replacement selector function must reside in the system heap and must conform to the interface defined in the previous section, “Adding a New Selector Code.”

To allow the new function to call the function it’s replacing, `ReplaceGestalt` returns the address of the previous function.

If you attempt to redefine a selector that is not yet defined, `ReplaceGestalt` returns an error code; in that case, the address of the previous function is undefined. Always test the result code of `ReplaceGestalt` before calling Gestalt with that selector or attempting to use the response parameter.

Note

If you modify the function associated with an existing Gestalt selector, do not use any bits in the response parameter that are not documented in this chapter. Apple reserves all undocumented bits in the response parameters returned by Apple-defined Gestalt selectors. ◆

Because `ReplaceGestalt` supplies the address of the function it’s replacing, you can use it to retrieve the address of the selector function associated with a selector code.

Getting Environmental Information Without the Gestalt Manager

You can call the `SysEnviron`s function, which predates the Gestalt Manager, to get a brief description of the operating environment. The `SysEnviron`s function is available on all models of the Macintosh computer since the Macintosh SE and Macintosh II.

Note

The `SysEnviron`s function is not part of the Gestalt Manager, but is documented in this chapter for the sake of completeness. ♦

The `SysEnviron`s function fills in a record that contains the model of the machine, the System file version number, the microprocessor type, a keyboard type code, and Boolean indicators of whether the machine has a floating-point unit or Color QuickDraw. The system environment record includes one detail not available through `Gestalt`: the working directory reference number of the folder or volume that holds the System file (although that information is available through the `FindFolder` function). See “The System Environment Record” beginning on page 1-28 for a complete description of the system environment record.

Gestalt Manager Reference

This section lists the Gestalt selector codes and their defined return values and describes the system environment record, the three Gestalt Manager functions, and the `SysEnviron`s function.

Constants

This section lists the Apple-defined Gestalt Manager selector codes, describes the formats of their responses, and lists the constants defined for their return values.

You pass a selector code when you call `Gestalt` to specify the kind of information you need. Apple defines two distinct kinds of selector codes: environmental selectors, which supply information you can use to control the behavior of your application, and informational selectors, which supply information you can't use to determine what hardware or software features are available.

The selector code constants use a set of suffixes that indicate what format the response value will take. Selectors with the suffix `Attr`, for example, return a 32-bit response value in which the individual bits represent specific attributes. The constants listed for these response values represent bit numbers. For a more general description of selectors and their response values, see “Interpreting Gestalt Responses” beginning on page 1-9.

Gestalt Manager

The `Gestalt` function accepts the following environmental selectors.

Selector	Response bits and response values
<code>gestaltAddressingModeAttr</code>	<p>Current addressing-mode attributes.</p> <pre>CONST gestalt32BitAddressing = 0; gestalt32BitSysZone = 1; gestalt32BitCapable = 2;</pre> <p>The <code>gestalt32BitAddressing</code> attribute indicates that the machine started up with 32-bit addressing. The <code>gestalt32BitSysZone</code> attribute indicates that the system heap has 32-bit clean block headers (regardless of the type of addressing the machine started up in). See the book <i>Inside Macintosh: Memory</i> for more information about 32-bit addressing.</p>
<code>gestaltAliasMgrAttr</code>	<p>Alias Manager attributes.</p> <pre>CONST gestaltAliasMgrPresent = 0; gestaltAliasMgrSupportsRemoteAppleTalk = 1;</pre>
<code>gestaltAppleEventsAttr</code>	<p>The Apple events attribute.</p> <pre>CONST gestaltAppleEventsPresent = 0; gestaltScriptingSupport = 1; gestaltOSLInSystem = 2;</pre>
<code>gestaltAppleTalkVersion</code>	<p>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.</p>
<code>gestaltATalkVersion</code>	<p>The version number of the AppleTalk driver, in the format introduced with AppleTalk version 56. (For a description of AppleTalk, see <i>Inside AppleTalk</i>, second edition.) The version is stored in the high 3 bytes of the return value.</p> <p>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.</p>

Gestalt Manager

Selector

gestaltATalkVersion
(continued)

gestaltAUXVersion

gestaltCFMAttr

gestaltCloseViewAttr

gestaltComponentMgr

gestaltCompressionMgr

gestaltConnMgrAttr

gestaltCRMAttr

gestaltCTBVersion

gestaltDBAccessMgrAttr

Response bits and response values

```
CONST
development = $20;
alpha       = $40;
beta        = $60;
final       = $80;
release     = $80;
```

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

Byte 0 always contains 0.

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, Gestalt returns gestaltUnknownErr.

Code Fragment Manager attributes.

```
CONST
gestaltCFMPresent = 0;
```

The CloseView attributes

```
CONST
gestaltCloseViewEnabled = 0;
gestaltCloseViewDisplayMgrFriendly = 1;
```

The version of the Component Manager.

The version of the Image Compression Manager.

Connection Manager attributes.

```
CONST
gestaltConnMgrPresent = 0;
gestaltConnMgrCMSearchFix = 1;
gestaltConnMgrErrorString = 2;
gestaltConnMgrMultiAsyncIO = 3;
```

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

Communications Resource Manager attributes.

```
CONST
gestaltCRMPresent = 0;
gestaltCRMPersistentFix = 1;
gestaltCRMToolRsrcCalls = 2;
```

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

The Data Access Manager attribute.

```
CONST
gestaltDBAccessMgrPresent = 0;
```

Gestalt Manager

Selector

gestaltDictionaryMgrAttr

gestaltDisplayMgrAttr

gestaltDITLExtAttr

gestaltDragMgrAttr

gestaltEasyAccessAttr

gestaltEditionMgrAttr

gestaltExtToolboxTable

gestaltFinderAttr

Response bits and response values

The Dictionary Manager attributes.

```
CONST
    gestaltDictionaryMgrPresent = 0;
```

The Display Manager attributes.

```
CONST
    gestaltDisplayMgrPresent = 0;
```

The Dialog Manager extensions attributes.

```
CONST
    gestaltDITLExtPresent = 0;
```

If this flag bit is `TRUE`, then the Dialog Manager extensions included in System 7 are available. See the book *Inside Macintosh: Macintosh Toolbox Essentials* for details about the Dialog Manager.

Drag Manager attributes.

```
CONST
    gestaltDragMgrPresent = 0;
```

Easy Access attributes.

```
CONST
    gestaltEasyAccessOff = 0;
    gestaltEasyAccessOn = 1;
    gestaltEasyAccessSticky = 2;
    gestaltEasyAccessLocked = 3;
```

Edition Manager attributes.

```
CONST
    gestaltEditionMgrPresent = 0;
    gestaltEditionMgrTranslationAware = 1;
```

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

Finder attributes.

```
CONST
    gestaltFinderDropEvent = 0;
    gestaltFinderMagicPlacement = 1;
    gestaltFinderCallsAEProcess = 2;
    gestaltOSLCompliantFinder = 3;
    gestaltFinderSupports4GBVolumes = 4;
    gestaltFinderHandlesCFMFailures = 5;
    gestaltFinderHasClippings = 6;
```

Gestalt Manager

Selector

gestaltFindFolderAttr

gestaltFirstSlotNumber

gestaltFontMgrAttr

gestaltFPUType

gestaltFSAttr

gestaltFXfrMgrAttr

gestaltHelpMgrAttr

gestaltIconUtilitiesAttr

gestaltKeyboardType

Response bits and response values

The FindFolder function attribute.

CONST

gestaltFindFolderPresent = 0;

The first physical slot.

The Font Manager attribute.

CONST

gestaltOutlineFonts = 0;

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

CONST

gestaltNoFPU = 0;

gestalt68881 = 1;

gestalt68882 = 2;

gestalt68040FPU = 3;

File system attributes.

CONST

gestaltFullExtFSDispatching = 0;

gestaltHasFSSpecCalls = 1;

gestaltHasFileSystemManager = 2;

gestaltFSMDoesDynamicLoad = 3;

gestaltFSSupports4GBVols = 4;

gestaltHasExtendedDiskInit = 6;

The File Transfer Manager attributes.

CONST

gestaltFXfrMgrPresent = 0;

gestaltFXfrMgrMultiFile = 1;

gestaltFXfrMgrErrorString = 2;

The Help Manager attribute.

CONST

gestaltHelpMgrPresent = 0;

The Icon Utilities attribute.

CONST

gestaltIconUtilitiesPresent = 0;

A constant that represents the type of keyboard.

CONST

gestaltMacKbd = 1;

gestaltMacAndPad = 2;

gestaltMacPlusKbd = 3;

gestaltExtADBKbd = 4;

gestaltStdADBKbd = 5;

gestaltPrtblADBKbd = 6;

gestaltPrtblISOKbd = 7;

Selector

`gestaltKeyboardType`
(continued)

Response bits and response values

```

gestaltStdISOADBKbd      = 8;
gestaltExtISOADBKbd     = 9;
gestaltADBKbdII         = 10;
gestaltADBIKbdII        = 11;
gestaltPwrBookADBKbd    = 12;
gestaltPwrBookISOADBKbd = 13;
gestaltAppleAdjustKeypad = 14;
gestaltAppleAdjustADBKbd = 15;
gestaltAppleAdjustISOKbd = 16;

```

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, described in *Inside Macintosh: Devices*. 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`, documented in *Inside Macintosh: Devices*.

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

`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, Gestalt returns an error when called with this selector.

`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.

`gestaltLowMemorySize`

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

Gestalt Manager

Selector

gestaltMiscAttr

gestaltMixedModeVersion

gestaltMMUType

gestaltNativeCPUtype

gestaltNotificationMgrAttr

gestaltNuBusConnectors

gestaltOSAttr

Response bits and response values

Information about miscellaneous pieces of the Operating System or hardware configuration.

CONST

```
gestaltScrollingThrottle = 0;
gestaltSquareMenuBar    = 2;
```

The version of Mixed Mode Manager.

A constant that represents the type of MMU currently installed.

CONST

```
gestaltNoMMU      = 0;
gestaltAMU        = 1;
gestalt68851      = 2;
gestalt68030MMU  = 3;
gestalt68040MMU  = 4;
gestaltEMMU1     = 5;
```

Native CPU type.

CONST

```
gestaltCPU68000 = $000;
gestaltCPU68010 = $001;
gestaltCPU68020 = $002;
gestaltCPU68030 = $003;
gestaltCPU68040 = $004;

gestaltCPU601   = $101;
```

Note, to check whether the native system architecture is a MC680x0 or a PowerPC microprocessor, use the `gestaltSysArchitecture` selector.

The Notification Manager attribute.

CONST

```
gestaltNotificationPresent = 0;
```

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.

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:

Gestalt Manager

Selector

gestaltOSAttr
(continued)

gestaltOSTable

gestaltParityAttr

gestaltPCXAttr

gestaltPhysicalRAMSize

gestaltPopupAttr

gestaltPowerMgrAttr

gestaltPPCToolboxAttr

Response bits and response values

```
CONST
    gestaltSysZoneGrowable      = 0;
    gestaltLaunchCanReturn      = 1;
    gestaltLaunchFullFileSpec   = 2;
    gestaltLaunchControl        = 3;
    gestaltTempMemSupport       = 4;
    gestaltRealTempMemory       = 5;
    gestaltTempMemTracked       = 6;
```

See the book *Inside Macintosh: Memory* for a full explanation of the temporary memory features, and see the book *Inside Macintosh: Processes* for a full explanation of the launch control features.

The base address of the Operating System trap dispatch table.

Information about the machine's parity-checking features.

```
CONST
    gestaltHasParityCapability = 0;
    gestaltParityEnabled      = 1;
```

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

PC Exchange attributes.

```
CONST
    gestaltPCXHas8and16BitFAT = 0;
    gestaltPCXHasProDOS       = 1;
```

The number of bytes of physical RAM currently installed.

