MSDN Home > MSDN Library > .NET Development >

Microsoft Win32 to Microsoft .NET Framework API Map

Page Options

Average rating:
8 out of 9

Rate this page

Print this page

E-mail this page

Add to Favorites

Microsoft .NET Framework and Microsoft Visual Studio User Education Teams
Microsoft Corporation

January 2004

Applies to:
- Microsoft® .NET Framework version 1.0 or 1.1
- Microsoft Visual Studio® .NET
- Microsoft Win32® functions

Summary: This article identifies the Microsoft .NET Framework version 1.0 or 1.1 APIs that provide similar functionality to Microsoft Win32 functions. One or more relevant .NET Framework APIs are shown for each Win32 function listed. The intended audience for this article is experienced Win32 developers who are creating applications or libraries based on the Microsoft .NET Framework, but anyone looking for a managed counterpart for a Win32 function could find this document useful. (85 printed pages)

In this article, the Microsoft® Win32® functions are organized by category. Each Win32 function that is listed belongs to only one category, and each category is presented in its own table. An alphabetical list of the Win32 categories is provided, as well as a hierarchical list. To locate a Win32 function, select the appropriate category from one of the category lists or search the article for the name of the function. If you do not find the Win32 function that you are seeking, it means that no equivalent .NET Framework API exists in the .NET Framework version 1.0 or 1.1 or that the Win32 function and the relevant .NET Framework API were inadvertently omitted from the list.

If you believe that a Win32 function and the relevant .NET Framework API have been omitted from this article, please let us know by using the feedback link. For more information on the APIs listed in this article, see the Platform SDK documentation or the .NET Framework SDK documentation in the MSDN® library.

Alphabetical Win32 Category List

A-C

Atom functions	Color functions
Bitmap functions	Combo box functions
Brush functions	Common dialog box functions
Button functions	Console functions
Clipboard functions	Coordinate functions
Clipping functons	Cursor functions

D-I

Debugging functions	File I/O functions
Device context functions	File system functions
Dialog box functions	Filled shape functions
Dynamic-link library functions	Font and text functions
Error handling functions	Icon functions
Event logging functions	Input Method Editor functions

J-N

Keyboard accelerator functions	Message and message queue functions
Keyboard input functions	Metafile functions
Line and curve functions	Mouse input functons
List box functions	Multiple display monitor functions

Memory management functions	Multiple-document interface functions
Menu functions	National language functions

O-R

Painting and drawing functions	PSAPI functions
Path functions	Rectangle functions
Pen functions	Region functions
Performance monitoring functions	Registry functions
Printing functions	Resource functions
Process and thread functions	Rich edit control functions

S-T

Scroll bar functions	Tape backup functions
Service functions	Text object model functions
String functions	Time functions
Structured exception-handling functions	Timer functions
Synchronization functions	Tool Help functions
System information functions	

U-Z

Unicode and character set functions	Window functions
User profile functions	Window class functions
Version information functions	Window procedure functions
	Window property functions

Hierarchical Win32 Category List

Base Services

Debugging and Error Handling	**Miscellaneous**
Debugging	Memory management
Error handling	User profile
Event logging	**Performance**
Structured exception-handling	Performance monitoring
DLLs, Processes, and Threads	PSAPI
Console applications	Tool Help
Dynamic-link library	**Windows System Information**
Process and thread	Registry
Service	System information
Synchronization	Time
Files and I/O	
File I/O	
File system	
Tape backup	

Graphics Device Interface

Bitmap	Metafile
Brush	Multiple display monitor
Clipping	National language
Color	Painting and drawing
Coordinate	Path
Device context	Pen
Filled shape	Printing and print spooler
Font and text	Rectangle

Input Method Editor	Region
Line and curve	Unicode and character set

User Interface

Controls	**User Input**
Button	Common dialog box
Combo box	Keyboard accelerator
List box	Keyboard input
Rich edit control	Mouse input
Scroll bar	**Windowing**
Text object model	Dialog box
Data Exchange	Message and message queue
Atom	Multiple-document interface
Clipboard	Timer
Resources	Window
Cursor	Window class
Icon	Window procedure
Menu	Window property
Resource	
String	
Version information	

Atom Functions

Win32 function	**Description**	**.NET Framework API**
AddAtom	Adds a character string to the local atom table.	**System.String.Intern** **Note** String is not exactly an atom.
FindAtom	Searches the local atom table for the specified character string.	**System.String.IsInterned** **Note** String is not exactly an atom.

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Bitmap Functions

Win32 function	**Description**	**.NET Framework API**
AlphaBlend	Displays a bitmap with transparent or semitransparent pixels.	**System.Drawing.Graphics.DrawImage**
BitBlt	Performs a bit-block transfer.	**System.Drawing.Graphics.DrawImage**
CreateBitmap	Creates a bitmap.	**System.Drawing.Bitmap constructor** **Note** Creates a DIB, not a DDB.
GetBitmapDimensionEx	Gets the dimensions of a bitmap.	**System.Drawing.Bitmap.Size**
GetDIBColorTable	Retrieves RGB color values from a DIB section bitmap.	**System.Drawing.Bitmap.Palette**
GetDIBits	Copies a bitmap into a buffer.	**System.Drawing.Bitmap.LockBits**
GetPixel	Gets the RGB color value of the pixel at a given coordinate.	**System.Drawing.Bitmap.GetPixel**
GradientFill	Fills rectangle and triangle structures.	**System.Drawing.Graphics.FillRectangle (Drawing2D.LinearGradiantBrush, Rectangle)**
LoadBitmap	Loads a bitmap from a module's executable file.	**System.Drawing.Bitmap.Bitmap(Type, String)**
SetPixel	Sets the color for a pixel.	**System.Drawing.Bitmap.SetPixel**
SetPixelV	Sets a pixel to the best approximation of a color.	**System.Drawing.Bitmap.SetPixel(…, Graphics.GetNearestColor(Color))**
StretchBlt	Copies a bitmap and stretches or compresses it.	**System.Drawing.Graphics.DrawImage**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Brush Functions

Win32 function	Description	.NET Framework API
CreateBrushIndirect	Creates a brush with a specified style, color, and pattern.	Any of the following: **System.Drawing.Brushes** **System.Drawing.SolidBrush** **System.Drawing.TextureBrush** **System.Drawing.Drawing2D.HatchBrush** **System.Drawing.Drawing2D.LinearGradientBrush** **System.Drawing.Drawing2D.PathGradientBrush**
CreateHatchBrush	Creates a brush with a hatch pattern and color.	**System.Drawing.Drawing2D.HatchBrush constructor**
CreatePatternBrush	Creates a brush with a bitmap pattern.	**System.Drawing.TextureBrush constructor**
CreateSolidBrush	Creates a brush with a solid color.	**System.Drawing.SolidBrush constructor**
GetSysColorBrush	Gets a handle for a brush that corresponds to a color index.	**System.Drawing.Brushes class**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Button Functions

Win32 function	Description	.NET Framework API
CheckDlgButton	Changes the check state of a button control.	**System.Windows.Forms.CheckBox.Checked**
CheckRadioButton	Adds a check mark to a specified radio button in a group and removes a check mark from all other radio buttons in the group.	**System.Windows.Forms.RadioButton.Checked**
IsDlgButtonChecked	Determines whether a button control has a check mark next to it or whether a three-state button control is grayed, checked, or neither.	**System.Windows.Forms.CheckBox.CheckState**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Clipboard Functions

Win32 function	Description	.NET Framework API
CountClipboardFormats	Retrieves the number of different data formats currently on the clipboard.	**System.Windows.Forms.DataObject.GetFormats**
EnumClipboardFormats	Enumerates the data formats currently available on the clipboard.	**System.Windows.Forms.DataObject.GetFormats**
GetClipboardData	Retrieves data from the clipboard in a specified format.	**System.Windows.Forms.ClipBoard.GetDataObject**
IsClipboardFormatAvailable	Determines whether the clipboard contains data in the specified format.	**System.Windows.Forms.DataObject.GetDataPresent**
RegisterClipboardFormat	Registers a new clipboard format.	**System.Windows.Forms.DataFormats.GetFormat**
SetClipboardData	Places data on the clipboard in a specified clipboard format.	**System.Windows.Forms.ClipBoard.SetDataObject**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Clipping Functions

Win32 function	Description	.NET Framework API
ExcludeClipRect	Creates a new clipping region that consists of the existing clipping region minus the specified rectangle.	**System.Drawing.Graphics.ExcludeClip**
ExtSelectClipRgn	Combines the specified region with the current clipping region using the specified mode.	**System.Drawing.Graphics.SetClip**
GetClipBox	Retrieves the dimensions of the tightest bounding rectangle that can be drawn around the current visible area on the device.	**System.Drawing.Graphics.VisibleClipBounds**
GetClipRgn	Retrieves a handle identifying the current application-defined clipping region for the specified device context.	**System.Drawing.Graphics.Clip**
IntersectClipRect	Creates a new clipping region from the intersection of the current clipping region and the specified rectangle.	**System.Drawing.Graphics.IntersectClip**
OffsetClipRgn	Moves the clipping region of a device context by the specified offsets.	**System.Drawing.Graphics.TranslateClip**
PtVisible	Determines whether the specified point is within the clipping region of a device context.	**System.Drawing.Graphics.IsVisible**
RectVisible	Determines whether any part of the specified rectangle lies within the clipping region of a device context.	**System.Drawing.Graphics.IsVisible**
SelectClipPath	Selects the current path as a clipping region for a device context, combining the new region with any existing clipping region by using the specified mode.	**System.Drawing.Graphics.SetClip**
SelectClipRgn	Selects a region as the current clipping region for the specified device context.	**System.Drawing.Graphics.Clip**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Color Functions