The attribute of the pop-up control definition.

```
CONST gestaltPopupPresent = 0;
```

Power Manager attributes.

```
CONST
    gestaltPMgrExists      = 0;
    gestaltPMgrCPUIdle     = 1;
    gestaltPMgrSCC         = 2;
    gestaltPMgrSound       = 3;
    gestaltPMgrDispatchExists = 4;
```

Program-to-Program Communication (PPC) Toolbox attributes. Note that these constants are defined as masks, not bit numbers.

```
CONST
    gestaltPPCToolboxPresent = $0000;
    gestaltPPCSupportsRealTime = $1000;
    gestaltPPCSupportsIncoming = $0001;
    gestaltPPCSupportsOutgoing = $0002;
```

Gestalt Manager

Selector

gestaltProcessorType

Response bits and response values

A constant that represents the type of microprocessor currently running.

```
CONST
    gestalt68000 = 1;
    gestalt68010 = 2;
    gestalt68020 = 3;
    gestalt68030 = 4;
    gestalt68040 = 5;
```

gestaltQuickdrawFeatures

QuickDraw features.

```
CONST
    gestaltHasColor           = 0;
    gestaltHasDeepGWorlds    = 1;
    gestaltHasDirectPixMaps  = 2;
    gestaltHasGrayishTextOr  = 3;
    gestaltSupportsMirroring  = 4;
```

gestaltQuickdrawVersion

The version of QuickDraw, 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.

```
CONST
    gestaltOriginalQD   = $000;
    gestalt8BitQD       = $100;
    gestalt32BitQD      = $200;
    gestalt32BitQD11    = $210;
    gestalt32BitQD12    = $220;
    gestalt32BitQD13    = $230;
```

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 routines to obtain that information.

gestaltQuickTimeVersion

The QuickTime version.

gestaltRealtimeMgrAttr

Realtime Manager attributes.

```
CONST
    gestaltRealtimeMgrPresent = 0;
```

gestaltResourceMgrAttr

The Resource Manager attribute.

```
CONST
    gestaltPartialRsrcs = 0;
```

Gestalt Manager

Selector

gestaltScrapMgrAttr

gestaltScriptCount

gestaltScriptMgrVersion

gestaltSerialAttr

gestaltSlotAttr

gestaltSoundAttr

Response bits and response values

Scrap Manager attributes.

```

CONST
    gestaltScrapMgrTranslationAware
        = 0;
    gestaltTranslationMgrHintOrder
        = 1;

```

The number of script systems currently active.

The version number of the Script Manager (in the low-order word of the return value).

Serial hardware attributes of the machine, such as whether or not the GPIa line is connected and can be used for external clocking.

```

CONST
    gestaltHasGPIaToDCDa = 0;
    gestaltHasGPIaToRTxCa = 1;
    gestaltHasGPIaToDCDb = 2;

```

Slot Manager attributes.

```

CONST
    gestaltSlotMgrExists = 0;
    gestaltNuBusPresent = 1;
    gestaltSESlotPresent = 2;
    gestaltSE30SlotPresent = 3;
    gestaltPortableSlotPresent = 4;

```

Sound attributes.

```

CONST
    gestaltStereoCapability = 0;
    gestaltStereoMixing = 1;
    gestaltSoundIOMgrPresent = 3;
    gestaltBuiltInSoundInput = 4;
    gestaltHasSoundInputDevice = 5;
    gestaltPlayAndRecord = 6;
    gestalt16BitSoundIO = 7;
    gestaltStereoInput = 8;
    gestaltLineLevelInput = 9;
    gestaltSndPlayDoubleBuffer = 10;
    gestaltMultiChannels = 11;
    gestalt16BitAudioSupport = 12;

```

Gestalt Manager

Selector

gestaltSoundAttr
(continued)

gestaltSpeechAttr

gestaltStandardFileAttr

gestaltStdNBPAAttr

gestaltSysArchitecture

Response bits and response values

If the bit `gestaltStereoCapability` is `TRUE`, the available hardware can play stereo sounds. The bit `gestaltStereoMixing` indicates that the sound hardware of the machine mixes both left and right channels of stereo sound into a single audio signal for the internal speaker. The `gestaltSoundIOMgrPresent` bit indicates that the new sound input routines are available, and the `gestaltBuiltInSoundInput` bit indicates that a built-in sound input device is available. The `gestaltHasSoundInputDevice` bit indicates that some sound input device is available.

Note, bits 7 through 12 are not defined for versions of the Sound Manager prior to version 3.0.

Speech Manager attributes.

```
CONST
    gestaltSpeechMgrPresent      = 0;
    gestaltSpeechHasPPCGLue     = 1;
```

Standard File Package attributes.

```
CONST
    gestaltStandardFile58      = 0;
    gestaltStandardFileTranslationAware
                                = 1;
    gestaltStandardFileHasColorIcons
                                = 2;
```

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.)

Information about the `StandardNBP` (Name-Binding Protocol) function.

```
CONST
    gestaltStdNBPPresent      = 0;
```

The native system architecture.

```
CONST
    gestalt68k                = 1;
    gestaltPowerPC            = 2;
```

If the `gestalt68k` flag bit is set, the native microprocessor is a MC680x0 chip. If the `gestaltPowerPC` flag bit is set, the native microprocessor is a PowerPC chip.

Gestalt Manager

Selector

gestaltTEAttr

gestaltTermMgrAttr

gestaltTextEditVersion

gestaltThreadMgrAtt

gestaltTimeMgrVersion

gestaltToolboxTable

gestaltTranslationAttr

gestaltTSMgrVersion

gestaltVersion

gestaltVMAttr

Response bits and response values

TextEdit attributes.

```
CONST
    gestaltTEHasGetHiliteRgn = 0;
```

Terminal Manager attributes.

```
CONST
    gestaltTermMgrPresent      = 0;
    gestaltTermMgrErrorString = 2;
```

A constant that indicates which version of TextEdit is present.

```
CONST
    gestaltTE1 = 1;
    gestaltTE2 = 2;
    gestaltTE3 = 3;
    gestaltTE4 = 4;
    gestaltTE5 = 5;
```

Thread Manager attributes.

```
CONST
    gestaltThreadMgrPresent      = 0;
    gestaltSpecificMatchSupport = 1;
```

A constant that indicates which version of the Time Manager is present.

```
CONST
    gestaltStandardTimeMgr = 1;
    gestaltRevisedTimeMgr  = 2;
    gestaltExtendedTimeMgr = 3;
```

The base address of the Toolbox trap dispatch table.

The Translation Manager attributes.

```
CONST
    gestaltTranslationMgrExists = 0;
```

The version of the Text Services.

The version of the Gestalt Manager (in the low-order word of the return value). The current version is 1, corresponding to a returned value of \$0001.

The virtual memory attributes.

```
CONST
    gestaltVMPresent      = 0;
```

Gestalt Manager

The `Gestalt` function also accepts the following informational selectors.

▲ **WARNING**

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

Selector	Meaning
<code>gestaltHardwareAttr</code>	<p>Low-level hardware configuration attributes.</p> <pre> CONST gestaltHasVIA1 = 0; gestaltHasVIA2 = 1; gestaltHasASC = 3; gestaltHasSCC = 4; gestaltHasSCSI = 7; gestaltHasSoftPowerOff = 19; gestaltHasSCSI961 = 21; gestaltHasSCSI962 = 22; gestaltHasUniversalROM = 24; </pre> <p>The <code>gestaltHasSCSI</code> 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. Those computers set the <code>gestaltHasSCSI961</code> or <code>gestaltHasSCSI962</code> bit to report a SCSI implementation based on the 53C96 chip installed on an internal or external bus, respectively.</p> <p>The <code>gestaltHasSCC</code> bit is normally returned as 0 on the Macintosh IIx 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, <code>gestaltHasSCC</code> is set.</p>
<code>gestaltMachineIcon</code>	The icon family resource ID for the current type of Macintosh.
<code>gestaltMachineType</code>	<p>A constant that indicates the model of computer.</p> <pre> CONST gestaltClassic = 1; gestaltMacXL = 2; gestaltMac512KE = 3; gestaltMacPlus = 4; gestaltMacSE = 5; gestaltMacII = 6; gestaltMacIIx = 7; gestaltMacIIcx = 8; gestaltMacSE030 = 9; gestaltPortable = 10; </pre>

Gestalt Manager

Selector	Meaning
gestaltMachineType (continued)	gestaltMacIici = 11;
	gestaltMacIifx = 13;
	gestaltMacClassic = 17;
	gestaltMacIisi = 18;
	gestaltMacLC = 19;
	gestaltQuadra900 = 20;
	gestaltPowerBook170 = 21;
	gestaltQuadra700 = 22;
	gestaltClassicII = 23;
	gestaltPowerBook100 = 24;
	gestaltPowerBook140 = 25;
	gestaltQuadra950 = 26;
	gestaltMacLcIII = 27;
	gestaltPowerBookDuo210 = 29;
	gestaltMacCentris650 = 30;
	gestaltPowerBookDuo230 = 32;
	gestaltPowerBook180 = 33;
	gestaltPowerBook160 = 34;
	gestaltMacQuadra800 = 35;
	gestaltMacLcII = 37;
	gestaltPowerBookDuo250 = 38;
	gestaltMacIivi = 44;
	gestaltPerforma600 = 45;
	gestaltMacIivx = 48;
	gestaltMacColorClassic = 49;
	gestaltPowerBook165c = 50;
	gestaltMacCentris610 = 52;
	gestaltMacQuadra610 = 53;
	gestaltPowerBook145 = 54;
	gestaltMacLc520 = 56;
	gestaltMacCentris660AV = 60;
	gestaltPowerBook180c = 71;
	gestaltPowerBookDuo270c = 77;
	gestaltMacQuadra840AV = 78;
	gestaltPowerBook165 = 84;
	gestaltMacTV = 88;
	gestaltMacLc475 = 89;
	gestaltMacLc575 = 92;
	gestaltMacQuadra605 = 94;
	gestaltPowerMac8100_80 = 65;
	gestaltPowerMac6100_60 = 75;
	gestaltPowerMac7100_66 = 112;

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`.

```
CONST
    kMachineNameStrID = -16395;
```

Gestalt Manager

Selector	Meaning
<code>gestaltROMSize</code>	The size of the installed ROM, in bytes. The value is returned in only one word.
<code>gestaltROMVersion</code>	The version number of the installed ROM (in the low-order word of the return value).
<code>gestaltSystemVersion</code>	The version number of the currently active System file, 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 <code>Gestalt</code> returns the value \$0701. Ignore the high-order word of the returned value.

Data Structures

This section describes the record filled in by the `SysEnviron`s function.

The System Environment Record

The `SysEnviron`s function fills in a system environment record, which describes some aspects of the software and hardware environment.

```

TYPE SysEnvRec =
  RECORD
    environsVersion: Integer;
    machineType: Integer;
    systemVersion: Integer;
    processor: Integer;
    hasFPU: Boolean;
    hasColorQD: Boolean;
    keyBoardType: Integer;
    atDrvrVersNum: Integer;
    sysVRefNum: Integer;
  END;

```

FIELD DESCRIPTIONS

`environsVersion`

The version number of the `SysEnviron`s function that was used to fill in the record.

When you call `SysEnviron`s, you specify a version number to ensure that you receive a system environment record that matches your expectations, as explained in the description of `SysEnviron`s beginning on page 1-32. If you request a more recent version of `SysEnviron`s than is available, `SysEnviron`s places its own version number in the `environsVersion` field and returns a function result `envVersTooBig`.

Gestalt Manager

`machineType` A code for the Macintosh model, which can be one of these values:

```
CONST
    envXL                = -2; {Macintosh XL}
    envMac               = -1; {Macintosh with 64K }
                        { ROM}
    envMachUnknown      =  0; {unknown model, }
                        { after Macintosh }
                        { IIfx}
    env512KE             =  1; {Macintosh 512K }
                        { enhanced}
    envMacPlus           =  2; {Macintosh Plus}
    envSE                =  3; {Macintosh SE}
    envMacII             =  4; {Macintosh II}
    envMacIIX            =  5; {Macintosh IIX}
    envMacIICx           =  6; {Macintosh IICx}
    envSE30              =  7; {Macintosh SE30}
    envPortable          =  8; {Macintosh Portable}
    envMacIICi           =  9; {Macintosh IICi}
    envMacIIfx           = 11; {Macintosh IIfx}
```

Note

Use Gestalt to obtain information about machine types not listed above. ♦

`systemVersion` The version number of the current System file, represented as two byte-long numbers with one or more implied decimal points. The value \$0410, for example, represents system software version 4.1. If you call `SysEnviron`s when a system earlier than 4.1 is running, the MPW glue places \$0 in this field and returns a result code of `envNotPresent`.

`processor` A code for the microprocessor, which can be one of these values:

```
CONST
    envCPUUnknown       =  0; {unknown }
                        { microprocessor}
    env68000             =  1; {MC68000}
    env68010             =  2; {MC68010}
    env68020             =  3; {MC68020}
    env68030             =  4; {MC68030}
    env68040             =  5; {MC68040}
```

`hasFPU` A Boolean value that indicates whether hardware floating-point processing is available.

`hasColorQD` A Boolean value that indicates whether Color QuickDraw is present. This field says nothing about the presence of a color monitor.

Gestalt Manager

`keyboardType` A code for the keyboard type, which can be one of these values:

```
CONST
    envUnknownKbd      = 0;  {Macintosh Plus with }
                          { keypad}
    envMacKbd          = 1;  {Macintosh}
    envMacAndPad       = 2;  {Macintosh with keypad}
    envMacPlusKbd      = 3;  {Macintosh Plus}
    envAExtendKbd      = 4;  {Apple extended}
    envStandADBKbd     = 5;  {standard ADB}
    envPrtblADBKbd     = 6;  {Macintosh Portable ADB}
    envPrtblISOKbd     = 7;  {Macintosh Portable ISO}
    envStdISOADBKbd    = 8;  {standard ISO ADB}
    envExtISOADBKbd    = 9;  {extended ISO ADB}
```

Note

Use Gestalt to obtain information about keyboard types not listed above. ♦

If the Apple Desktop Bus is in use, this field returns the keyboard type of the keyboard on which the last keystroke was made.

`atDrvrVersNum` The version number of the AppleTalk driver (specifically, the .MPP driver) currently installed. If AppleTalk is not loaded, this field is 0.

`sysVRefNum` The working-directory reference number of the folder or volume that holds the open System file.

Gestalt Manager Routines

This section describes the three Gestalt Manager functions, `Gestalt`, `NewGestalt`, and `ReplaceGestalt`. It also describes the `SysEnviron` function, which can give you a brief description of the operating environment when `Gestalt` is not available. The Gestalt Manager functions allow you to

- find out what hardware and software features are present
- add new selectors to those understood by the `Gestalt` function
- replace the functions associated with known selectors

Getting Information About the Operating Environment

This section describes both the `Gestalt` function, which you use to find out about the operating environment, and the `SysEnviron` function, which you use only when `Gestalt` is not available.

Gestalt

You can use the `Gestalt` function to obtain information about the operating environment. You specify what information you need by passing one of the selector codes recognized by `Gestalt`.

```
FUNCTION Gestalt (selector: OSType; VAR response: LongInt): OSErr;
```

`selector` The selector code for the information you need.

`response` On exit, the requested information whose format depends on the selector code specified in the selector parameter.

DESCRIPTION

The `Gestalt` function places the information requested by the `selector` parameter in the variable parameter `response`. Note that `Gestalt` 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. Although the `response` parameter is declared as a variable parameter, you cannot use it to pass information to `Gestalt` or to a `Gestalt` selector function. `Gestalt` interprets the `response` parameter as an address at which it is to place the result returned by the selector function specified by the `selector` parameter. `Gestalt` ignores any information already at that address.

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 supply information you can't use to determine what hardware or software features are available. You can use one of the selector codes defined by Apple (listed in the "Constants" section beginning on page 1-14) 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 at any time, 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.

Gestalt Manager

ASSEMBLY-LANGUAGE INFORMATION

The registers on entry and exit for the `Gestalt` function are

Registers on entry

D0 Selector code

Registers on exit

A Response

0

D0 Result
code

RESULT CODES

<code>noErr</code>	0	No error
<code>gestaltUnknownErr</code>	-5550	Could not obtain the response
<code>gestaltUndefSelectorErr</code>	-5551	Undefined selector

SEE ALSO

See the documentation of the features you're interested in for more information on the various response values and their meanings.

See "Interpreting Gestalt Responses" beginning on page 1-9 for a discussion of the different response value formats and a sample function that checks an attributes value for a specific feature.

See "Getting Information About the Operating Environment" beginning on page 1-6 for a sample function that calls the `Gestalt` function and checks the validity of the return value. See the "Constants" section beginning on page 1-14 for a list of selector codes defined by Apple and the formats of their responses.

***SysEnviron**s*

You can use the `SysEnviron`s function when you need information about the operating environment and the `Gestalt` function is not available.

```
FUNCTION SysEnviron (versionRequested: Integer;
                    VAR theWorld: SysEnvRec): OSErr;
```

`versionRequested`

The version number of `SysEnviron`s you expect.

`theWorld` A system environment record.

DESCRIPTION

The `SysEnviron`s function fills in a system environment record identified by the variable parameter `theWorld`. It returns a result code.

You use the `versionRequested` parameter to tell `SysEnviron`s which version of the system environment record you're prepared to receive. This chapter documents version 2, which contains the same fields as version 1 but recognizes a more complete set of descriptive constants. Apple will raise the `SysEnviron`s version number in the future only if the record structure changes. You can trust any future revision to return the version 2 record if you request it, although the record might contain whatever constants are then current. To request the most recent version, you can use the constant `curSysEnvVers`:

```
CONST
    curSysEnvVers = 2;
```

ASSEMBLY-LANGUAGE INFORMATION

The registers on entry and exit for the `SysEnviron`s function are

Registers on entry

A Address of a system environment record
0
D0 Version requested

Registers on exit

A Address of a system environment record
0
D0 Result code

RESULT CODES

<code>noErr</code>	0	No error
<code>envNotPresent</code>	-5500	<code>SysEnviron</code> s trap not present
<code>envBadVers</code>	-5501	Nonpositive version number passed
<code>envVersTooBig</code>	-5502	Requested version of <code>SysEnviron</code> s not available

SEE ALSO

See "The System Environment Record" beginning on page 1-28 for a detailed description of the system environment record.

Adding a Selector Code

You can add your own selector code using the `NewGestalt` function.

NewGestalt

You can use the `NewGestalt` function to add a selector code to those already recognized by `Gestalt`.

```
FUNCTION NewGestalt (selector: OSType;
                   gestaltFunction: SelectorFunctionUUP)
                   : OSErr;
```

`selector` The selector code you're adding, which is a four-character sequence of type `OSType`.

`gestaltFunction` A pointer to the selector function that `Gestalt` executes when it receives the new selector code.

DESCRIPTION

The `NewGestalt` function registers a specified selector code with the Gestalt Manager so that when `Gestalt` is called with that selector code, the specified selector function is executed. The function result of `NewGestalt` is a result code.

Before calling `NewGestalt`, you must define a selector function and install it in the system heap. The selector function must conform to the interface defined in "Adding a New Selector Code" beginning on page 1-10.

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

The `NewGestalt` function might move memory and should not be called at interrupt time.

ASSEMBLY-LANGUAGE INFORMATION

The registers on entry and exit for the `NewGestalt` function are

Registers on entry

A Address of new selector function
0
D0 Selector code

Registers on exit

D0 Result
 code

RESULT CODES

noErr	0	No error
memFullErr	-108	Ran out of memory
gestaltDupSelectorErr	-5552	Selector already exists
gestaltLocationErr	-5553	Function not in system heap

SEE ALSO

See “Adding a New Selector Code” beginning on page 1-10 for a sample selector function and a sample procedure that installs it. For information about the `Gestalt` function, see page 1-31.

Modifying a Selector Function

You can install your own selector function for an established selector code using the `ReplaceGestalt` function.

ReplaceGestalt

You can use the `ReplaceGestalt` function to replace the function that is currently associated with a selector.

```
FUNCTION ReplaceGestalt (selector: OSType;
                        gestaltFunction: SelectorFunctionUUP;
                        VAR oldGestaltFunction:
                        SelectorFunctionUUP): OSErr;
```

`selector` The selector code for the function being replaced.

`gestaltFunction`
A pointer to the new selector function.

`oldGestaltFunction`
On exit, a pointer to the function previously associated with the specified selector.

DESCRIPTION

The `ReplaceGestalt` function replaces the selector function associated with an existing selector code.

So that your function can call the function previously associated with the selector, `ReplaceGestalt` places the address of the old selector function in the `oldGestaltFunction` parameter. If `ReplaceGestalt` returns an error of any type, then the value of `oldGestaltFunction` is undefined.

Gestalt Manager

SPECIAL CONSIDERATIONS

The `ReplaceGestalt` function might move memory and should not be called at interrupt time.

ASSEMBLY-LANGUAGE INFORMATION

The registers on entry and exit for the `ReplaceGestalt` function are

Registers on entry

A Address of new selector function

0

D0 Selector code

Registers on exit

A Address of old selector function

0

D0 Result code

RESULT CODES

<code>noErr</code>	0	No error
<code>gestaltUndefSelectorErr</code>	-5551	Undefined selector
<code>gestaltLocationErr</code>	-5553	Function not in system heap

SEE ALSO

See “Modifying a Selector Function” on page 1-13 for a discussion of replacing selector functions. See “Adding a New Selector Code” beginning on page 1-10 for a sample selector function.

Application-Defined Routines

This section describes the `Gestalt` selector function, which is the function `Gestalt` executes to retrieve the information specified by a selector.

The Selector Function

If you add your own selector code or modify an existing selector code, you supply a selector function that returns the information associated with the selector.

MySelectorFunction

The selector function is responsible for placing the requested information in the response parameter and returning an appropriate error code.

```
FUNCTION MySelectorFunction (selector: OSType;
                             VAR response: LongInt): OSErr;
```

`selector` The selector code that triggers the function.
`response` On exit, the information.

DESCRIPTION

The 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 Gestalt returns as its function result.

A selector function can call Gestalt or even other selector functions. It must reside in the system heap.

ASSEMBLY-LANGUAGE INFORMATION

The registers on entry and exit for the selector function are

Registers on entry

D0 Selector code

Registers on exit

A Response
 0
 D0 Result
 code

RESULT CODES

<code>noErr</code>	0	No error
<code>gestaltUnknownErr</code>	-5550	Could not obtain the response

SEE ALSO

See “Adding a New Selector Code” beginning on page 1-10 for a sample selector function and a procedure that installs it in the system heap. For information about the `NewGestalt` function, see page 1-34. For information about the `ReplaceGestalt` function, see page 1-35.

Summary of the Gestalt Manager

Pascal Summary

Constants

Environmental Selector Codes

CONST

gestaltAddressingModeAttr	= 'addr';	{addressing-mode attributes}
gestaltAliasMgrAttr	= 'alis';	{Alias Manager attributes}
gestaltAppleEventsAttr	= 'evnt';	{Apple events attributes}
gestaltAppleTalkVersion	= 'atlk';	{old format AppleTalk version}
gestaltATalkVersion	= 'atkv';	{new format AppleTalk version}
gestaltAUXVersion	= 'a/ux';	{A/UX version, if present}
gestaltCFMAttr	= 'cfrg';	{Code Fragment Manager attr}
gestaltCloseViewAttr	= 'BSDa';	{CloseView attributes}
gestaltComponentMgr	= 'cpnt';	{Component Manager version}
gestaltCompressionMgr	= 'icmp';	{Image Compression Manager { version}}
gestaltConnMgrAttr	= 'conn';	{Connection Manager attributes}
gestaltCRMAttr	= 'crm';	{Communication Resource { Manager attr}}
gestaltCTBVersion	= 'ctbv';	{Comm Toolbox version}
gestaltDBAccessMgrAttr	= 'dbac';	{Data Access Manager attributes}
gestaltDictionaryMgrAttr	= 'dict';	{Dictionary Manager attributes}
gestaltDisplayMgrAttr	= 'dply';	{Display Manager attributes}
gestaltDisplayMgrVers	= 'dplv';	{Display Manager version}
gestaltDITLExtAttr	= 'ditl';	{Dialog Manager extensions}
gestaltDragMgrAttr	= 'drag';	{Drag Manager attributes}
gestaltEasyAccessAttr	= 'easy';	{Easy Access attributes}
gestaltEditionMgrAttr	= 'edtn';	{Edition Manager attributes}
gestaltExtToolboxTable	= 'xttt';	{Toolbox trap dispatch table}
gestaltFinderAttr	= 'fndr';	{Finder attributes}
gestaltFindFolderAttr	= 'fold';	{FindFolder attributes}
gestaltFirstSlotNumber	= 'slt1';	{first physical slot}
gestaltFontMgrAttr	= 'font';	{Font Manager attributes}
gestaltFPUType	= 'fpu';	{floating-point unit type}