Win32 function	Description	.NET Framework API
CreateHalftonePalette	Creates a halftone palette for the specified device context.	**System.Drawing.Graphics.GetHalftonePalette**
CreatePalette	Creates a logical palette.	**System.Drawing.Image.Palette**
GetNearestColor	Retrieves a color value identifying a color from the system palette that will be displayed when the specified color value is used.	**System.Drawing.Graphics.GetNearestColor**
GetPaletteEntries	Retrieves a specified range of palette entries from the given logical palette.	**System.Drawing.Image.Palette.Entries**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Combo Box Functions

Win32 function	Description	.NET Framework API
DlgDirListComboBox	Replaces the contents of a combo box with the names of specified subdirectories and files.	**System.Windows.Forms.ComboBox.Items.AddRange (System.IO.Directory.GetFileSystemEntries)**

DlgDirSelectComboBoxEx	Retrieves the current selection from a combo box filled by using **DlgDirListComboBox**.	**System.Windows.Forms.ComboBox.SelectedItem**
GetComboBoxInfo	Retrieves information about a specified combo box.	**System.Windows.Forms.ComboBox** properties

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Common Dialog Box Functions

Win32 function	Description	.NET Framework API
ChooseColor	Creates a **Color** dialog box that enables the user to select a color.	**System.Windows.Forms.ColorDialog** constructor **Note** Dialog cannot be modified through a hook.
ChooseFont	Creates a **Font** dialog box that enables the user to choose attributes for a logical font.	**System.Windows.Forms.FontDialog** constructor **Note** Dialog cannot be modified through a hook.
GetFileTitle	Retrieves the name of the specified file.	**System.IO.FileInfo.Name** **Note** Does not take user preferences into account.
GetOpenFileName	Creates an **Open** dialog box that lets the user specify the drive, directory, and name of a file or set of files to open.	**System.Windows.Forms.OpenFileDialog.ShowDialog** **Note** Dialog cannot be modified through a hook.
GetSaveFileName	Creates a **Save** dialog box that lets the user specify the drive, directory, and name of a file to save.	**System.Windows.Forms.SaveFileDialog.ShowDialog** **Note** Dialog cannot be modified through a hook.
PageSetupDlg	Creates a **Page Setup** dialog box that enables the user to specify the attributes of a printed page.	**System.Windows.Forms.PageSetupDialog.ShowDialog** **Note** Dialog cannot be modified through a hook.
PrintDlg	Displays a **Print** dialog box.	**System.Windows.Forms.PrintDialog.ShowDialog** **Note** Dialog cannot be modified through a hook.

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Console Functions

Win32 function	Description	.NET Framework API
GetStdHandle	Retrieves a handle for the standard input, standard output, or standard error device.	**System.Console.OpenStandardIn** **System.Console.OpenStandardOut** **System.Console.OpenStandardError**
ReadConsole	Reads character input from the console input buffer and removes it from the buffer.	**System.Console.Read**
SetStdHandle	Sets the handle for the standard input, standard output, or standard error device.	**System.Console.SetIn** **System.Console.SetOut** **System.Console.SetError**
WriteConsole	Writes a character string to a console screen buffer beginning at the current cursor location.	**System.Console.Write**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Coordinate Functions

Win32 function	Description	.NET Framework API
ClientToScreen	Converts the client-area coordinates of a specified point to screen coordinates.	**System.Windows.Forms.Form.PointToScreen**
CombineTransform	Concatenates two world-space to	**System.Drawing.Drawing2D.Matrix.Multiply**

	page-space transformations.	
DPtoLP	Converts device coordinates into logical coordinates.	**System.Drawing.Graphics.TransformPoints**
GetMapMode	Retrieves the current mapping mode.	**System.Drawing.Graphics.PageUnit**
GetWindowExtEx	Retrieves the x-extent and y-extent of the window for the specified device context.	**System.Drawing.Graphics.PageScale**
GetWindowOrgEx	Retrieves the x-coordinates and y-coordinates of the window origin for the specified device context.	**System.Drawing.Graphics.Transform.OffsetX** **System.Drawing.Graphics.Transform.OffsetY**
GetWorldTransform	Retrieves the current world-space to page-space transformation.	**System.Drawing.Graphics.Transform**
LPtoDP	Converts logical coordinates into device coordinates.	**System.Drawing.Graphics.TransformPoints**
ModifyWorldTransform	Changes the world transformation for a device context using the specified mode.	**System.Drawing.Graphics.Transform**
OffsetViewportOrgEx	Modifies the viewport origin for a device context using the specified horizontal and vertical offsets.	**System.Drawing.Graphics.TranslateTransform**
OffsetWindowOrgEx	Modifies the window origin for a device context using the specified horizontal and vertical offsets.	**System.Drawing.Graphics.TranslateTransform**
ScreenToClient	Converts the screen coordinates of a specified point on the screen-to-client coordinates.	**System.Windows.Forms.Form.PointToClient**
SetMapMode	Sets the mapping mode of the specified device context.	**System.Drawing.Graphics.PageUnit**
SetWindowExtEx	Sets the horizontal and vertical extents of the window for a device context by using the specified values.	**System.Drawing.Graphics.PageScale** **Drawing.Graphics.ScaleTransform**
SetWindowOrgEx	Specifies which window point maps to the viewport origin (0,0).	**System.Drawing.Graphics.TranslateTransform**
SetWorldTransform	Sets a two-dimensional linear transformation between world space and page space for the specified device context.	**System.Drawing.Graphics.Transform**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Cursor Functions

Win32 function	Description	.NET Framework API
ClipCursor	Confines the cursor to a rectangular area on the screen.	**System.Windows.Forms.Cursor.Clip**
CopyCursor	Copies the specified cursor.	**System.Windows.Forms.Cursor.CopyHandle**
CreateCursor	Creates a cursor having the specified size, bit patterns, and hot spot.	**System.Windows.Forms.Cursor** constructor **System.Windows.Forms.Cursor.Size** **System.Windows.Forms.Cursor.Position**
DestroyCursor	Destroys a cursor.	**System.Windows.Forms.Cursor.Dispose**
GetClipCursor	Retrieves the screen coordinates of the rectangular area to which the cursor is confined.	**System.Windows.Forms.Cursor.Clip**
GetCursor	Retrieves a handle to the current cursor.	**System.Windows.Forms.Cursor.Handle**
GetCursorInfo	Retrieves information about the global cursor.	**System.Windows.Forms.Cursor.Position** **System.Windows.Forms.SystemInformation.CursorSize**
GetCursorPos	Retrieves the cursor's position.	**System.Windows.Forms.Control.MousePosition**

		System.Windows.Forms.Cursor.Position
LoadCursor	Loads a cursor resource from an executable file.	**System.Windows.Forms.Cursor** constructor
LoadCursorFromFile	Creates a cursor based on data contained in a file.	**System.Windows.Forms.Cursor** constructor
SetCursor	Sets the cursor shape.	**System.Windows.Forms.Cursor.Current**
SetCursorPos	Moves the cursor to the specified screen coordinates.	**System.Windows.Forms.Cursor.Position**
ShowCursor	Displays or hides the cursor.	**System.Windows.Forms.Cursor.Show** **Windows.Forms.Cursor.Hide**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Debugging Functions

Win32 function	Description	.NET Framework API
DebugActiveProcess	Enables a debugger to attach to an active process and debug it.	**System.Diagnostics.Debugger.Launch**
DebugBreak	Causes a breakpoint exception to occur in the current process.	**System.Diagnostics.Debugger.Break**
IsDebuggerPresent	Determines whether the calling process is running under the context of a debugger.	**System.Diagnostics.Debugger.IsAttached**
OutputDebugString	Sends a string to the debugger for display.	**System.Diagnostics.Debugger.Log**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Device Context Functions

Win32 function	Description	.NET Framework API
ChangeDisplaySettings	Changes the settings of the default display device to the specified graphics mode.	**System.Management.ManagementClass ("Win32_VideoController")**
ChangeDisplaySettingsEx	Changes the settings of the specified display device to the specified graphics mode.	**System.Management.ManagementClass ("Win32_VideoController")**
DeviceCapabilities	Retrieves the capabilities of a printer device driver.	**System.Drawing.Printing.PageSettings** **System.Drawing.Printing.PrinterSettings**
EnumDisplayDevices	Retrieves information about the display devices in a system.	**System.Management.ManagementClass ("Win32_DesktopMonitor")**
EnumDisplaySettings	Retrieves information about one of the graphics modes for a display device.	**System.Management.ManagementClass ("CIM_VideoControllerResolution")**
GetDC	Retrieves a handle to a display device context for the client area of a specified window or for the entire screen.	To retrieve a **Graphics** object, use any of the following: **System.Drawing.Graphics.FromHwnd (Windows.Forms.Control.Handle)** **System.Drawing.Printing.PrintPageEventArgs.Graphics** **System.Windows.Forms.PaintEventArgs.Graphics** To explicitly retrieve a handle, use the following: **System.Drawing.Graphics.GetHdc**
GetDCEx	Retrieves a handle to a display device context for the client area of a specified window or for the entire screen.	To retrieve a **Graphics** object, use any of the following: **System.Drawing.Graphics.FromHwnd (Windows.Forms.Control.Handle)** **System.Drawing.Printing.PrintPageEventArgs.Graphics** **System.Windows.Forms.PaintEventArgs.Graphics** To explicitly retrieve a handle, use the following:

		System.Drawing.Graphics.GetHdc
GetDeviceCaps	Retrieves device-specific information for the specified device.	**System.Drawing.Graphics** properties **System.Drawing.Printing.PrinterSettings** **System.Management** classes
GetLayout	Retrieves the layout of a device context.	**System.Windows.Forms.Control.RightToLeft**
GetObject	Retrieves information for the specified graphics object.	**System.Drawing.Bitmap** properties **System.Drawing.Brush** properties **System.Drawing.Font** properties **System.Drawing.Pen** properties
GetObjectType	Retrieves the type of the specified object.	**System.Object.GetType**
GetStockObject	Retrieves a handle to one of the stock pens, brushes, fonts, or palettes.	**System.Drawing.Brushes** **System.Drawing.Pens**
ReleaseDC	Releases a device context, freeing it for use by other applications.	**System.Drawing.Graphics.ReleaseHdc**
SetLayout	Sets the layout for a device context.	**System.Windows.Forms.Control.RightToLeft = System.Windows.Forms.RightToLeft.Yes**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Dialog Box Functions