Gestalt Manager

gestaltFSAttr	= 'fs ';	{file system attributes}
gestaltFXfrMgrAttr	= 'fxfr';	{File Transfer Manager attr}
gestaltHardwareAttr	= 'hdwr';	{hardware attributes}
gestaltHelpMgrAttr	= 'help';	{Help Manager attributes}
gestaltIconUtilitiesAttr	= 'icon';	{Icon Utilities attributes}
gestaltKeyboardType	= 'kbd ';	{keyboard type code}
gestaltLogicalPageSize	= 'pgsz';	{logical page size}
gestaltLogicalRAMSize	= 'lram';	{logical RAM size}
gestaltLowMemorySize	= 'lmem';	{size of low memory}
gestaltMiscAttr	= 'misc';	{miscellaneous attributes}
gestaltMixedModeVersion	= 'mixd';	{MixedMode version}
gestaltMMUType	= 'mmu ';	{MMU type}
gestaltNativeCPUtype	= 'cput';	{Native CPU type}
gestaltNotificationMgrAttr	= 'nmgr';	{Notification Manager attr}
gestaltNuBusConnectors	= 'sltc';	{NuBus connector bitmap}
gestaltNuBusSlotCount	= 'nubs';	{count of logical NuBus slots}
gestaltOSAttr	= 'os ';	{Operating System attributes}
gestaltOSTable	= 'ostt';	{base address of Operating } { System trap dispatch table}
gestaltParityAttr	= 'prty';	{parity attributes}
gestaltPCXAttr	= 'pcxg';	{PC exchange attributes}
gestaltPhysicalRAMSize	= 'ram ';	{physical RAM size}
gestaltPopupAttr	= 'pop!';	{pop-up 'CDEF' attributes}
gestaltPowerMgrAttr	= 'powr';	{Power Manager attributes}
gestaltPPCToolboxAttr	= 'ppc ';	{PPC Toolbox attributes}
gestaltProcessorType	= 'proc';	{microprocessor type code}
gestaltQuickdrawFeatures	= 'qdrw';	{QuickDraw features}
gestaltQuickdrawVersion	= 'qd ';	{QuickDraw version}
gestaltQuickTime	= 'qtim';	{QuickTime version}
gestaltRealtimeAttr	= 'rtmr';	{Realtime Manager attributes}
gestaltResourceMgrAttr	= 'rsrc';	{Resource Manager attributes}
gestaltScrapMgrAttr	= 'scra';	{Scrap Manager attributes}
gestaltScriptCount	= 'scr#';	{number of active script } { systems}
gestaltScriptMgrVersion	= 'scri';	{Script Manager version}
gestaltSerialAttr	= 'ser ';	{serial hardware attributes}
gestaltSoundAttr	= 'snd ';	{sound attributes}
gestaltSpeechAttr	= 'ttsc';	{Speech Manager attributes}
gestaltStandardFileAttr	= 'stdf';	{Standard File attributes}
gestaltStdNBPAttr	= 'nlup';	{StandardNBP attributes}
gestaltSysArchitecture	= 'sysa';	{native system architecture}
gestaltTEAttr	= 'teat';	{TextEdit attributes}
gestaltTermMgrAttr	= 'term';	{Terminal Manager attributes}

Gestalt Manager

```

gestaltTextEditVersion      = 'te  ';    {TextEdit version code}
gestaltThreadMgrAtt        = 'thds';    {Thread Manager attributes}
gestaltTimeMgrVersion      = 'tmgr';    {Time Manager version code}
gestaltTranslationAttr     = 'xlat';    {Translation Manager attributes}
gestaltTSMgrVersion        = 'tsmv';    {Text Services Manager version}
gestaltToolboxTable        = 'tbtt';    {base address of Toolbox trap }
                                { dispatch table}

gestaltVersion              = 'vers';    {Gestalt version}
gestaltVMAttr               = 'vm  ';    {virtual memory attributes}

```

Informational Selector Codes

CONST

```

gestaltHardwareAttr        = 'hdwr';    {hardware attributes}
gestaltMachineIcon         = 'micn';    {machine 'ICON'/'cicn' res ID}
gestaltMachineType         = 'mach';    {Macintosh model code}
gestaltROMSize              = 'rom  ';    {ROM size}
gestaltROMVersion          = 'romv';    {ROM version}
gestaltSystemVersion       = 'sysv';    {System file version number}

```

Environmental Selector Response Values

CONST

```

{gestaltAddressingModeAttr response bits}
gestalt32BitAddressing     = 0;        {booted in 32-bit mode}
gestalt32BitSysZone        = 1;        {32-bit compatible system zone}
gestalt32BitCapable        = 2;        {machine is 32-bit capable}

{gestaltAliasMgrAttr response bits}
gestaltAliasMgrPresent     = 0;        {Alias Manager is present}
gestaltAliasMgrSupportsRemoteAppletalk
                            = 1;        {Alias Manager knows about }
                                    { remote AppleTalk}

{gestaltAppleEventsAttr response bits}
gestaltAppleEventsPresent  = 0;        {Apple events available}
gestaltScriptingSupport    = 1;
gestaltOSLInSystem         = 2;        {OSL in system}

{gestaltATalkVersion release stage constants}
development                 = $20;    {development}
alpha                       = $40;    {alpha}
beta                        = $60;    {beta}
final                       = $80;    {final}
release                     = $80;    {release}

```

Gestalt Manager

```

{gestaltCFMAttr response bits}
gestaltsCFMPresent          = 0;      {Code Fragment Manager present}

{gestaltCloseViewAttr response bits}
gestaltCloseViewEnabled     = 0;      {CloseView enabled}
gestaltCloseViewDisplayMgrFriendly
                             = 1;      {CloseView compatible with }
                                     { Display Manger}

{gestaltConnMgrAttr response bits}
gestaltConnMgrPresent       = 0;      {Connection Manager present}
gestaltConnMgrCMSearchFix   = 1;      {CMAddSearch fix present}
gestaltConnMgrErrorString   = 2;      {has CMGetErrorString}
gestaltConnMgrMultiAsyncIO  = 3;      {has CMNewIOPB, CMDisposeIOPB, }
                                     { CMPBRead, CMPBWrite,and }
                                     { CMPBIOKill}

{gestaltCRMAttr response bits}
gestaltCRMPresent           = 0;      {Communication Resource Manager }
                                     { present}
gestaltCRMPersistentFix     = 1;      {fix for persistent tools}
gestaltCRMToolRsrcCalls     = 2;      {tool resource calls available}

{gestaltDBAccessMgrAttr response bits}
gestaltDBAccessMgrPresent   = 0;      {Data Access Manager present}

{gestaltDisplayMgrAttr response bits}
gestaltDisplayMgrPresent    = 0;      {Display Manager Present}

{gestaltDictionaryMgrAttr response bits}
gestaltDictionaryMgrPresent = 0;      {Dictionary Manager present}

{gestaltDITLExtAttr response bits}
gestaltDITLExtPresent       = 0;      {Dialog Manager extensions }
                                     { present}

{gestaltDragMgrAttr response bits}
gestaltDragMgrPresent       = 0;      {Drag Manager present}

{gestaltEasyAccessAttr response bits}
gestaltEasyAccessOff        = 0;      {Easy Access present but off}
gestaltEasyAccessOn         = 1;      {Easy Access on}
gestaltEasyAccessSticky     = 2;      {Easy Access sticky}
gestaltEasyAccessLocked     = 3;      {Easy Access locked}

```

Gestalt Manager

```

{gestaltEditionMgrAttr response bits}
gestaltEditionMgrPresent      = 0;      {Edition Manager present}
gestaltEditionMgrTranslationAware = 1;  {Edition Manager aware of }
                                   { Translation Manager}

{gestaltFinderAttr response bits}
gestaltFinderDropEvent       = 0;      {Finder recognizes drop event}
gestaltFinderMagicPlacement  = 1;      {Finder supports magic icon }
                                   { placement}
gestaltFinderCallsAEProcess  = 2;      {Finder calls }
                                   { AEProcessAppleEvent}
gestaltFinderOSLCompliantFinder = 3;  {Finder is scriptable and }
                                   { recordable}

gestaltFinderSupports4GBVolumes = 4;   {Finder handles 4GB volumes}
gestaltFinderHandlesCFMFailures = 5;  {Finder handles Code Fragment }
                                   { Manager errors}
gestaltFinderHasClippings     = 6;     {Finder supports Drag Manager }
                                   { clipping files}

{gestaltFindFolderAttr response bits}
gestaltFindFolderPresent     = 0;      {FindFolder available}

{gestaltFontMgrAttr response values}
gestaltOutlineFonts          = 0;      {outline fonts supported}

{gestaltFPUType response values}
gestaltNoFPU                  = 0;     {no FPU}
gestalt68881                   = 1;    {Motorola 68881 FPU}
gestalt68882                   = 2;    {Motorola 68882 FPU}
gestalt68040FPU                = 3;    {built-in 68040 }
                                   { floating-point processing}

{gestaltFSAttr response bits}
gestaltFullExtFSDispatching   = 0;    {new HFSDispatch available}
gestaltHasFSSpecCalls         = 1;    {has FSSpec calls}
gestaltHasFileSystemManager   = 2;    {has File System Manager}
gestaltHasFileSystemManager   = 3;    {supports dynamic loading}
gestaltFSSupports4GBVols      = 4;    {supports 4 gigabyte volume}
gestaltHasExtendedDiskInit    = 6;    {has extended disk }
                                   { initialization calls}

```


Gestalt Manager

```

{gestaltFXfrMgrAttr response bits}
gestaltFXfrMgrPresent      = 0;      {File Transfer Manager present}
gestaltFXfrMgrMultiFile    = 1;      {supports FTSend and FTReceive}
gestaltFXfrMgrErrorString  = 2;      {supports FTGetErrorString}

{gestaltHelpMgrAttr response bits}
gestaltHelpMgrPresent      = 0;      {Help Manager present}

{gestaltIconUtilitiesAttr response value}
gestaltIconUtilitiePresents = 0;      {Icon Utilities are present}

{gestaltKeyboardType response values}
gestaltMacKbd              = 1;      {Macintosh}
gestaltMacAndPad           = 2;      {Macintosh with keypad}
gestaltMacPlusKbd         = 3;      {Macintosh Plus}
gestaltExtADBKbd          = 4;      {extended ADB}
gestaltStdADBKbd          = 5;      {standard ADB}
gestaltPrtblADBKbd        = 6;      {Portable ADB}
gestaltPrtblISOKbd        = 7;      {Portable ISO ADB}
gestaltStdISOADBKbd       = 8;      {ISO standard ADB}
gestaltExtISOADBKbd       = 9;      {ISO extended ADB}
gestaltADBKbdII           = 10;     {ADB II}
gestaltADBISOKbdII        = 11;     {ISO ADB II}
gestaltPwrBookADBKbd      = 12;     {PowerBook ADB}
gestaltPwrBookISOADBKbd   = 13;     {PowerBook ISO ADB}
gestaltAppleAdjustKeypad  = 14;     {Adjustable Keypad}
gestaltAppleAdjustADBKbd  = 15;     {Adjustable ADB}
gestaltAppleAdjustISOKbd  = 16;     {Adjustable ISO}

{gestaltMiscAttr response bits}
gestaltScrollingThrottle  = 0;      {scrolling throttle is on}
gestaltSquareMenuBar      = 2;      {menu bar is square}

{gestaltMMUType response values}
gestaltNoMMU              = 0;      {no MMU}
gestaltAMU                 = 1;      {Mac II address management unit}
gestalt68851               = 2;      {Motorola 68851 PMMU}
gestalt68030MMU           = 3;      {built-in 68030 MMU}
gestalt68040MMU           = 4;      {built-in 68040 MMU}
gestaltEMMU1              = 5;      {emulated MMU type 1}

{gestaltNativeCPUtype response values}
gestaltCPU68000           = $000;    {Macintosh 68000 CPU}
gestaltCPU68010           = $001;    {Macintosh 68010 CPU}
gestaltCPU68020           = $002;    {Macintosh 68020 CPU}

```

Gestalt Manager

```

gestaltCPU68030      = $003;    {Macintosh 68030 CPU}
gestaltCPU68040      = $004;    {Macintosh 68040 CPU}
gestaltCPU601        = $101;    {PowerPC 601 CPU}

{gestaltNotificationMgrAttr response bits}
gestaltNotificationPresent = 0;    {Notification Manager present}

{gestaltOSAttr response bits}
gestaltSysZoneGrowable  = 0;    {system heap can grow}
gestaltLaunchCanReturn  = 1;    {can return from launch}
gestaltLaunchFullFileSpec = 2;    {LaunchApplication available}
gestaltLaunchControl    = 3;    {Process Manager available}
gestaltTempMemSupport   = 4;    {temporary memory support }
                             { available}
gestaltRealTempMemory   = 5;    {temporary memory handles are }
                             { real}
gestaltTempMemTracked   = 6;    {temporary memory handles are}
                             { tracked}

{gestaltParityAttr response bits}
gestaltHasParityCapability = 0;    {machine can check parity}
gestaltParityEnabled      = 1;    {parity RAM is installed}

{gestaltPCXAttr response bits}
gestaltPCXHas8and16BitFat = 0;    {PC exchange supports both }
                             { 8 and 16 bit FATs}
gestaltPCXHasProDOS       = 1;    {PC exchange supports ProDos}

{gestaltPopupAttr response bits}
gestaltPopupPresent       = 0;    {pop-up 'CDEF' is present}

{gestaltPowerMgrAttr response bits}
gestaltPMgrExists        = 0;    {Power Manager is present}
gestaltPMgrCPUIdle       = 1;    {CPU can idle}
gestaltPMgrSCC           = 2;    {Power Manager can stop SCC }
                             { clock}
gestaltPMgrSound         = 3;    {Power Manager can turn off }
                             { sound power}
gestaltPMgrDispatchExists = 4;    {Power Manager dispatch exists}

{gestaltPPCToolboxAttr response masks}
gestaltPPCToolboxPresent = $0000; {PPC Toolbox is present;
                                     { PPCInit has been called}
gestaltPPCSupportsRealTime = $1000; {supports real-time delivery}
gestaltPPCSupportsIncoming = $0001; {accepts sessions from remote }

```

Gestalt Manager

```

gestaltPPCSupportsOutGoing    = $0002;    { computers }
                                   { can initiate sessions with }
                                   { remote computers }

{gestaltProcessorType response values}
gestalt68000                  = 1;        {68000 microprocessor}
gestalt68010                  = 2;        {68010 microprocessor}
gestalt68020                  = 3;        {68020 microprocessor}
gestalt68030                  = 4;        {68030 microprocessor}
gestalt68040                  = 5;        {68040 microprocessor}

{gestaltQuickdrawFeatures response bits}
gestaltHasColor                = 0;        {Color QuickDraw present}
gestaltHasDeepGWorlds         = 1;        {graphics worlds can be deeper }
                                   { than 1 bit}
gestaltHasDirectPixMaps       = 2;        {PixMaps can be direct }
                                   { (16- or 32-bit)}
gestaltHasGrayishTextOr       = 3;        {supports text mode }
                                   { grayishTextOr}
gestaltSupportsMirroring       = 4;        {supports video mirroring }
                                   { using the Display Manager}

{gestaltQuickdrawVersion response values}
gestaltOriginalQD              = $000;    {original 1-bit QuickDraw}
gestalt8BitQD                  = $100;    {8-bit QuickDraw}
gestalt32BitQD                 = $200;    {32-Bit QuickDraw vers. 1.0}
gestalt32BitQD11               = $210;    {32-Bit QuickDraw vers. 1.1}
gestalt32BitQD12               = $220;    {32-Bit QuickDraw vers. 1.2}
gestalt32BitQD13               = $230;    {32-Bit QuickDraw vers. 1.3}

{gestaltRealtimeAttr response bits}
gestaltRealtimeMgrPresent      = 0;        {Realtime Manager present}

{gestaltResourceMgrAttr response bits}
gestaltPartialRsrcs            = 0;        {partial resources supported}

{gestaltScrapMgrAttr response bits}
gestaltScrapMgrTranslationAware
                                   = 0;        {aware of Translation Manager}
gestaltTranslationMgrHintOrder
                                   = 1;        {hint order reversal present}

```

Gestalt Manager

```

{gestaltSerialAttr response bits}
gestaltHasGPIaToDCDa      = 0;      {GPI connected to DCD on port A}
gestaltHasGPIaToRTxCa    = 1;      {GPI connected to RTxC on }
                                { port A}
gestaltHasGPIaToDCDb    = 2;      {GPI connected to DCD on port B}

{gestaltSoundAttr response bits}
gestaltStereoCapability  = 0;      {stereo capability present}
gestaltStereoMixing     = 1;      {stereo mixing on internal }
                                { speaker}
gestaltSoundIOMgrPresent = 3;      {sound input routines present}
gestaltBuiltInSoundInput = 4;      {built-in input device present}
gestaltHasSoundInputDevice = 5;    {sound input device present}
gestaltPlayAndRecord    = 6;      {built-in hardware can play }
                                { and record simultaneously}
gestalt16BitSoundIO     = 7;      {sound hardware can play and }
                                { record 16-bit samples}
gestaltStereoInput      = 8;      {sound hardware can }
                                { record stereo}
gestaltSndPlayDoubleBuffer = 10;   {SndPlayDouble buffer present}
gestaltMultiChannels    = 11;     {multiple channel support}
gestalt16BitAudioSupport = 12;    {16-bit audio data supported}

{gestaltSpeechAttr response bits}
gestaltSpeechMgrPresent  = 0;      {Speech Manager present}
gestaltSpeechHasPPCGLue = 1;      {Speech Manager has native PPC }
                                { glue for API}

{gestaltStandardFileAttr response bits}
gestaltStandardFile58    = 0;      {has functions new with 7.0}
gestaltStandardFileTranslationAware = 1;
                                {aware of Translation Manager}
gestaltStandardFileHasColorIcons = 2;
                                {dialog boxes use small color }
                                { icons}

{gestaltStdNBPAttr response bits}
gestaltStdNBPPresent     = 0;      {StandardNBP is present}

{gestaltSysArchitecture response bits}
gestalt68k                = 1;      {MC680x0 architecture}
gestaltPowerPC            = 2;      {PowerPC architecture}

{gestaltTEAttr response bits}
gestaltTEHasGetHiliteRgn = 0;      {TextEdit has TEGetHiliteRgn}

```

Gestalt Manager

```

{gestaltTermMgrAttr response bits}
gestaltTermMgrPresent      = 0;      {Terminal Manager present}
gestaltTermMgrErrorString  = 2;      {supports error string }
                                   { function}

{gestaltTextEditVersion response values}
gestaltTE1                 = 1;      {in MacIIci ROM}
gestaltTE2                 = 2;      {with 6.0.4 scripts on Mac IIci}
gestaltTE3                 = 3;      {with 6.0.4 scripts on other }
                                   { machines}

gestaltTE4                 = 4;      {in 6.0.5 and 7.0}
gestaltTE5                 = 5;      {TextWidthHook available}

{gestaltThreadMgrAttr response bits}
gestaltThreadMgrPresent    = 0;      {Thread Manger present}
gestaltSpecificMatchSupport = 1;      {Thread Manager supports }
                                   { exact match creation option}

{gestaltTimeMgrVersion response values}
gestaltStandardTimeMgr     = 1;      {standard Time Manager}
gestaltRevisedTimeMgr     = 2;      {revised Time Manager}
gestaltExtendedTimeMgr    = 3;      {extended Time Manager}

{gestaltTranslationAttr response codes}
gestaltTranslationMgrExists = 0;      {Translation Manager present}

{gestaltVMAttr response bits}
gestaltVMPresent          = 0;      {virtual memory present}

```

Informational Selector Response Values

CONST

```

{gestaltHardwareAttr response bits}
gestaltHasVIA1             = 0;      {has VIA1 chip}
gestaltHasVIA2             = 1;      {has VIA2 chip}
gestaltHasASC              = 3;      {has Apple sound chip}
gestaltHasSCC              = 4;      {has SCC}
gestaltHasSCSI             = 7;      {has SCSI}
gestaltHasSoftPowerOff     = 19;     {capable of software power off}
gestaltHasSCSI961         = 21;     {has 53C96 SCSI on internal bus}
gestaltHasSCSI962         = 22;     {has 53C96 SCSI on external bus}
gestaltHasUniversalROM     = 24;     {has universal ROM}

```

Gestalt Manager

```
{gestaltMachineType response values}
gestaltClassic           = 1;      {Macintosh 128K}
gestaltMacXL             = 2;      {Macintosh XL}
gestaltMac512KE         = 3;      {Macintosh 512K enhanced}
gestaltMacPlus          = 4;      {Macintosh Plus}
gestaltMacSE            = 5;      {Macintosh SE}
gestaltMacII            = 6;      {Macintosh II}
gestaltMacIIX           = 7;      {Macintosh IIX}
gestaltMacIICX          = 8;      {Macintosh IICX}
gestaltMacSE030        = 9;      {Macintosh SE/30}
gestaltPortable         = 10;     {Macintosh Portable}
gestaltMacIICI          = 11;     {Macintosh IICI}
gestaltMacIIFX          = 13;     {Macintosh IIFX}
gestaltMacClassic      = 17;     {Macintosh Classic}
gestaltMacIISI          = 18;     {Macintosh IISI}
gestaltMacLC            = 19;     {Macintosh LC}
gestaltQuadra900       = 20;     {Macintosh Quadra 900}
gestaltPowerBook170    = 21;     {Macintosh PowerBook 170}
gestaltQuadra700       = 22;     {Macintosh Quadra 700}
gestaltClassicII       = 23;     {Macintosh Classic II}
gestaltPowerBook100    = 24;     {Macintosh PowerBook 100}
gestaltPowerBook140    = 25;     {Macintosh PowerBook 140}
gestaltQuadra950       = 26;     {Macintosh Quadra 950}
gestaltMacLClIIII      = 27;     {Macintosh LC III}
gestaltPowerBookDuo210 = 29;     {Macintosh PowerBook Duo 210}
gestaltMacCentris650   = 30;     {Macintosh Centris 650}
gestaltPowerBookDuo230 = 32;     {Macintosh PowerBook Duo 230}
gestaltPowerBook180    = 33;     {Macintosh PowerBook 180}
gestaltPowerBook160    = 34;     {Macintosh PowerBook 160}
gestaltMacQuadra800    = 35;     {Macintosh Quadra 800}
gestaltMacLClII        = 37;     {Macintosh LC II}
gestaltPowerBookDuo250 = 38;     {Macintosh PowerBook Duo 230}
gestaltMacIIVI          = 44;     {Macintosh IIVI}
gestaltPerforma600     = 45;     {Macintosh Performa 600}
gestaltMacIIVX          = 48;     {Macintosh IIVX}
gestaltMacColorClassic = 49;     {Macintosh Color Classic}
gestaltPowerBook165c   = 50;     {Macintosh PowerBook 165c}
gestaltMacCentris610   = 52;     {Macintosh Centris 610}
gestaltMacQuadra610    = 53;     {Macintosh Quadra 610}
gestaltPowerBook145    = 54;     {Macintosh PowerBook 145}
gestaltMacLCl520       = 56;     {Macintosh LC 520}
gestaltMacCentris660AV = 60;     {Macintosh Centris 660 AV}
gestaltPowerBook180c   = 71;     {Macintosh PowerBook 180c}
```

Gestalt Manager

```

getstaltPowerBookDuo270c      = 77;      {Macintosh PowerBook Duo 270c}
getstaltMacQuadra840AV       = 78;      {Macintosh Quadra 840 AV}
getstaltPowerBook165         = 84;      {Macintosh PowerBook 165}
getstaltMacTV                 = 88;      {Macintosh TV}
getstaltMacLC475              = 89;      {Macintosh LC 475}
getstaltMacLC575              = 92;      {Macintosh LC 575}
getstaltMacQuadra605         = 94;      {Macintosh Quadra 605}

getstaltPowerMac8100_80      = 65;      {Power Macintosh 8100/80}
getstaltPowerMac6100_60      = 75;      {Power Macintosh 6100/60}
getstaltPowerMac7100_66      = 112;     {Power Macintosh 7100/66}

kMachineNameStrID            = -16395;   {'STR#' resource that }
                                { contains machine names}

```

SysEnviron Constants

CONST

```

curSysEnvVers                 = 2;      {current SysEnviron version}

{machine types}
envXL                         = -2;     {Macintosh XL}
envMac                         = -1;     {Macintosh with 64K ROM}
envMachUnknown                 = 0;     {unknown model, after }
                                { Macintosh IIfx}

env512KE                       = 1;     {Macintosh 512K enhanced}
envMacPlus                     = 2;     {Macintosh Plus}
envSE                           = 3;     {Macintosh SE}
envMacII                       = 4;     {Macintosh II}
envMacIIX                      = 5;     {Macintosh IIX}
envMacIICX                     = 6;     {Macintosh IICX}
envSE30                        = 7;     {Macintosh SE30}
envPortable                    = 8;     {Macintosh Portable}
envMacIICi                     = 9;     {Macintosh IICi}
envMacIIFx                     = 11;    {Macintosh IIFx}

{system environment record microprocessor codes}
envCPUUnknown                  = 0;     {unknown microprocessor}
env68000                       = 1;     {68000 microprocessor}
env68010                       = 2;     {68010 microprocessor}
env68020                       = 3;     {68020 microprocessor}
env68030                       = 4;     {68030 microprocessor}
env68040                       = 5;     {68040 microprocessor}