Win32 function	Description	.NET Framework API
CreateDialog	Creates a modeless dialog box from a dialog box template resource.	**System.Windows.Forms.Form** constructor **System.Windows.Forms.Form** properties
CreateDialogIndirect	Creates a modeless dialog box from a dialog box template in memory.	**System.Windows.Forms.Form** constructor **System.Windows.Forms.Form** properties
DefDlgProc	Calls the default dialog box window procedure to provide default processing for any window messages that a dialog box with a private window class does not process.	**System.Windows.Forms.Form.DefWndProc**
DialogBox	Creates a modal dialog box from a dialog box template resource.	**System.Windows.Forms.Form** constructor **System.Windows.Forms.Form.ShowDialog**
DialogBoxIndirect	Creates a modal dialog box from a dialog box template in memory.	**System.Windows.Forms.Form** constructor **System.Windows.Forms.Form.ShowDialog**
DialogProc	An application-defined callback function used with the **CreateDialog** and **DialogBox** families of functions.	**System.Windows.Forms.Form.WndProc**
EndDialog	Destroys a modal dialog box.	**System.Windows.Forms.Form.Close**
GetDlgCtrlID	Retrieves the identifier of the specified control.	**System.Windows.Forms.Form.Controls.IndexOf**
GetDlgItem	Retrieves a handle to a control in the specified dialog box.	**System.Windows.Forms.Form.Controls.Item**

GetDlgItemInt	Translates the text of a specified control in a dialog box into an integer value.	**System.Int32.Parse(Form.Text)**
		System.UInt32.Parse(Form.Text)

GetDlgItemText	Retrieves the title or text associated with a control in a dialog box.	**System.Windows.Forms.Form.Controls.Item [].Text**
GetNextDlgGroupItem	Retrieves a handle to the first control in a group of controls that precedes (or follows) the specified control.	Iterate through the controls: **System.Windows.Forms.Form.Controls.Item** Remember to determine whether the control is active by referencing the **Enabled** and **Visible** properties of the control. **Note** The order that the controls are presented in

		might not be the same order in which **GetNextDlgGroupItem** presents the controls.
GetNextDlgTabItem	Retrieves a handle to the first control that has the WS_TABSTOP style that precedes (or follows) the specified control.	Iterate through the controls: **System.Windows.Forms.Form.Controls.Item** Remember to determine whether the control is active by referencing the **Enabled**, **Visible**, and **TabStop** properties of the control. **Note** The order that the controls are presented in might not be the same order in which **GetNextDlgGroupItem** presents the controls.
MessageBox	Creates, displays, and operates a message box.	**System.Windows.Forms.MessageBox.Show**
MessageBoxEx	Creates, displays, and operates a message box.	**System.Windows.Forms.MessageBox.Show**
MessageBoxIndirect	Creates, displays, and operates a message box.	**System.Windows.Forms.MessageBox.Show**
SendDlgItemMessage	Sends a message to the specified control in a dialog box.	**System.Windows.Forms.Form.WndProc (System.Windows.Forms.Message)**
SetDlgItemInt	Sets the text of a control in a dialog box to the string representation of a specified integer value.	**System.Windows.Forms.Control.Text = System.Int32.ToString**
SetDlgItemText	Sets the title or text of a control in a dialog box.	**System.Windows.Forms.Control.Text**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Dynamic-Link Library Functions

Win32 function	Description	.NET Framework API
GetModuleFileName	Retrieves the full path and file name for the file containing the specified module.	**System.Reflection.Module.FullyQualifiedName**
GetModuleHandle	Retrieves a module handle for the specified module.	**System.Runtime.InteropServices.Marshal.GetHINSTANCE**
LoadLibrary	Maps the specified executable module into the address space of the calling process.	Any of the following: **System.Reflection.Assembly.Load** **System.Reflection.Assembly.LoadFrom** **System.Reflection.Assembly.LoadWithPartialName**
LoadLibraryEx	Maps the specified executable module into the address space of the calling process.	Any of the following: **System.Reflection.Assembly.Load** **System.Reflection.Assembly.LoadFrom** **System.Reflection.Assembly.LoadWithPartialName**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Error Handling Functions

Win32 function	Description	.NET Framework API
FatalAppExit	Displays a message box and terminates the application when the message box is closed.	**System.Windows.Forms.MessageBox.Show** **System.Diagnostics.Process.Kill**
GetLastError	Retrieves the calling thread's last-error code value.	**System.Runtime.InteropServices.Marshall.GetLastWin32Error**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Event Logging Functions

Win32 function	Description	.NET Framework API
ClearEventLog	Clears the specified event log, and optionally saves the current copy of the log file to a backup file.	**System.Diagnostics.EventLog.Clear**
CloseEventLog	Closes a read handle to the specified event log.	**System.Diagnostics.EventLog.Close**
DeregisterEventSource	Closes a write handle to the specified event log.	**System.Diagnostics.EventLog.DeleteEventSource**
GetNumberOfEventLogRecords	Retrieves the number of records in the specified event log.	**System.Diagnostics.EventLog.Entries.Count**
GetOldestEventLogRecord	Retrieves the absolute record number of the oldest record in the specified event log.	**System.Diagnostics.EventLogEntryCollection[0]** **System.Diagnostics.EventLogEntry.TimeWritten**
NotifyChangeEventLog	Enables an application to receive notification when an event is written to the specified event log file.	**System.Diagnostics.EventLog.EnableRaisingEvents**
OpenBackupEventLog	Opens a handle to a backup event log.	**System.Diagnostics.EventLog** constructor
OpenEventLog	Opens a handle to an event log.	**System.Diagnostics.EventLog** constructor
ReadEventLog	Reads a whole number of entries from the specified event log.	**System.Diagnostics.EventLogEntryCollection.CopyTo**
RegisterEventSource	Retrieves a registered handle to an event log.	**System.Diagnostics.EventLog.CreateEventSource**
ReportEvent	Writes an entry at the end of the specified event log.	**System.Diagnostics.EventLog.WriteEntry**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

File I/O Functions

Win32 function	Description	.NET Framework API
CopyFile	Copies an existing file to a new file.	Either of the following: **System.IO.File.Copy** **IO.FileInfo.CopyTo**
CreateDirectory	Creates a new directory.	Any of the following: **System.IO.Directory.CreateDirectory** **System.IO.Directory.CreateSubdirectory** **System.IO.DirectoryInfo.Create**
CreateFile	Creates or opens a file object.	**System.IO.File** class and **System.IO.FileInfo** class
DeleteFile	Deletes an existing file.	Any of the following: **System.IO.Directory.Delete** **System.IO.DirectoryInfo.Delete** **System.IO.File.Delete** **System.IO.FileInfo.Delete**

FileIOCompletionRoutine	An application-defined callback function used with **ReadFileEx** and **WriteFileEx**.	User defined.
FindClose	Closes the specified search handle.	See **FindFirstFile** and **FindNextFile**
FindCloseChangeNotification	Stops change notification handle monitoring.	**System.IO.FileSystemWatcher.EnableRaisingEvents**
FindFirstChangeNotification	Creates a change notification handle.	**System.IO.FileSystemWatcher** class
FindFirstFile	Searches a directory for a file whose name matches the specified file name.	Any of the following: **System.IO.Directory.Exists** **System.IO.Directory.GetDirectories** **System.IO.Directory.GetFileSystemEntries** **System.IO.Directory.GetFiles** **System.IO.DirectoryInfo.Exists** **System.IO.DirectoryInfo.GetDirectories** **System.IO.DirectoryInfo.GetFileSystemInfos** **System.IO.DirectoryInfo.GetFiles** **System.IO.File.Exists** **System.IO.FileInfo.Exists**
FindFirstFileEx	Searches a directory for a file whose name and attributes match the specified file name.	Same as **FindFirstFile**
FindNextChangeNotification	Requests that the operating system signal a change notification handle the next time it detects an appropriate change.	**System.IO.FileSystemWatcher** class
FindNextFile	Continues a file search.	Any of the following: **System.IO.Directory.GetDirectories** **System.IO.Directory.GetFileSystemEntries** **System.IO.Directory.GetFiles** **System.IO.DirectoryInfo.GetDirectories** **System.IO.DirectoryInfo.GetFileSystemInfos** **System.IO.DirectoryInfo.GetFiles**
FlushFileBuffers	Clears the buffers for the specified file and causes all buffered data to be written to the file.	**System.IO.FileStream.Flush**
GetBinaryType	Determines whether a file is executable.	**System.IO.FileSystemInfo.Extension** **Note** Only manipulates strings; does not do a file-system check.
GetCurrentDirectory	Retrieves the current directory for the current process.	**System.IO.Directory.GetCurrentDirectory**
GetDriveType	Determines whether a disk drive is a removable, fixed, CD-ROM, RAM disk, or network drive.	**System.Management.ManagementObject ("Win32_LogicalDisk.DeviceID=\"C:\"").Get** **System.Management.ManagementObject.Properties** **System.IO.DriveInfo.DriveType**
GetFileAttributes	Retrieves attributes for a specified file or directory.	Any of the following: **System.IO.Directory.GetAttributes** **System.IO.File.GetAttributes** **System.IO.FileSystemInfo.Attributes**
GetFileAttributesEx	Retrieves attributes for a specified file or directory.	Any of the following: **System.IO.Directory.GetAttributes** **System.IO.File.GetAttributes** **System.IO.FileSystemInfo.Attributes**
GetFileInformationByHandle	Retrieves file information	**System.IO.FileInfo** properties

	for a specified file.	
GetFileSize	Retrieves the size of a specified file.	Either of the following: **System.IO.FileInfo.Length** **System.IO.FileStream.Length**
GetFileSizeEx	Retrieves the size of a specified file.	Either of the following: **System.IO.FileInfo.Length** **System.IO.FileStream.Length**
GetFullPathName	Retrieves the full path and file name of a specified file.	Either of the following: **System.IO.FileSystemInfo.FullName** **System.IO.Path.GetFullPath**
GetTempFileName	Creates a name for a temporary file.	**System.IO.Path.GetTempFileName**
GetTempPath	Retrieves the path of the directory designated for temporary files.	**System.IO.Path.GetTempPath**
LockFile	Locks a region in an open file.	**System.IO.FileStream.Lock**
LockFileEx	Locks a region in an open file for shared or exclusive access.	**System.IO.FileStream.Lock**
MoveFile	Moves an existing file or a directory.	Any of the following: **System.IO.Directory.Move** **System.IO.DirectoryInfo.MoveTo** **System.IO.File.Move** **System.IO.FileInfo.MoveTo**
ReadDirectoryChangesW	Retrieves information describing the changes occurring within a directory.	**System.IO.FileSystemWatcher** class
ReadFile	Reads data from a file, starting at the specified position.	Any of the following: **System.IO.FileStream.BeginRead/EndRead** **System.IO.FileStream.Read** **System.IO.FileStream.ReadByte**
ReadFileEx	Reads data from a file asynchronously.	**System.IO.FileStream.BeginRead/EndRead**
RemoveDirectory	Deletes an existing empty directory.	Either of the following: **System.IO.Directory.Delete** **System.IO.DirectoryInfo.Delete**
SearchPath	Searches for the specified file.	Any of the following: **System.IO.Directory.Exists** **System.IO.Directory.GetDirectories** **System.IO.Directory.GetFileSystemEntries** **System.IO.Directory.GetFiles** **System.IO.DirectoryInfo.Exists** **System.IO.DirectoryInfo.GetDirectories** **System.IO.DirectoryInfo.GetFileSystemInfos** **System.IO.DirectoryInfo.GetFiles** **System.IO.File.Exists** **System.IO.FileInfo.Exists**
SetCurrentDirectory	Changes the current directory for the current process.	**System.IO.Directory.SetCurrentDirectory**
SetEndOfFile	Moves the end-of-file position for the specified file.	**System.IO.FileStream.SetLength**
SetFileAttributes	Sets the attributes for a file.	**System.IO.File.SetAttributes**
SetFilePointer	Moves the file pointer of an open file.	**Either of the following:** **System.IO.FileStream.Position** **System.IO.FileStream.Seek**
SetFilePointerEx	Moves the file pointer of an open file.	Either of the following: **System.IO.FileStream.Position**

		System.IO.FileStream.Seek
UnlockFile	Unlocks a previously locked region in an open file.	**System.IO.FileStream.Unlock**
UnlockFileEx	Unlocks a previously locked region in an open file.	**System.IO.FileStream.Unlock**
WriteFile	Writes data to a file.	Any of the following: **System.IO.FileStream.BeginWrite** and **EndWrite** **System.IO.FileStream.Write** **System.IO.FileStream.WriteByte** **System.IO.BinaryWriter.Write**
WriteFileEx	Writes data to a file asynchronously.	**System.IO.FileStream.BeginWrite** and **EndWrite**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

File System Functions

Used with file systems:

Win32 function	Description	.NET Framework API
GetDiskFreeSpace	Retrieves information about the specified disk, including the amount of free space on the disk.	**System.Management.ManagementObject ("Win32_LogicalDisk.DeviceID=\"C:\"").Get** **System.Management.ManagementObject.Properties**
GetDiskFreeSpaceEx	Retrieves information about the specified disk, including the amount of free space on the disk.	**System.Management.ManagementObject ("Win32_LogicalDisk.DeviceID=\"C:\"").Get** **System.Management.ManagementObject.Properties**
GetLogicalDriveStrings	Fills a buffer with strings that specify valid drives in the system.	Either of the following: **System.Environment.GetLogicalDrives** **System.IO.Directory.GetLogicalDrives**

Used with an encrypted file system:

Win32 function	Description	.NET Framework API
DecryptFile	Decrypts an encrypted file or directory.	**System.Security.Cryptography.CryptoStream.Read**
EncryptFile	Encrypts a file or directory.	**System.Security.Cryptography.CryptoStream.Write**
FileEncryptionStatus	Retrieves the encryption status of the specified file.	**System.IO.FileSystemInfo.Attributes**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Filled Shape Functions

Win32 function	Description	.NET Framework API
Chord	Draws an area bounded by an ellipse and a line segment.	**System.Drawing.Graphics.DrawArc** **System.Drawing.Graphics.DrawLine**
Ellipse	Draws an ellipse.	**System.Drawing.Graphics.DrawEllipse** **System.Drawing.Graphics.FillEllipse**
FillRect	Fills a rectangle using a brush.	**System.Drawing.Graphics.FillRectangle**
InvertRect	Inverts the color values of the pixels in a rectangle.	**System.Windows.Forms.ControlPaint.DrawReversibleFrame** **System.Windows.Forms.ControlPaint.DrawReversibleLine**
Pie	Draws a pie-shaped wedge bounded by an ellipse and two radials.	**System.Drawing.Graphics.DrawPie** **System.Drawing.Graphics.FillPie**
Polygon	Draws a polygon.	**System.Drawing.Graphics.DrawPolygon** **System.Drawing.Graphics.FillPolygon**

Rectangle	Draws a rectangle.	**System.Drawing.Graphics.DrawRectangle** **System.Drawing.Graphics.FillRectangle**
RoundRect	Draws a rectangle with rounded corners.	**System.Drawing.Drawing2D.GraphicsPath.AddLines** **System.Drawing.Pen.LineJoin = System.Drawing.Pen.LineJoin.Round** **System.Drawing.Graphics.DrawPath** **System.Drawing.Graphics.FillPath**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Font and Text Functions

Win32 function	Description	.NET Framework API
AddFontMemResourceEx	Adds an embedded font to the system font table.	**System.Drawing.Text.PrivateFontCollection.AddMemoryF**
AddFontResourceEx	Adds a private or non-enumerable font to the system font table.	**System.Drawing.Text.PrivateFontCollection.AddFontFile**
CreateFont	Creates a logical font.	**System.Drawing.Font** constructor
DrawText	Draws formatted text in a rectangle.	**System.Drawing.Graphics.DrawString**
DrawTextEx	Draws formatted text in a rectangle.	**System.Drawing.Graphics.DrawString**
EnumFontFamExProc	An application-defined callback function used with **EnumFontFamiliesEx** to process fonts.	**System.Drawing.Text.InstalledFontCollection.Families**
EnumFontFamiliesEx	Enumerates all fonts in the system with certain characteristics.	**System.Drawing.Text.InstalledFontCollection.Families**
ExtTextOut	Draws a character string.	**System.Drawing.Graphics.DrawString**
GetCharABCWidths	Gets the widths of consecutive characters from the TrueType font.	Get a **Region[]** for the characters: **System.Drawing.Graphics.MeasureCharacterRanges;** Determine the width of each **Region**: **System.Drawing.Rectangle.Round (Region.GetBounds).Width**
GetCharABCWidthsFloat	Gets the widths of consecutive characters from the current font.	Get a **Region[]** for the characters: **System.Drawing.Graphics.MeasureCharacterRanges;** Determine the width of each **Region**: **System.Drawing.Region.GetBounds.Width**
GetCharWidth32	Gets the widths of consecutive characters from the current font.	Get a **Region[]** for the characters: **System.Drawing.Graphics.MeasureCharacterRanges;** Determine the width of each **Region**: **System.Drawing.Rectangle.Round (Region.GetBounds).Width**
GetCharWidthFloat	Gets the fractional widths of consecutive characters from the current font.	Get a **Region[]** for the characters: **System.Drawing.Graphics.MeasureCharacterRanges;** Determine the width of each **Region**: **System.Drawing.Region.GetBounds.Width**
GetOutlineTextMetrics	Gets text metrics for TrueType fonts.	**System.Drawing.Font** properties
GetTabbedTextExtent	Computes the width and height of a character string, including tabs.	**System.Drawing.Graphics.MeasureString**

GetTextAlign	Gets the text-alignment setting for a device context.	**System.Drawing.StringFormat.Alignment** **System.Drawing.StringFormat.LineAlignment**
GetTextColor	Gets the text color for a device context.	**System.Drawing.SolidBrush.Color**
GetTextExtentPoint32	Computes the width and height of a string of text.	**System.Drawing.Graphics.MeasureString**
GetTextFace	Gets the name of the font that is selected into a device context.	**System.Drawing.Font.Name**
GetTextMetrics	Fills a buffer with the metrics for a font.	**System.Drawing.Font** properties
RemoveFontMemResourceEx	Removes a font whose source was embedded in a document from the system font table.	**System.Drawing.Text.PrivateFontCollection.Dispose**
RemoveFontResourceEx	Removes a private or non-enumerable font from the system font table.	**System.Drawing.Text.PrivateFontCollection.Dispose**
SetTextAlign	Sets the text-alignment flags for a device context.	**System.Drawing.StringFormat.Alignment** **System.Drawing.StringFormat.LineAlignment**
SetTextColor	Sets the text color for a device context.	Either of the following: **System.Drawing.Brushes** **System.Drawing.SolidBrush.SolidBrush(Color)**
TabbedTextOut	Writes a character string at a location, expanding tabs to specified values.	**System.Drawing.Graphics.DrawString** using **System.Drawing.StringFormat.SetTabStops**
TextOut	Writes a character string at a location.	**System.Drawing.Graphics.DrawString**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Icon Functions