```

Gestalt Manager

```
{system environment record keyBoardType codes}
envUnknownKbd           = 0;      {Macintosh Plus with keypad}
envMacKbd               = 1;      {Macintosh}
envMacAndPad            = 2;      {Macintosh with keypad}
envMacPlusKbd           = 3;      {Macintosh Plus}
envAExtendKbd           = 4;      {Apple extended}
envStandADBKbd          = 5;      {standard ADB}
envPrtblADBKbd          = 6;      {Macintosh Portable ADB}
envPrtblISOKbd          = 7;      {Macintosh Portable ISO}
envStdISOADBKbd         = 8;      {standard ISO ADB}
envExtISOADBKbd         = 9;      {extended ISO ADB}
```

Data Types

```
TYPE SysEnvRec =          {system environment record}
RECORD
    environsVersion: Integer; {SysEnvirons version number}
    machineType:      Integer; {Macintosh model code}
    systemVersion:   Integer; {System file version number}
    processor:       Integer; {microprocessor type code}
    hasFPU:          Boolean;  {floating-point unit flag}
    hasColorQD:      Boolean;  {Color QuickDraw flag}
    keyBoardType:    Integer;  {keyboard type code}
    atDrvrVersNum:   Integer;  {AppleTalk driver version number}
    sysVRefNum:      Integer;  {working directory reference number of }
                                { folder or volume containing open }
                                { System file}
END;
```

Gestalt Manager Routines

Getting Information About the Operating Environment

```
FUNCTION Gestalt          (selector: OSType;
                          VAR response: LongInt): OSErr;

FUNCTION SysEnvirons     (versionRequested: Integer;
                          VAR theWorld: SysEnvRec): OSErr;
```


Adding a Selector Code

```
FUNCTION NewGestalt      (selector: OSType;
                        gestaltFunction: SelectorFunctionUUP): OSErr;
```

Modifying a Selector Function

```
FUNCTION ReplaceGestalt (selector: OSType;
                        gestaltFunction: SelectorFunctionUUP;
                        VAR oldGestaltFunction: SelectorFunctionUUP)
                        : OSErr;
```

Application-Defined Routines

```
FUNCTION MySelectorFunction
                        (selector: OSType; VAR response: LongInt)
                        : OSErr;
```

C Summary

Constants

Environmental Selector Codes

```
#define gestaltAddressingModeAttr  'addr'  /*addressing-mode attributes*/
#define gestaltAliasMgrAttr        'alis'  /*Alias Manager attributes*/
#define gestaltAppleEventsAttr    'evnt'  /*Apple events attributes*/
#define gestaltAppleTalkVersion   'atlk'  /*old format AppleTalk version*/
#define gestaltATalkVersion       'atkv'  /*new format AppleTalk version*/
#define gestaltAUXVersion         'a/ux'  /*A/UX version, if present*/
#define gestaltCFMAttr            'cfrg'  /*Code Fragment Manager attr*/
#define gestaltCloseViewAttr      'BSDa'  /*CloseView attributes*/
#define gestaltComponentMgr       'cpnt'  /*Component Manager version*/
#define gestaltCompressionMgr     'icmp'  /*Image Compression Manager */
                                     /* version*/

#define gestaltConnMgrAttr        'conn'  /*Connection Manager attr*/
#define gestaltCRMAttr            'crm'   /*Comm Resource Manager attr*/
#define gestaltCTBVersion        'ctbv'  /*Comm Toolbox version*/
#define gestaltDBAccessMgrAttr   'dbac'  /*Data Access Manager attr*/
#define gestaltDictionaryMgrAttr  'dict'  /*Dictionary Manager attr*/
#define gestaltDisplayMgrAttr    'dply'  /*Display Manager attributes*/
#define gestaltDisplayMgrVers    'dplv'  /*Display Manager version*/
```

Gestalt Manager

```

#define gestaltDITLExtAttr      'ditl' /*Dialog Manager extensions*/
#define gestaltDragMgrAttr     'drag' /*Drag Manager attributes*/
#define gestaltEasyAccessAttr  'easy' /*Easy Access attributes*/
#define gestaltEditionMgrAttr  'edtn' /*Edition Manager attributes*/
#define gestaltExtToolboxTable 'xttt' /*Toolbox trap dispatch table*/
#define gestaltFinderAttr     'fndr' /*Finder attributes*/
#define gestaltFindFolderAttr  'fold' /*FindFolder attributes*/
#define gestaltFirstSlotNumber 'slt1' /*first physical slot*/
#define gestaltFontMgrAttr     'font' /*Font Manager attributes*/
#define gestaltFPUType        'fpu' /*floating-point unit type*/
#define gestaltFSAttr         'fs' /*file system attributes*/
#define gestaltFXfrMgrAttr     'fxfr' /*File Transfer Manager attr*/
#define gestaltHelpMgrAttr     'help' /*Help Manager attributes*/
#define gestaltKeyboardType    'kbd' /*keyboard type code*/
#define gestaltLogicalPageSize 'pgsz' /*logical page size*/
#define gestaltLogicalRAMSize  'lram' /*logical RAM size*/
#define gestaltLowMemorySize   'lmem' /*size of low memory*/
#define gestaltMiscAttr       'misc' /*miscellaneous attributes*/
#define gestaltMixedModeVersion 'mixd' /*MixedMode version*/
#define gestaltMMUType        'mmu' /*MMU type*/
#define gestaltNativeCPUtype   'cput' /*Native CPU type*/
#define gestaltNotificationMgrAttr 'nmgr' /*Notification Manager attr*/
#define gestaltNuBusConnectors 'sltc' /*NuBus connector bitmap*/
#define gestaltNuBusSlotCount  'nubs' /*count of logical NuBus slots*/
#define gestaltOSAttr         'os' /*Operating System attributes*/
#define gestaltOSTable        'ostt' /*base address of Operating */
                                /* System trap dispatch table*/

#define gestaltParityAttr      'prty' /*parity attributes*/
#define gestaltPCXAttr        'pcxg' /*PC exchange attributes*/
#define gestaltPhysicalRAMSize 'ram' /*physical RAM size*/
#define gestaltPopupAttr      'pop!' /*pop-up 'CDEF' attributes*/
#define gestaltPowerMgrAttr    'powr' /*Power Manager attributes*/
#define gestaltPPCToolboxAttr  'ppc' /*PPC Toolbox attributes*/
#define gestaltProcessorType   'proc' /*microprocessor type code*/
#define gestaltQuickdrawFeatures 'qdrw' /*QuickDraw features*/
#define gestaltQuickdrawVersion 'qd' /*QuickDraw version*/
#define gestaltQuickTime       'qtim' /*QuickTime version*/
#define gestaltRealtimeAttr    'rtmr' /*Realtime Manager attributes*/
#define gestaltResourceMgrAttr 'rsrc' /*Resource Manager attributes*/
#define gestaltScrapMgrAttr    'scra' /*Scrap Manager attributes*/
#define gestaltScriptCount     'scr#' /*number of active script */
                                /* systems*/

#define gestaltScriptMgrVersion 'scri' /*Script Manager version*/

```

Gestalt Manager

```

#define gestaltSerialAttr      'ser ' /*serial hardware attributes*/
#define gestaltSoundAttr      'snd ' /*sound attributes*/
#define gestaltSpeechAttr     'ttsc' /*Speech Manager attributes*/
#define gestaltStandardFileAttr 'stdf' /*Standard File attributes*/
#define gestaltStdNBPAttr     'nlup' /*StandardNBP attributes*/
#define gestaltSysArchitecture 'sysa' /*native system architecture*/
#define gestaltTEAttr        'teat' /*TextEdit attributes*/
#define gestaltTermMgrAttr    'term' /*Terminal Manager attributes*/
#define gestaltTextEditVersion 'te ' /*TextEdit version code*/
#define gestaltThreadMgrAttr  'thds' /*Thread Manager attributes*/
#define gestaltTimeMgrVersion 'tmgr' /*Time Manager version code*/
#define gestaltToolboxTable   'tbtt' /*base address of Toolbox */
                                /* trap dispatch table*/

#define gestaltTranslationAttr 'xlat' /*Translation Manager */
                                /* attributes*/

#define gestaltTSMgrVersion   'tsmv' /*Text Services Manager */
                                /*version*/

#define getstaltIconUtilities  'icon' /*Icon Utilities attributes*/
#define gestaltVersion        'vers' /*Gestalt version*/
#define gestaltVMAAttr        'vm ' /*virtual memory attributes*/

```

Informational Selector Codes

```

#define gestaltHardwareAttr    'hdwr' /*hardware attributes*/
#define gestaltMachineIcon     'micn' /*machine 'ICON'/'cicn' res ID*/
#define gestaltMachineType     'mach' /*Macintosh model code*/
#define gestaltROMSize         'rom ' /*ROM size*/
#define gestaltROMVersion      'romv' /*ROM version*/
#define gestaltSystemVersion   'sysv' /*System file version number*/

```

Environmental Selector Response Values

```

enum {
    /*gestaltAddressingModeAttr response bits*/
    gestalt32BitAddressing      = 0, /*booted in 32-bit mode*/
    gestalt32BitSysZone         = 1, /*32-bit compatible system */
                                /* zone*/
    gestalt32BitCapable         = 2  /*machine is 32-bit capable*/
};

enum {
    /*gestaltAliasMgrAttr response bits*/
    gestaltAliasMgrPresent      = 0, /*Alias Manager present*/

```

Gestalt Manager

```

    gestaltAliasMgrSupportsRemoteAppletalk    /*Alias Manager knows about */
                                                = 1    /* remote Appletalk*/
};

enum {
    /*gestaltAppleEventsAttr response bits*/
    gestaltAppleEventsPresent                = 0,    /*Apple Events available*/
    gestaltScriptingSupport                  = 1,
    gestaltOSLInSystem                      = 2    /*OSL in system*/
};

enum {
    /*gestaltATalkVersion release stage constants*/
    development                              = $20,    /*development*/
    alpha                                    = $40,    /*alpha*/
    beta                                     = $60,    /*beta*/
    final                                    = $80,    /*final*/
    release                                  = $80    /*release*/
};

enum {
    /*gestaltCFMAttr response bits*/
    gestaltCFMPresent                        = 0    /*Code Fragment Manager */
                                                /* present*/
};

enum {
    /*gestaltCloseViewAttr response bits*/
    gestaltCloseViewEnabled                 = 0,    /*CloseView enabled*/
    gestaltCloseViewDisplayMgrFriendly
                                                = 1    /*CloseView compatible with */
                                                /* Display Manger*/
};

enum {
    /*gestaltConnMgrAttr response bits*/
    gestaltConnMgrPresent                   = 0,    /*Connection Manager present*/
    gestaltConnMgrCMSearchFix              = 1,    /*CMAddSearch fix present*/
    gestaltConnMgrErrorString              = 2,    /*has CMGetErrorString*/
    gestaltConnMgrMultiAsyncIO            = 3    /*has CMNewIOPB, */
                                                /* CMDisposeIOPB, CMPBRead, */
                                                /* CMPBWrite, CMPBIOKill*/
};

```

Gestalt Manager

```

enum {
    /*gestaltCRMAttr response bits*/
    gestaltCRMPresent          = 0,      /*Comm Resource Manager */
                                      /* present*/

    gestaltCRMPersistentFix    = 1,      /*fix for persistent tools*/
    gestaltCRMToolRsrcCalls    = 2,      /*tool resource calls */
                                      /* available*/
};

enum {
    /*gestaltDBAccessMgrAttr response bits*/
    gestaltDBAccessMgrPresent  = 0        /*Data Access Manager present*/
};

enum {
    /*gestaltDictionaryMgrAttr response bits*/
    gestaltDictionaryMgrPresent = 0        /*Dictionary Manager present*/
};

enum {
    /*gestaltDisplayMgrAttr response bits*/
    gestaltDisplayMgrPresent    = 0        /*Display Manager Present*/
};

enum {
    /*gestaltDITLExtAttr response bits*/
    gestaltDITLExtPresent       = 0        /*Dialog Manager extensions */
                                      /* present*/
};

enum {
    /*gestaltDragMgrAttr response bits*/
    gestaltDragMgrPresent       = 0        /*Drag Manager present*/
};

enum {
    /*gestaltEasyAccessAttr response bits*/
    gestaltEasyAccessOff        = 0,      /*Easy Access present but off*/
    gestaltEasyAccessOn         = 1,      /*Easy Access on*/
    gestaltEasyAccessSticky     = 2,      /*Easy Access sticky*/
    gestaltEasyAccessLocked     = 3,      /*Easy Access locked*/
};