Win32 function	**Description**	**.NET Framework API**
CopyIcon	Copies an icon from another module.	**System.Drawing.Icon.FromHandle**
CreateIcon	Creates an icon with a specified size, color, and bit pattern.	**System.Drawing.Icon** constructor
DestroyIcon	Destroys an icon.	**System.Drawing.Icon.Dispose**
DrawIcon	Draws an icon.	**System.Drawing.Graphics.DrawIcon**
DuplicateIcon	Creates a duplicate of a specified icon.	**System.Drawing.Icon.Clone**
LoadIcon	Loads an icon from the executable file associated with an application.	**System.Drawing.Icon** constructor

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Input Method Editor Functions

Win32 function	**Description**	**.NET Framework API**
ImmAssociateContext	Associates the specified input context with the specified window.	Either of the following: **System.Windows.Forms.Control.DefaultImeMode** **System.Windows.Forms.Control.ImeMode**
ImmAssociateContextEx	Changes the association between the input method context and the specified window or its children.	Any of the following: **System.Windows.Forms.Control.DefaultImeMode** **System.Windows.Forms.Control.ImeMode** **System.Windows.Forms.Control.ResetImeMode**

ImmDisableIME	Disables the Input Method Editor.	**System.Windows.Forms.Control.ImeMode = System.Windows.Forms.ImeMode.Disable**
ImmGetContext	Retrieves the input context associated with the specified window.	**System.Windows.Forms.Control.ImeMode**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Keyboard Accelerator Functions

Win32 function	**Description**	**.NET Framework API**
TranslateAccelerator	Processes accelerator keys for menu commands.	**System.Windows.Forms.MenuItem.Shortcut**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Keyboard Input Functions

Win32 function	**Description**	**.NET Framework API**
ActivateKeyboardLayout	Sets the input locale identifier for the calling thread or the current process.	**System.Windows.Forms.InputLanguage.CurrentInputLanguage**
EnableWindow	Enables or disables mouse and keyboard input to the specified window or control.	**System.Windows.Forms.Control.Enabled**
GetActiveWindow	Retrieves the window handle to the active window attached to the calling thread's message queue.	**System.Windows.Forms.Form.ActiveForm**
GetFocus	Retrieves the handle to the window that has the keyboard focus.	**System.Windows.Forms.Control.Controls[].Focused**
GetKeyboardLayout	Retrieves the active input locale identifier for the specified thread.	**System.Windows.Forms.InputLanguage.CurrentInputLanguage**
GetKeyboardLayoutList	Retrieves the input locale identifiers corresponding to the current set of input locales in the system.	**System.Windows.Forms.InputLanguage.InstalledInputLanguages**

GetKeyboardLayoutName	Retrieves the name of the active input locale identifier.	**System.Windows.Forms.InputLanguage.LayoutName**
GetKeyNameText	Retrieves a string that represents the name of a key.	**System.Windows.Forms.KeyEventArgs.KeyCode.ToString**
IsWindowEnabled	Determines whether the specified window is enabled for mouse and keyboard input.	**System.Windows.Forms.Form.Enabled**
SetActiveWindow	Activates a window.	**System.Windows.Forms.Activate**
SetFocus	Sets the keyboard focus to the specified window.	**System.Windows.Forms.Form.Focus**
ToAscii	Translates the specified virtual-key code and keyboard state to the corresponding character or characters.	**System.Windows.Forms.KeyPressEventArgs.KeyChar**
ToUnicode	Translates the specified virtual-key code and keyboard state to the corresponding Unicode character or characters.	**System.Windows.Forms.KeyPressEventArgs.KeyChar**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Line and Curve Functions

Win32 function	Description	.NET Framework API
AngleArc	Draws a line segment and an arc.	**System.Drawing.Graphics.DrawLine** **System.Drawing.Graphics.DrawArc**
Arc	Draws an elliptical arc.	**System.Drawing.Graphics.DrawArc**
ArcTo	Draws an elliptical arc.	**System.Drawing.Graphics.DrawArc**
LineTo	Draws a line from the current position up to, but not including, the specified point.	**System.Drawing.Graphics.DrawLine**
PolyBezier	Draws one or more B&x00E9;zier curves.	**System.Drawing.Graphics.DrawBeziers**
Polyline	Draws a series of line segments by connecting the points in the specified array.	**System.Drawing.Graphics.DrawLines**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

List Box Functions

Win32 function	Description	.NET Framework API
DlgDirList	Replaces the contents of a list box with the names of the subdirectories and files in a specified directory.	**System.Windows.Forms.ListBox.Items.AddRange (System.IO.Directory.GetFileSystemEntries)**
DlgDirSelectEx	Retrieves the current selection from a single-selection list box.	**System.Windows.Forms.ListBox.SelectedItem**
GetListBoxInfo	Retrieves information about the specified list box.	**System.Windows.Forms.ListBox** properties

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Memory Management Functions

Memory management functions:

Win32 function	Description	.NET Framework API
GlobalMemoryStatus	Obtains information about the system's current usage of both physical and virtual memory.	**System.Management.ManagementClass ("Win32_OperatingSystem")** **System.Management.ManagementClass.GetInstances** **System.Management.ManagementObject.Properties**
GlobalMemoryStatusEx	Obtains information about the system's current usage of both physical and virtual memory.	**System.Management.ManagementClass ("Win32_OperatingSystem")** **System.Management.ManagementClass.GetInstances** **System.Management.ManagementObject.Properties**

Global and local functions:

Win32 function	Description	.NET Framework API
GlobalAlloc	Allocates the specified number of bytes from the heap.	**System.Runtime.InteropServices.Marshall.AllocHGlobal**
GlobalFree	Frees the specified global memory object.	**System.Runtime.InteropServices.Marshall.FreeHGlobal**
GlobalReAlloc	Changes the size or attributes of a specified global memory object.	**System.Runtime.InteropServices.Marshall.ReAllocHGlobal**
LocalAlloc	Allocates the specified number of bytes from the heap.	**System.Runtime.InteropServices.Marshall.AllocHGlobal**
LocalFree	Frees the specified local memory object.	**System.Runtime.InteropServices.Marshall.FreeHGlobal**
LocalReAlloc	Changes the size or the attributes of a specified local memory object.	**System.Runtime.InteropServices.Marshall.ReAllocHGlobal**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Menu Functions

Win32 function	Descriptions	.NET Framework API
AppendMenu	Appends an item to various menus and submenus.	**System.Windows.Forms.Menu.MenuItemCollection.Add**
CheckMenuItem	Checks or clears a menu item.	**System.Windows.Forms.MenuItem.Checked**
CheckMenuRadioItem	Checks a specified menu item and makes it a radio item.	**System.Windows.Forms.MenuItem.RadioCheck**

CreateMenu	Creates a menu.	Any of the following: **System.Windows.Forms.ContextMenu** constructor **System.Windows.Forms.MainMenu** constructor **System.Windows.Forms.Menu.MenuItemCollection** constructor
CreatePopupMenu	Creates a drop-down menu, submenu, or shortcut menu.	Either of the following: **System.Windows.Forms.ContextMenu** constructor **System.Windows.Forms.Menu.MenuItemCollection** constructor
DeleteMenu	Deletes a menu item.	**System.Windows.Forms.Menu.MenuItemCollection.Remove**
DestroyMenu	Deletes a menu.	**System.Windows.Forms.Menu.Dispose**
EnableMenuItem	Enables, disables, or grays a menu item.	**System.Windows.Forms.MenuItem.Enabled**
GetMenu	Gets a handle to the menu.	**System.Windows.Forms.Form.Menu.Handle**
GetMenuBarInfo	Gets information about a menu bar.	**System.Windows.Forms.Form.Menu**
GetMenuCheckMarkDimensions	Returns the dimensions of the default check-mark bitmap.	**System.Windows.Forms.SystemInformation.MenuCheckSize**
GetMenuDefaultItem	Gets the default menu item on a menu.	**System.Windows.Forms.MenuItem.DefaultItem**
GetMenuInfo	Gets information about a menu.	**System.Windows.Forms.MenuItem** properties
GetMenuItemCount	Gets the number of items in a menu.	**System.Windows.Forms.Menu.MenuItemCollection.Count**
GetMenuItemID	Gets the identifier of a menu item.	**System.Windows.Forms.MenuItem.MenuID**
GetMenuItemInfo	Gets information about a menu item.	**System.Windows.Forms.MenuItem** properties
GetMenuItemRect	Gets the bounding rectangle for a menu item.	**System.Windows.Forms.MenuItem.DrawItem** **System.Windows.Forms.DrawItemEventArgs.Bounds**
GetMenuState	Gets the menu flags associated with a menu item.	**System.Windows.Forms.MenuItem.Break** **System.Windows.Forms.MenuItem.BarBreak** **System.Windows.Forms.MenuItem.Checked** **System.Windows.Forms.MenuItem.Enabled** **System.Windows.Forms.MenuItem.OwnerDraw**
GetMenuString	Copies the text string of a menu item into a buffer.	**System.Windows.Forms.MenuItem.Text**

GetSubMenu	Gets a handle to the drop-down menu or submenu activated by a menu item.	**System.Windows.Forms.Menu.MenuItems[].Handle**
InsertMenu	Inserts a new menu item into a menu.	**System.Windows.Forms.Menu.MenuItemCollection.Add**
InsertMenuItem	Inserts a new menu item in a menu.	**System.Windows.Forms.Menu.MenuItemCollection.Add**
ModifyMenu	Changes an existing menu item.	**System.Windows.Forms.MenuItem.Break** **System.Windows.Forms.MenuItem.BarBreak** **System.Windows.Forms.MenuItem.Checked** **System.Windows.Forms.MenuItem.Enabled** **System.Windows.Forms.MenuItem.OwnerDraw** **System.Windows.Forms.MenuItem.Text**
RemoveMenu	Deletes a menu item or detaches a submenu from a menu.	**System.Windows.Forms.Menu.MenuItemCollection.Remove**
SetMenu	Assigns a new menu to a window.	**System.Windows.Forms.Form.Menu**
SetMenuDefaultItem	Sets the default menu item for a menu.	**System.Windows.Forms.MenuItem.DefaultItem**
SetMenuInfo	Sets information for a menu.	**System.Windows.Forms.MenuItem** properties
SetMenuItemBitmaps	Associates a bitmap with a menu item.	**System.Windows.Forms.MenuItem.DrawItem**
SetMenuItemInfo	Changes information about a menu item.	**System.Windows.Forms.MenuItem** properties
TrackPopupMenu	Displays a shortcut menu at a location and tracks item selection on a menu.	**System.Windows.Forms.ContextMenu.Show**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Message and Message Queue Functions

Win32 function	Description	.NET Framework API
GetMessagePos	Retrieves the position of the cursor for the last message retrieved by **GetMessage**.	Inside an event handler: **System.Windows.Forms.Windows.Forms.Control.MousePosition**
PostQuitMessage	Indicates to the system that a thread has made a request to terminate (quit).	**System.Windows.Forms.Application.ExitThread**
SendMessage	Sends the specified	Any of the following:

	message to a window or windows.	**System.Windows.Forms.Control.DefWndProc** **System.Windows.Forms.Control.WndProc** **System.Windows.Forms.Form.DefWndProc** **System.Windows.Forms.Form.WndProc**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Metafile Functions

Win32 function	Description	.NET Framework API
CopyEnhMetaFile	Copies the contents of an enhanced-format metafile to a specified file.	**System.Drawing.Imaging.Metafile.Save**
CreateEnhMetaFile	Creates a device context for an enhanced-format metafile.	**System.Drawing.Imaging.Metafile** constructor
DeleteEnhMetaFile	Deletes an enhanced-format metafile or an enhanced-format metafile handle.	**System.Drawing.Imaging.Metafile.Dispose**
EnhMetaFileProc	An application-defined callback function used with **EnumEnhMetaFile**.	**System.Drawing.Graphics.EnumerateMetafileProc**
EnumEnhMetaFile	Enumerates the records within an enhanced-format metafile.	**System.Drawing.Graphics.EnumerateMetafile**
GdiComment	Copies a comment from a buffer into a specified enhanced-format metafile.	**System.Drawing.Graphics.AddMetafileComment**
GetEnhMetaFile	Creates a handle that identifies the enhanced-format metafile stored in the specified file.	**System.Drawing.Imaging.Metafile** constructor
GetEnhMetaFileBits	Retrieves the contents of the specified enhanced-format metafile and copies it into a buffer.	**System.Drawing.Imaging.Metafile.Save**
GetEnhMetaFileHeader	Retrieves the record containing the header for the specified enhanced-format metafile.	**System.Drawing.Imaging.Metafile.GetMetafileHeader**
GetEnhMetaFilePaletteEntries	Retrieves optional palette entries from the specified enhanced metafile.	**System.Drawing.Imaging.Metafile.Palette.Entries**
GetWinMetaFileBits	Converts the enhanced-format records from a metafile into Windows-format records.	**System.Drawing.Imaging.Metafile.Save**
PlayEnhMetaFile	Displays the picture stored in the specified enhanced-format metafile.	**System.Drawing.Graphics.DrawImage**

PlayEnhMetaFileRecord	Plays an enhanced-metafile record by executing the graphics device interface functions identified by the record.	**System.Drawing.Imaging.Metafile.PlayRecord** **System.Drawing.Imaging.PlayRecordCallback**
SetEnhMetaFileBits	Creates a memory-based enhanced-format metafile from the specified data.	**System.Drawing.Imaging.Metafile.Save**
SetWinMetaFileBits	Converts a metafile from the older Windows format to the new enhanced format.	**System.Drawing.Imaging.Metafile.Save**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Mouse Input Functions

Win32 function	Description	.NET Framework API
GetDoubleClickTime	Gets the double-click time for the mouse.	**System.Windows.Forms.SystemInformation.DoubleClickTime**
mouse_event	Synthesizes mouse motion and button clicks.	**System.Windows.Forms.MouseEventArgs**
ReleaseCapture	Releases the mouse capture and restores mouse input processing.	**System.Windows.Forms.Control.Capture**
SetCapture	Sets the mouse capture to a window.	**System.Windows.Forms.Control.Capture**
TrackMouseEvent –or– **_TrackMouseEvent**	Posts messages when a mouse leaves a window or hovers over a window.	**System.Windows.Forms.Control.OnMouseLeave** **System.Windows.Forms.Control.OnMouseHover**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Multiple Display Monitor Functions

Win32 function	Description	.NET Framework API
GetMonitorInfo	Retrieves information about a display monitor.	**System.Windows.Forms.Screen.Bounds** **System.Windows.Forms.Screen.Primary** **System.Windows.Forms.Screen.WorkingArea**
MonitorFromPoint	Retrieves a handle to the display monitor that contains a specified point.	**System.Windows.Forms.Screen.FromPoint**
MonitorFromRect	Retrieves a handle to the display monitor that has the largest area of intersection with a specified rectangle.	**System.Windows.Forms.Screen.FromRectangle**
MonitorFromWindow	Retrieves a handle to the display monitor that has the largest area of intersection with the bounding rectangle of a specified window.	**System.Windows.Forms.Screen.FromHandle**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Multiple-Document Interface Functions

Win32 function	Description	.NET Framework API
CreateMDIWindow	Creates a multiple-document interface (MDI) child window.	**System.Windows.Forms.Form** constructor **System.Windows.Forms.Form.AutoScroll**

		System.Windows.Forms.Form.DesktopBounds **System.Windows.Forms.Form.MdiParent** **System.Windows.Forms.Form.WindowState**
TranslateMDISysAccel	Processes accelerator keystrokes for window menu commands of the MDI child windows associated with the specified MDI client window.	Keyboard events for MDI children are automatically translated.

Alphabetical Win32 Category List
Hierarchical Win32 Category List

National Language Functions

Win32 function	Description	.NET Framework API
ConvertDefaultLocale	Converts a default locale value to an actual locale identifier.	**System.Globalization.CultureInfo.CurrentCulture** **System.Globalization.CultureInfo.InstalledUICultur**
EnumCalendarInfo	Enumerates calendar information for a specified locale.	Access the **System.Globalization.CultureInfo.DateTin** **System.Globalization.DateTimeFormatInfo** object, ar properties: **System.Globalization.DateTimeFormatInfo.Abbrevia** **System.Globalization.DateTimeFormatInfo.Abbrevia** **System.Globalization.DateTimeFormatInfo.DayNam** **System.Globalization.DateTimeFormatInfo.LongDat** **System.Globalization.DateTimeFormatInfo.MonthNa** **System.Globalization.DateTimeFormatInfo.ShortDa**
EnumCalendarInfoEx	Enumerates calendar information for a specified locale.	Iterate through the calendars: **System.Globalization.CultureInfo.DateTimeFormat.** **System.Globalization.CultureInfo.OptionalCalendar**
EnumCalendarInfoProc	An application-defined function used with **EnumCalendarInfo**.	Same as **EnumCalendarInfo**
EnumCalendarInfoProcEx	An application-defined function used with **EnumCalendarInfoEx**.	Same as **EnumCalendarInfoEx**
EnumCodePagesProc	An application-defined function used with **EnumSystemCodePages**.	Same as **EnumSystemCodePages**
EnumDateFormats	Enumerates the long or short date formats that are available for a specified locale.	**System.DateTime.GetDateTimeFormats(System.Glo**
EnumDateFormatsEx	Enumerates the long or short date formats that are available for a specified locale, including date formats for any alternate calendars.	Iterate through the calendars: **System.Globalization.CultureInfo.DateTimeFormat.** **System.Globalization.CultureInfo.OptionalCalendar** **System.DateTime.GetDateTimeFormats(System.Glo**
EnumDateFormatsProc	An application-defined function used with **EnumDateFormats**.	Same as **EnumDateFormats**
EnumDateFormatsProcEx	An application-defined function used with **EnumDateFormatsEx**.	Same as **EnumDateFormatsEx**
EnumGeoInfoProc	A callback function used with **EnumSystemGeoID**.	Same as **EnumSystemGeoID**
EnumLocalesProc	An application-defined function used with **EnumSystemLocales**.	Same as **EnumSystemLocales**
EnumSystemCodePages	Enumerates the code pages that are either installed on or supported by a system.	**System.Globalization.CultureInfo.GetCultures (System.Globalization.CultureTypes.AllCultures)** ret **CultureInfo** The various code page properties of **System.Globalizati**