```

Gestalt Manager

```

enum {
    /*gestaltEditionMgrAttr response bits*/
    gestaltEditionMgrPresent          = 0,    /*Edition Manager present*/
    gestaltEditionMgrTranslationAware = 1     /*Edition Manager aware of */
                                           /* Translation Manager*/
};

enum {
    /*gestaltFinderAttr response bits*/
    gestaltFinderDropEvent            = 0,    /*Finder recognizes drop event*/
    gestaltFinderMagicPlacement       = 1,    /*Finder supports magic icon */
                                           /* placement*/
    gestaltFinderCallsAEProcess       = 2,    /*Finder calls */
                                           /* AEProcessAppleEvent*/
    gestaltFinderOSLCompliantFinder   = 3,    /*Finder is scriptable and */
                                           /* recordable*/
    gestaltFinderSupports4GBVolumes   = 4,    /*Finder handles 4GB volumes*/
    gestaltFinderHandlesCFMFailures   = 5,    /*Finder handles Code */
                                           /* *Fragment Manager errors*/
    gestaltFinderHasClippings         = 6     /*Finder supports Drag */
                                           /* Manager clipping files*/
};

enum {
    /*gestaltFindFolderAttr response bits*/
    gestaltFindFolderPresent          = 0     /*FindFolder available*/
};

enum {
    /*gestaltFontMgrAttr response bits*/
    gestaltOutlineFonts               = 0     /*outline fonts supported*/
};

enum {
    /*gestaltFPUType response values*/
    gestaltNoFPU                      = 0,    /*no FPU*/
    gestalt68881                      = 1,    /*Motorola 68881 FPU*/
    gestalt68882                      = 2,    /*Motorola 68882 FPU*/
    gestalt68040FPU                   = 3     /*built-in 68040 */
                                           /* floating-point processing*/
};

```

Gestalt Manager

```

enum {
    /*gestaltFSAttr response bits*/
    gestaltFullExtFSDispatching    = 0,    /*new HFSDispatch available*/
    gestaltHasFSSpecCalls          = 1,    /*has FSSpec calls*/
    gestaltHasFileSystemManager    = 2,    /*has File System Manager*/
    gestaltHasFileSystemManager    = 3,    /*supports dynamic loading*/
    gestaltFSSupports4GBVols       = 4,    /*supports 4 gigabyte volume*/
    gestaltHasExtendedDiskInit     = 6     /*has extended disk */
                                         /* initialization calls*/
};

enum {
    /*gestaltFXfrMgrAttr response bits*/
    gestaltFXfrMgrPresent          = 0,    /*File Transfer Manager */
                                         /* present*/
    gestaltFXfrMgrMultiFile       = 1,    /*supports FTSend and */
                                         /* FTReceive*/
    gestaltFXfrMgrErrorString     = 2     /*supports FTGetErrorString*/
};

enum {
    /*gestaltHelpMgrAttr response bits*/
    gestaltHelpMgrPresent         = 0     /*Help Manager present*/
};

enum {
    /*gestaltIconUtilitiesAttr response bits*/
    gestaltIconUtilitiesPresent   = 0     /*icon utilities present*/
};

enum {
    /*gestaltKeyboardType response values*/
    gestaltMacKbd                 = 1,    /*Macintosh*/
    gestaltMacAndPad              = 2,    /*Macintosh with keypad*/
    gestaltMacPlusKbd             = 3,    /*Macintosh Plus*/
    gestaltExtADBKbd              = 4,    /*extended ADB*/
    gestaltStdADBKbd              = 5,    /*standard ADB*/
    gestaltPrtblADBKbd            = 6,    /*Portable ADB */
    gestaltPrtblISOKbd            = 7,    /*Portable ISO ADB*/
    gestaltStdISOADBKbd           = 8,    /*ISO standard ADB*/
    gestaltExtISOADBKbd           = 9,    /*ISO extended ADB*/
    gestaltADBKbdII               = 10,   /*ADB II*/
    gestaltADBISOKbdII            = 11,   /*ISO ADB II*/
    gestaltPwrBookADBKbd          = 12,   /*PowerBook ADB*/
};

```

Gestalt Manager

```

    gestaltPwrBookISOADBkbd      = 13,    /*PowerBook ISO ADB*/
    gestaltAppleAdjustKeypad     = 14,    /*Adjustable Keypad*/
    gestaltAppleAdjustADBkbd     = 15,    /*Adjustable ADB*/
    gestaltAppleAdjustISOKbd     = 16     /*Adjustable ISO*/
};

enum {
    /*gestaltMiscAttr return bits*/
    gestaltScrollingThrottle     = 0,      /*scrolling throttle is on*/
    gestaltSquareMenuBar         = 2       /*menu bar is square*/
};

enum {
    /*gestaltMMUType return values*/
    gestaltNoMMU                 = 0,      /*no MMU*/
    gestaltAMU                   = 1,      /*Mac II address management */
                                        /* unit*/

    gestalt68851                 = 2,      /*Motorola 68851 PMMU*/
    gestalt68030MMU              = 3,      /*built-in 68030 MMU*/
    gestalt68040MMU              = 4,      /*built-in 68040 MMU*/
    gestaltEMMU1                 = 5       /*emulated MMU type 1*/
};

enum {
    /*gestaltNativeCPUtype response values*/
    gestaltCPU68000              = $000,   /*Macintosh 68000 CPU*/
    gestaltCPU68010              = $001,   /*Macintosh 68010 CPU*/
    gestaltCPU68020              = $002,   /*Macintosh 68020 CPU*/
    gestaltCPU68030              = $003,   /*Macintosh 68030 CPU*/
    gestaltCPU68040              = $004,   /*Macintosh 68040 CPU*/
    gestaltCPU601                = $101,   /*PowerPC 601 CPU*/
};

enum {
    /*gestaltNotificationMgrAttr response bits*/
    gestaltNotificationPresent    = 0       /*Notification Manager present*/
};

enum {
    /*gestaltOSAttr response bits*/
    gestaltSysZoneGrowable       = 0,      /*system heap can grow*/
    gestaltLaunchCanReturn       = 1,      /*can return from launch*/
    gestaltLaunchFullFileSpec    = 2,      /*LaunchApplication available*/
    gestaltLaunchControl         = 3,      /*Process Manager available*/
};

```


Gestalt Manager

```

gestaltTempMemSupport      = 4,      /*temporary memory support */
                             /* available*/
gestaltRealTempMemory     = 5,      /*temporary memory handles */
                             /* are real*/
gestaltTempMemTracked     = 6,      /*temporary memory handles */
                             /* are tracked*/
};

enum {
    /*gestaltParityAttr response bits*/
    gestaltHasParityCapability = 0,    /*machine can check parity*/
    gestaltParityEnabled      = 1     /*parity RAM is installed*/
};

enum {
    /*gestaltPCXAttr response bits*/
    gestaltPCXHas8and16BitFat = 0,    /*PC exchange supports both */
                                     /* 8 and 16 bit FATs*/
    /*gestaltPCXHasProDOS     = 1     /*PC exchange supports ProDos*/
};

enum {
    /*gestaltPopupAttr response bits*/
    gestaltPopupPresent      = 0     /*pop-up 'CDEF' is present*/
};

enum {
    /*gestaltPowerMgrAttr response bits*/
    gestaltPMgrExists        = 0,    /*Power Manager is present*/
    gestaltPMgrCPUIdle       = 1,    /*CPU can idle*/
    gestaltPMgrSCC           = 2,    /*Power Manager can stop SCC */
                                     /* clock*/
    gestaltPMgrSound         = 3,    /*Power Manager can turn off */
                                     /* sound power*/
    gestaltPMgrDispatchExists = 4     /*Power Mgr dispatch exists*/
};

enum {
    /* gestaltPPCToolboxAttr response bits*/
    gestaltPPCToolboxPresent = 0x0000, /*PPC Toolbox is present; */
                                     /* PPCInit has been called*/
    gestaltPPCSupportsRealTime = 0x1000, /*supports real-time delivery*/
    gestaltPPCSupportsIncoming = 0x0001, /*accepts sessions from */
                                     /* remote computers*/
};

```

Gestalt Manager

```

    gestaltPPCSupportsOutGoing      = 0x0002 /*can initiate sessions with */
                                        /* remote computers*/
};

enum {
    /*gestaltProcessorType response values*/
    gestalt68000                    = 1,    /*68000 microprocessor*/
    gestalt68010                    = 2,    /*68010 microprocessor*/
    gestalt68020                    = 3,    /*68020 microprocessor*/
    gestalt68030                    = 4,    /*68030 microprocessor*/
    gestalt68040                    = 5     /*68040 microprocessor*/
};

enum {
    /*gestaltQuickdrawFeatures response bits*/
    gestaltHasColor                  = 0,    /*Color QuickDraw present*/
    gestaltHasDeepGWorlds           = 1,    /*graphics worlds can be */
                                        /* deeper than 1 bit*/
    gestaltHasDirectPixMaps         = 2,    /*PixMaps can be direct */
                                        /* (16- or 32-bit)*/
    gestaltHasGrayishTextOr         = 3,    /*supports text mode */
                                        /* grayishTextOr*/
    gestaltSupportsMirroring         = 4     /*supports video mirroring */
                                        /* using the Display Manager*/
};

enum {
    /*gestaltQuickdrawVersion response values*/
    gestaltOriginalQD                = 0x000, /*original 1-bit QuickDraw*/
    gestalt8BitQD                    = 0x100, /*8-bit QuickDraw*/
    gestalt32BitQD                   = 0x200, /*32-Bit QuickDraw vers. 1.0*/
    gestalt32BitQD11                 = 0x210, /*32-Bit QuickDraw vers. 1.1*/
    gestalt32BitQD12                 = 0x220, /*32-Bit QuickDraw vers. 1.2*/
    gestalt32BitQD13                 = 0x230 /*32-Bit QuickDraw vers. 1.3*/
};

enum {
    /*gestaltRealtimeAttr response bits*/
    gestaltRealtimeMgrPresent        = 0     /*Realtime Manager present*/
};

```

Gestalt Manager

```

enum {
    /*gestaltResourceMgrAttr response bits*/
    gestaltPartialRsrcs          = 0          /*partial resources supported*/
};

enum {
    /*gestaltScrapMgrAttr response bits*/
    gestaltScrapMgrTranslationAware = 0,      /*aware of Translation Manager*/
    gestaltTrasnlationMgrHintOrder  = 1      /*hint order reversal present*/
};

enum {
    /*gestaltSerialAttr response bits*/
    gestaltHasGPIaToDCDa          = 0,        /*GPI connected to DCD on */
                                           /* port A*/
    gestaltHasGPIaToRTxCa         = 1,        /*GPI connected to RTxC on */
                                           /* port A*/
    gestaltHasGPIbToDCDb         = 2,        /*GPI connected to DCD on */
                                           /* port B*/
};

enum {
    /*gestaltSoundAttr response bits*/
    gestaltStereoCapability        = 0,        /*stereo capability present*/
    gestaltStereoMixing            = 1,        /*stereo mixing on internal */
                                           /* speaker*/
    gestaltSoundIOMgrPresent       = 3,        /*sound input routines present*/
    gestaltBuiltInSoundInput       = 4,        /*built-in input device */
                                           /* present*/
    gestaltHasSoundInputDevice     = 5,        /*sound input device present*/
    gestaltPlayAndRecord           = 6,        /*built-in hardware can play */
                                           /* and record simultaneously*/
    getstalt16BitSoundIO           = 7,        /*sound hardware can play and */
                                           /* record 16-bit samples*/
    getstaltStereoInput            = 8,        /*sound hardware can */
                                           /* record steore*/
    getstaltSndPlayDoubleBuffer    = 10,      /*SndPlayDouble buffer present*/
    getstaltMultiChannels          = 11,      /*multiple channel support*/
    getstalt16BitAudioSuuport      = 12      /*16-bit audio data supported*/
};

enum {
    /*gestaltSpeechAttr response bits*/
    gestaltSpeechMgrPresent        = 0,        /*Speech Manager present*/
};

```

Gestalt Manager

```

    gestaltSpeechHasPPCGLue          = 1      /*Speech Manager has native *
                                           /* PPC glue for API*/
};

enum {
    /*gestaltStandardFileAttr response bits*/
    gestaltStandardFile58             = 0,    /*has functions new with 7.0*/
    gestaltStandardFileTranslationAware = 1,  /*aware of Translation Manager*/
    gestaltStandardFileHasColorIcons  = 2    /*dialog boxes use small */
                                           /* color icons*/
};

enum {
    /*gestaltStdNBPAAttr response bits*/
    gestaltStdNBPPresent              = 0      /*StandardNBP is present*/
};

enum {
    /*gestaltSysArchitecture response bits*/
    gestalt68k                       = 1,     /*MC680x0 architecture*/
    gestaltPowerPC                    = 2     /*PowerPC architecture*/
};

enum {
    /*gestaltTEAttr response bits*/
    gestaltTEHasGetHiliteRgn         = 0      /*TextEdit has TEGetHiliteRgn*/
};

enum {
    /*gestaltTermMgrAttr response bits*/
    gestaltTermMgrPresent            = 0,     /*Terminal Manager present*/
    gestaltTermMgrErrorString        = 2     /*supports error string */
                                           /* function*/
};

enum {
    /*gestaltTextEditVersion response codes */
    gestaltTE1                       = 1,     /*in MacIIci ROM*/
    gestaltTE2                       = 2,     /*with 6.0.4 scripts on */
                                           /* MacIIci*/
    gestaltTE3                       = 3,     /*with 6.0.4 scripts on*/
                                           /* other machines*/
    gestaltTE4                       = 4,     /*in 6.0.5 and 7.0*/
    gestaltTE5                       = 5     /*TextWidthHook available*/
};

```

Gestalt Manager

```

enum {
    /*gestaltThreadMgrAttr response bits*/
    gestaltThreadMgrPresent      = 0,      /*Thread Manager present*/
    gestaltSpecificMatchSupports = 1      /*Thread Manager supports */
                                        /* exact match creation option*/
};

enum {
    /*gestaltTimeMgrVersion response codes*/
    gestaltStandardTimeMgr      = 1,      /*standard Time Manager*/
    gestaltRevisedTimeMgr      = 2,      /*revised Time Manager*/
    gestaltExtendedTimeMgr     = 3      /*extended Time Manager*/
};

enum {
    /*gestaltTranslationAttr response codes*/
    gestaltTranslationMgrExists = 0      /*Translation Manager present*/
};

enum {
    /*gestaltVMAttr response bits*/
    gestaltVMPresent           = 0      /*virtual memory present*/
};

```

Informational Selector Response Values

```

enum {
    /*gestaltHardwareAttr response bits*/
    gestaltHasVIA1              = 0,      /*has VIA1 chip*/
    gestaltHasVIA2              = 1,      /*has VIA2 chip*/
    gestaltHasASC                = 3,      /*has Apple Sound Chip*/
    gestaltHasSCC                = 4,      /*has SCC*/
    gestaltHasSCSI               = 7,      /*has SCSI*/
    gestaltHasSoftPowerOff      = 19,     /*capable of software power */
                                        /* off*/
    gestaltHasSCSI961           = 21,     /*has 53C96 SCSI on internal */
                                        /* bus*/
    gestaltHasSCSI962           = 22,     /*has 53C96 SCSI on external */
                                        /* bus*/
    gestaltHasUniversalROM      = 24      /*has universal ROM*/
};

```

Gestalt Manager

```

enum {
    /*gestaltMachineType response codes*/
    gestaltClassic          = 1,      /*Macintosh 128K*/
    gestaltMacXL            = 2,      /*Macintosh XL*/
    gestaltMac512KE         = 3,      /*Macintosh 512K enhanced*/
    gestaltMacPlus          = 4,      /*Macintosh Plus*/
    gestaltMacSE            = 5,      /*Macintosh SE*/
    gestaltMacII            = 6,      /*Macintosh II*/
    gestaltMacIIX           = 7,      /*Macintosh IIX*/
    gestaltMacIICX          = 8,      /*Macintosh IICX*/
    gestaltMacSE030         = 9,      /*Macintosh SE/30*/
    gestaltPortable         = 10,     /*Macintosh Portable*/
    gestaltMacIICI          = 11,     /*Macintosh IICI*/
    gestaltMacIIFX          = 13,     /*Macintosh IIFX*/
    gestaltMacClassic       = 17,     /*Macintosh Classic*/
    gestaltMacIISI          = 18,     /*Macintosh IISI*/
    gestaltMacLC            = 19,     /*Macintosh LC*/
    gestaltQuadra900        = 20,     /*Macintosh Quadra 900*/
    gestaltPowerBook170     = 21,     /*Macintosh PowerBook 170*/
    gestaltQuadra700        = 22,     /*Macintosh Quadra 700*/
    gestaltClassicII        = 23,     /*Macintosh Classic II*/
    gestaltPowerBook100     = 24,     /*Macintosh PowerBook 100*/
    gestaltPowerBook140     = 25,     /*Macintosh PowerBook 140*/
    gestaltQuadra950        = 26,     /*Macintosh Quadra 950*/
    gestaltMacLClII        = 27,     /*Macintosh LC III*/
    gestaltPowerBook210     = 29,     /*Macintosh PowerBook Duo 210*/
    gestaltMacCentris650    = 30,     /*Macintosh Centris 650*/
    gestaltPowerBook230     = 32,     /*Macintosh PowerBook Duo 230*/
    gestaltPowerBook180     = 33,     /*Macintosh PowerBook 180*/
    gestaltPowerBook160     = 34,     /*Macintosh PowerBook 160*/
    gestaltMacQuadra800     = 35,     /*Macintosh Quadra 800*/
    gestaltMacLClII        = 37,     /*Macintosh LC II*/
    gestaltPowerBookDuo250  = 38,     /*Macintosh PowerBook Duo 230*/
    gestaltMacIIVI          = 44,     /*Macintosh IIVI*/
    gestaltPerforma600      = 45,     /*Macintosh Performa 600*/
    gestaltMacIIVX          = 48,     /*Macintosh IIVX*/
    gestaltMacColorClassic  = 49,     /*Macintosh Color Classic*/
    gestaltPowerBook165c    = 50,     /*Macintosh PowerBook 165c*/
    gestaltMacCentris610    = 52,     /*Macintosh Centris 610*/
    gestaltMacQuadra610     = 53,     /*Macintosh Quadra 610*/
    gestaltPowerBook145     = 54,     /*Macintosh PowerBook 145*/
    getstaltMacLC520        = 56,     /*Macintosh LC 520*/
    getstaltMacCentris660AV = 60,     /*Macintosh Centris 660 AV*/
}

```

Gestalt Manager

```

getstaltPowerBook180c           = 71,    /*Macintosh PowerBook 180c*/
getstaltPowerBookDuo270c       = 77,    /*Macintosh PowerBook Duo 270c*/
getstaltMacQuadra840AV        = 78,    /*Macintosh Quadra 840 AV*/
getstaltPowerBook165          = 84,    /*Macintosh PowerBook 165*/
getstaltMacTV                  = 88,    /*Macintosh TV*/
getstaltMacLC475               = 89,    /*Macintosh LC 475*/
getstaltMacLC575               = 92,    /*Macintosh LC 575*/
getstaltMacQuadra605          = 94,    /*Macintosh Quadra 605*/

getstaltPowerMac8100_80        = 65,    /*Power Macintosh 8100/80*/
getstaltPowerMac6100_60        = 75,    /*Power Macintosh 6100/60*/
getstaltPowerMac7100_66        = 112   /*Power Macintosh 7100/66*/

};

enum {
    kMachineNameStrID           = -16395 /*'STR#' resource that */
};
/* contains machine names*/

```

SysEnviron Constants

```

enum {
    curSysEnvVers              = 2      /*current SysEnviron version*/
};

enum {
    /*machine types*/
    envXL                      = -2,    /*Macintosh XL*/
    envMac                     = -1,    /*Macintosh with 64K ROM*/
    envMachUnknown             = 0,    /*unknown model, after */
    /* Macintosh IIfx*/
    env512KE                   = 1,    /*Macintosh 512K enhanced*/
    envMacPlus                 = 2,    /*Macintosh Plus*/
    envSE                      = 3,    /*Macintosh SE*/
    envMacII                   = 4,    /*Macintosh II*/
    envMacIIX                  = 5,    /*Macintosh IIX*/
    envMacIICx                 = 6,    /*Macintosh IICx*/
    envSE30                    = 7,    /*Macintosh SE30*/
    envPortable                = 8,    /*Macintosh Portable*/
    envMacIICi                 = 9,    /*Macintosh IICi*/
    envMacIIfx                 = 11,   /*Macintosh IIfx*/
    envMacClassic              = 15,   /*Macintosh Classic*/
    envMacIIsi                 = 16,   /*Macintosh IIsi*/
    envMacLC                   = 17,   /*Macintosh LC*/
}

```

Gestalt Manager

```

envMacQuadra900          = 18,    /*Macintosh Quadra 900*/
envMacPowerBook170      = 19,    /*Macintosh PowerBook 170*/
envMacQuadra700         = 20,    /*Macintosh Quadra 700*/
envMacClassicII        = 21,    /*Macintosh Classic II*/
envMacPowerBook100     = 22,    /*Macintosh PowerBook 100*/
envMacPowerBook140     = 23,    /*Macintosh PowerBook 140*/
envMacQuadra950        = 24,    /*Macintosh Quadra 950*/
envMacLCII              = 35,    /*Macintosh LC II*/
envMacPowerBook145     = 52     /*Macintosh PowerBook 145*/
};

enum {
    /*CPU types*/
    envCPUUnknown        = 0,     /*unknown microprocessor*/
    env68000              = 1,     /*68000 microprocessor*/
    env68010              = 2,     /*68010 microprocessor*/
    env68020              = 3,     /*68020 microprocessor*/
    env68030              = 4,     /*68030 microprocessor*/
    env68040              = 5,     /*68040 microprocessor*/
};

enum {
    /*keyboard types*/
    envUnknownKbd        = 0,     /*Macintosh Plus with keypad*/
    envMacKbd            = 1,     /*Macintosh*/
    envMacAndPad         = 2,     /*Macintosh with keypad*/
    envMacPlusKbd        = 3,     /*Macintosh Plus*/
    envAExtendKbd        = 4,     /*Apple extended*/
    envStandADBKbd       = 5,     /*standard ADB*/
    envPrtblADBKbd       = 6,     /*Macintosh Portable ADB*/
    envPrtblISOKbd       = 7,     /*Macintosh Portable ISO*/
    envStdISOADBKbd      = 8,     /*standard ISO ADB */
    envExtISOADBKbd      = 9,     /*extended ISO ADB*/
};

```

Data Types

```

struct SysEnvRec {
    short      environsVersion;    /*system environment record*/
    short      machineType;        /*SysEnviron version number*/
    short      systemVersion;      /*Macintosh model code*/
    short      processor;          /*System file version number*/
    Boolean     hasFPU;             /*microprocessor type code*/
    Boolean     hasFPU;             /*floating-point unit flag*/
};

```


Gestalt Manager

```

Boolean      hasColorQD;          /*Color QuickDraw flag*/
short        keyBoardType;        /*keyboard type code*/
short        atDrvrVersNum;       /*AppleTalk driver version number*/
short        sysVRefNum           /*working-directory reference */
                                     /* number of folder or volume */
                                     /* containing open System file*/
};

typedef struct SysEnvRec SysEnvRec;

```

Gestalt Manager Routines

Getting Information About the Operating Environment

```

pascal OSErr Gestalt      (OSType selector, long *response);
pascal OSErr SysEnvirons (short versionRequested, SysEnvRec *theWorld);

```

Adding a Selector Code

```

pascal OSErr NewGestalt (OSType selector,
                        SelectorFunctionUUP gestaltFunction);

```

Modifying a Selector Function

```

pascal OSErr ReplaceGestalt
                        (OSType selector,
                        SelectorFunctionUUP gestaltFunction,
                        SelectorFunctionUUP *oldGestaltFunction);

```

Application-Defined Routines

```

pascal OSErr MySelectorFunc
                        (OSType selector, long *response);

```

Assembly-Language Summary

Data Structures

SysEnvRec Data Structure

0	environsVersion	word	SysEnvironments version number
2	machineType	word	Macintosh model code
4	systemVersion	word	System file version number
6	processor	word	microprocessor type code
8	hasFPU	byte	floating-point unit flag
9	hasColorQD	byte	Color QuickDraw flag
10	keyBoardType	word	keyboard type code
12	atDrvrsVersNum	word	AppleTalk driver version number
14	sysVRefNum	word	working-directory reference number of directory or volume containing open System file

Result Codes

noErr	0	No error
memFullErr	-108	Ran out of memory
envNotPresent	-5500	SysEnvironments trap not present
envBadVers	-5501	Nonpositive version number passed
envVersTooBig	-5502	Requested version of SysEnvironments not available
gestaltUnknownErr	-5550	Could not obtain the response
gestaltUndefSelectorErr	-5551	Undefined selector
gestaltDupSelectorErr	-5552	Selector already exists
gestaltLocationErr	-5553	Function not in system heap