EnumSystemGeoID	Enumerates all GeoIDs on the system.	Retrieve an array of **CultureInfo** by calling **System.Globalization.CultureInfo.GetCultures (System.Globalization.CultureTypes.AllCultures)**, ar **CultureInfo** properties: **System.Globalization.CultureInfo.ThreeLetterISOLa** **System.Globalization.CultureInfo.ThreeLetterWindc** **System.Globalization.CultureInfo.TwoLetterISOLan**
EnumSystemLocales	Enumerates the locales that are either installed on or supported by a system.	**System.Globalization.CultureInfo.GetCultures (System.Globalization.CultureTypes.SpecificCulture** **System.Globalization.CultureInfo.GetCultures (System.Globalization.CultureTypes.InstalledWin32**
EnumTimeFormats	Enumerates the time formats that are available for a specified locale.	**System.DateTime.GetDateTimeFormats**
EnumTimeFormatsProc	An application-defined function used with **EnumTimeFormats**.	Same as **EnumTimeFormats**
GetACP	Retrieves the current ANSI code-page identifier for the system.	**System.Globalization.CultureInfo.InstalledUICultur**
GetCalendarInfo	Retrieves information about a calendar.	**System.Globalization.CultureInfo.DateTimeFormat**
GetCurrencyFormat	Formats a number string as a currency string for a specified locale.	Create a **System.Globalization.CultureInfo** object usir either the culture name or identifier (or use the current cu available on **System.Threading.Thread.CurrentCultur** call **System.Convert.ToDouble(str)** where **str** is the nu format; and then call **System.Double.ToString("C", ci)** (or **Sys ("C")** to use the current culture), where **C** is the standar and **ci** is the **CultureInfo** object you created **Note** For more information on the variety of formatting the .NET Framework, see **System.String.Format**.
GetDateFormat	Formats a date as a date string for a specified locale.	**System.DateTime(…, CultureInfo(LCID).Calendar).T**
GetLocaleInfo	Retrieves information about a locale.	**System.Globalization.CultureInfo.DateTimeFormat** **System.Globalization.CultureInfo.NumberFormat** **System.Globalization.CultureInfo.TextInfo**
GetNumberFormat	Formats a number string as a number string customized for a specified locale.	Create a **System.Globalization.CultureInfo** object usir either the culture name or identifier (or use the current cu available on **System.Threading.Thread.CurrentCultur** call **System.Convert.ToDouble(str).** where **str** is the n format; and then call **System.Double.ToString("N", ci)** (or **Sys ("N")** to use the current culture), where **N** is the standar and **ci** is the **CultureInfo** object you created **Note** For more information on the variety of formatting the .NET Framework, see **System.String.Format**.
GetOEMCP	Retrieves the current original equipment manufacturer (OEM) code-page identifier for the system.	**System.Globalization.CultureInfo.InstalledUICultur**
GetSystemDefaultLangID	Retrieves the language identifier of the system locale.	**System.Globalization.CultureInfo.InstalledUICultur**
GetSystemDefaultLCID	Retrieves the system default locale identifier.	**System.Globalization.CultureInfo.InstalledUICultur**
GetSystemDefaultUILanguage	Retrieves the language identifier for the default language of the system.	**System.Globalization.CultureInfo.InstalledUICultur**

GetThreadLocale	Retrieves the current locale for the calling thread.	Either of the following: **System.Globalization.CultureInfo.CurrentCulture** **System.Threading.Thread.CurrentCulture**
GetTimeFormat	Formats time as a time string for a specified locale.	**System.DateTime(…, CultureInfo(LCID).Calendar).T**
GetUserDefaultLangID	Retrieves the language identifier of the current user locale.	**System.Globalization.CultureInfo.CurrentCulture**
GetUserDefaultLCID	Retrieves the user default–locale identifier.	**System.Globalization.CultureInfo.CurrentCulture**
GetUserDefaultUILanguage	Retrieves the language identifier for the current user language.	**System.Globalization.CultureInfo.CurrentUICulture**
GetUserGeoID	Gets information about the user's location.	**System.Globalization.RegionInfo**
IsValidCodePage	Determines whether a specified code page is valid.	Use an array of **CultureInfo** objects returned by: **System.Globalization.CultureInfo.GetCultures (System.Globalization.CultureTypes.AllCultures)** Compare to the code page properties of **System.Globalization.CultureInfo.TextInfo**
IsValidLocale	Determines whether a specified locale identifier is valid.	**try{ System.Globalization.CultureInfo(LCID) }**
LCMapString	Maps one character string to another, performing a specified locale-dependent transformation.	**System.Globalization.SortKey**
SetCalendarInfo	Sets an item of locale information for a calendar.	**System.Globalization.CultureInfo.DateTimeFormat**
SetLocaleInfo	Sets an item of locale information.	**System.Globalization.CultureInfo.DateTimeFormat** **System.Globalization.CultureInfo.NumberFormat**
SetThreadLocale	Sets the current locale for the calling thread.	**System.Threading.Thread.CurrentCulture** **Note** Sets **langid** only.

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Painting and Drawing Functions

Win32 function	Description	.NET Framework API
BeginPaint	Prepares a window for painting.	**System.Windows.Forms.Form.OnPaint**
DrawFrameControl	Draws a frame control.	**System.Window.Forms.ControlPaint**
GetBkColor	Returns the background color for a device context.	**System.Windows.Forms.Form.BackColor**
GetBkMode	Returns the background mix mode for a device context.	**System.Windows.Forms.Form.GetStyle (ControlStyles.Opaque)**
GetUpdateRect	Gets the coordinates of the smallest rectangle that encloses the update region of a window.	**System.Windows.Forms.PaintEventArgs.ClipRectangle**
GetWindowDC	Gets the device context for a window, including title bar, menus, and scroll bars.	Any of the following: **System.Drawing.Graphics.FromHwnd (Windows.Forms.Control.Handle)** **System.Windows.Forms.Control.CreateGraphics.GetHdc** **System.Windows.Forms.PaintEventArgs.Graphics.GetHdc**
GetWindowRgn	Gets a copy of the window region of a window.	**System.Windows.Forms.Control.Region**
GrayString	Draws gray text at a location.	**System.Drawing.Graphics.DrawString(…, Drawing.Brushes.Gray, …)**

InvalidateRect	Adds a rectangle to the update region of a window.	**System.Windows.Forms.Control.Invalidate**
InvalidateRgn	Invalidates the client area within a region.	**System.Windows.Forms.Control.Invalidate (System.Windows.Forms.Form.ClientRectangle)**
RedrawWindow	Updates a region in the client area of a window.	**System.Windows.Forms.Control.Update**
SetBkColor	Sets the background to a color value.	**System.Windows.Forms.Form.BackColor**
SetBkMode	Sets the background mix mode of a device context.	**System.Windows.Forms.Form.SetStyle (System.Windows.Forms.ControlStyles.Opaque, System.Boolean)**
SetWindowRgn	Sets the window region of a window.	**System.Windows.Forms.Control.Region**
UpdateWindow	Updates the client area of a window.	**System.Windows.Forms.Control.Update**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Path Functions

Win32 function	Description	.NET Framework API
CloseFigure	Closes an open figure in a path.	**System.Drawing.Drawing2D.GraphicsPath.CloseFigure**
FillPath	Closes any open figures in the current path and fills the interior of the path by using the current brush and polygon-filling mode.	**System.Drawing.Graphics.FillPath** **System.Drawing.Drawing2D.GraphicsPath.CloseFigures**
FlattenPath	Transforms any curves in the path that is selected into the current device context, turning each curve into a sequence of lines.	**System.Drawing.Drawing2D.GraphicsPath.Flatten**
GetMiterLimit	Retrieves the miter limit for the specified device context.	**System.Drawing.Pen.MiterLimit**
GetPath	Retrieves the coordinates defining the endpoints of lines and the control points of curves found in the path that is selected into the specified device context.	**System.Drawing.Drawing2D.GraphicsPath.PathData**
PathToRegion	Creates a region from the path that is selected into the specified device context.	**System.Drawing.Region.Region (System.Drawing.Drawing2D.GraphicsPath)**
SetMiterLimit	Sets the limit for the length of miter joins for the specified device context.	**System.Drawing.Pen.MiterLimit**
StrokeAndFillPath	Closes any open figures in a path, strokes the outline of the path by using the current pen, and fills its interior by using the current brush.	**System.Drawing.Graphics.DrawPath** **System.Drawing.Graphics.FillPath** **System.Drawing.Drawing2D.GraphicsPath.CloseFigures**
StrokePath	Renders the specified path by using the current pen.	**System.Drawing.Graphics.DrawPath**
WidenPath	Redefines the current path as the area that would be painted if the path were stroked using the pen currently selected into the given device context.	**System.Drawing.Drawing2D.GraphicsPath.Widen**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Pen Functions

Win32 function	Description	.NET Framework API
CreatePen	Creates a logical pen that has the specified style, width, and color.	**System.Drawing.Pen** constructor
ExtCreatePen	Creates a logical cosmetic or geometric pen that has the specified style, width, and brush attributes.	**System.Drawing.Pen** constructor

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Performance Monitoring Functions

Win32 function	Description	.NET Framework API
PdhAddCounter	Initializes a counter structure for the specified counter in the specified query.	**System.Diagnostics.PerformanceCounter** constructor
PdhCalculateCounterFromRawValue	Computes the current value of a counter using the raw counter data passed in the parameter list.	**System.Diagnostics.CounterSample.Calculate**
PdhCloseLog	Closes the specified log file.	**System.Diagnostics.EventLog.Close**
PdhCloseQuery	Closes all counters contained in the specified query, closes all handles related to the query, and frees all memory associated with the query.	**System.Diagnostics.PerformanceCounterCategory.Delete**
PdhCollectQueryData	Collects the current raw data value for all counters in the specified query and updates the status code of each counter.	**System.Diagnostics.PerformanceCounterCategory.ReadCat**
PdhEnumObjectItems	Returns the available counters and instances provided by the specified	**System.Diagnostics.PerformanceCounterCategory.GetCoun** **System.Diagnostics.PerformanceCounterCategory.GetInsta**

	object on the specified computer.	
PdhEnumObjects	Returns a list of objects available on the specified computer.	**System.Diagnostics.PerformanceCounterCategory.GetCateg**
PdhFormatFromRawValue	Converts one raw performance data value to the specified format or two raw performance data values if the format conversion is time-based.	**System.Diagnostics.CounterSample.Calculate**
PdhGetCounterInfo	Retrieves information about a counter, such as data size, counter type, path, and user-supplied data values.	**System.Diagnostics.PerformanceCounter** properties
PdhGetCounterTimeBase	Returns the time base of the specified counter.	**System.Diagnostics.CounterSample.CounterFrequency**
PdhGetFormattedCounterValue	Returns the current value of a specified counter in the format requested by the caller.	**System.Diagnostics.PerformanceCounter.NextValue**
PdhGetRawCounterArray	Returns an array of raw values from the specified counter.	**System.Diagnostics.PerformanceCounterCategory.ReadCat**
PdhGetRawCounterValue	Returns the current raw value of the counter.	**System.Diagnostics.PerformanceCounter.NextSample** **Note** **System.Diagnostics.InstanceData** holds the raw data.
PdhMakeCounterPath	Creates a full counter path using the members specified in the structure	All of the following: **System.Diagnostics.PerformanceCounter.MachineName** **System.Diagnostics.PerformanceCounter.CategoryName** **System.Diagnostics.PerformanceCounter.InstanceName** **System.Diagnostics.PerformanceCounter.CounterName**

	passed in the parameter list.	
PdhOpenLog	Opens the specified log file.	**System.Diagnostics.EventLog** constructor **Note** Must explicitly handle writing to the log file.
PdhOpenQuery	Creates and initializes a unique query structure that is used to manage the collection of performance data.	**System.Diagnostics.PerformanceCounterCategory** construct
PdhValidatePath	Validates that the specified counter is present on the computer that is specified in the counter path.	Either of the following: **System.Diagnostics.PerformanceCounterCategory.Exists** **System.Diagnostics.PerformanceCounterCategory.Instance**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Printing Functions

Functions used to print:

Win32 function	Description	.NET Framework API
AbortDoc	Stops the current print job and erases everything drawn since the last call to **StartDoc**.	**System.Drawing.Printing.PrintPageEventArgs.Cancel**
DeviceCapabilities	Retrieves the capabilities of a printer device driver.	**System.Drawing.Printing.PrinterSettings.PrinterName** **System.Drawing.Printing.PrinterSettings** properties
EndDoc	Ends a print job.	Either of the following: **System.Drawing.Printing.PrintDocument.OnEndPrint** **System.Drawing.Printing.PrintController.OnEndPrint**
EndPage	Notifies the device that the application has finished writing to a page.	**System.Drawing.Printing.PrintController.OnEndPage**
StartDoc	Starts a print job.	**System.Drawing.Printing.PrintDocument.Print** **System.Drawing.Printing.PrintDocument.OnBeginPrint** **System.Drawing.Printing.PrintController.OnStartPrint**
StartPage	Prepares the printer driver to accept data.	**System.Drawing.Printing.PrintDocument.OnQueryPageSettings** **System.Drawing.Printing.PrintController.OnStartPage** **System.Drawing.Printing.PrintDocument.OnPrintPage**

Functions used to access the print spooler:

Win32 function	Description	.NET Framework API
DocumentProperties	Retrieves or modifies	**System.Windows.Forms.PageSetupDialog**

	printer initialization information or displays a printer-configuration property sheet for the specified printer.	**System.Windows.Forms.PrintDialog**
EndPagePrinter	Notifies the print spooler that the application is at the end of a page in a print job.	**System.Drawing.Printing.PrintPageEventArgs.HasMorePages**
EnumForms	Enumerates the forms supported by the specified printer.	**System.Drawing.Printing.PrinterSettings.PaperSizes**
EnumPrinters	Enumerates available printers, print servers, domains, or print providers.	**System.Drawing.Printing.PrinterSettings.InstalledPrinters**
GetDefaultPrinter	Retrieves the printer name of the default printer for the current user on the local computer.	**System.Drawing.Printing.PrinterSettings** constructor **System.Drawing.Printing.PrinterSettings.PrinterName**
GetForm	Retrieves information about a specified form.	**System.Drawing.Printing.PrinterSettings.PaperSizes[]**
GetPrinter	Retrieves information about a specified printer.	**System.Drawing.Printing.PrinterSettings.PrinterName** **System.Drawing.Printing.PrinterSettings** properties
PrinterProperties	Displays a printer-properties property sheet for the specified printer.	**System.Windows.Forms.PageSetupDialog** **System.Windows.Forms.PrintDialog**
SetPrinter	Sets the data for a specified printer or sets the state of the specified printer by pausing printing, resuming printing, or clearing all print jobs.	**System.Drawing.Printing.PrinterSettings.PrinterName** **System.Drawing.Printing.PrinterSettings** properties

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Process and Thread Functions

Functions used with processes:

Win32 function	**Description**	**.NET Framework API**
CreateProcess	Creates a new process and its primary thread.	**System.Diagnostics.Process** constructor **System.Diagnostics.Process.Start**
ExitProcess	Ends a process and all its threads.	**System.Diagnostics.Process.CloseMainWindow**
GetCommandLine	Retrieves the command-line string for the current process.	Either of the following: **System.Environment.CommandLine** **System.Diagnostics.ProcessStartInfo.Arguments**
GetCurrentProcess	Retrieves a pseudo handle for the current process.	**System.Diagnostics.Process.Handle**
GetCurrentProcessId	Retrieves the process identifier of the calling	**System.Diagnostics.Process.Id**

	process.	
GetEnvironmentStrings	Retrieves the environment block for the current process.	**System.Diagnostics.ProcessStartInfo.EnvironmentVariables**
GetEnvironmentVariable	Retrieves the value of the specified variable from the environment block of the calling process.	**System.Diagnostics.ProcessStartInfo.EnvironmentVariables[]**
GetExitCodeProcess	Retrieves the termination status of the specified process.	**System.Diagnostics.Process.ExitCode**
GetPriorityClass	Retrieves the priority class for the specified process.	**System.Diagnostics.Process.PriorityClass**
GetProcessAffinityMask	Retrieves a process affinity mask for the specified process and the system affinity mask for the system.	**System.Diagnostics.Process.ProcessorAffinity**
GetProcessPriorityBoost	Retrieves the priority boost control state of the specified process.	**System.Diagnostics.Process.PriorityBoostEnabled**
GetProcessTimes	Retrieves timing information for the specified process.	**System.Diagnostics.Process.ExitTime** **System.Diagnostics.Process.PrivilegedProcessorTime** **System.Diagnostics.Process.StartTime** **System.Diagnostics.Process.TotalProcessorTime** **System.Diagnostics.Process.UserProcessorTime**
GetProcessWorkingSetSize	Retrieves the minimum and maximum working set sizes of the specified process.	**System.Diagnostics.Process.MinWorkingSet** **System.Diagnostics.Process.MaxWorkingSet**
GetStartupInfo	Retrieves the contents of the **STARTUPINFO** structure that was specified when the calling process was created.	**System.Diagnostics.Process.StandardError** **System.Diagnostics.Process.StandardInput** **System.Diagnostics.Process.StandardOutput** **System.Diagnostics.Process.StartInfo.WindowStyle**
OpenProcess	Opens an existing process object.	**System.Diagnostics.Process.GetProcessById**
SetEnvironmentVariable	Sets the value of an environment variable for the current process.	**System.Diagnostics.ProcessStartInfo.EnvironmentVariables[]**
SetPriorityClass	Sets the priority class for the specified process.	**System.Diagnostics.Process.PriorityClass**
SetProcessAffinityMask	Sets a processor affinity mask for the threads of a	**System.Diagnostics.Process.ProcessorAffinity**

	specified process.	
SetProcessPriorityBoost	Disables the ability of the system to temporarily boost the priority of the threads of the specified process.	**System.Diagnostics.Process.PriorityBoostEnabled**
SetProcessWorkingSetSize	Sets the minimum and maximum working set sizes for the specified process.	**System.Diagnostics.Process.MaxWorkingSet** **System.Diagnostics.Process.MinWorkingSet**
TerminateProcess	Terminates the specified process and all of its threads.	**System.Diagnostics.Process.Kill**

Functions used with threads:

Win32 function	**Description**	**.NET Framework API**
CreateThread	Creates a thread to execute within the virtual address space of the calling process.	**System.Threading.Thread** constructor **System.Threading.Thread.Start**
GetCurrentThread	Retrieves a pseudo handle for the current thread.	**System.Threading.Thread.CurrentThread**
GetCurrentThreadId	Retrieves the thread identifier of the calling thread.	Either of the following: **System.AppDomain.GetCurrentThreadId** **System.Diagnostics.ProcessThread.Id**
GetThreadPriority	Retrieves the priority value for the specified thread.	Either of the following: **System.Diagnostics.ProcessThread.PriorityLevel** **System.Threading.Thread.Priority**
GetThreadPriorityBoost	Retrieves the priority boost control state of the specified thread.	**System.Diagnostics.ProcessThread.PriorityBoostEnabled**
GetThreadTimes	Retrieves timing information for the specified thread.	**System.Diagnostics.ProcessThread.PrivilegedProcessorTime** **System.Diagnostics.ProcessThread.StartTime** **System.Diagnostics.ProcessThread.TotalProcessorTime** **System.Diagnostics.ProcessThread.UserProcessorTime**
ResumeThread	Decrements the suspend count for a thread.	**System.Threading.Thread.Resume**
SetThreadAffinityMask	Sets a processor affinity mask for the specified thread.	**System.Diagnostics.ProcessThread.ProcessorAffinity**
SetThreadIdealProcessor	Specifies a preferred processor for a thread.	**System.Diagnostics.ProcessThread.IdealProcessor**
SetThreadPriority	Sets the priority value for the specified thread.	Either of the following: **System.Diagnostics.ProcessThread.PriorityLevel** **System.Threading.Thread.Priority**
SetThreadPriorityBoost	Enables or disables the ability of the system to temporarily boost the priority of a thread.	**System.Diagnostics.ProcessThread.PriorityBoostEnabled**

Sleep	Suspends the execution of the current thread for a specified interval.	**System.Threading.Thread.Sleep**
SuspendThread	Suspends the specified thread.	**System.Threading.Thread.Suspend**
SwitchToThread	Causes the calling thread to yield execution to another thread that is ready to run on the current processor.	**System.Threading.Thread.Sleep(0)**
TerminateThread	Terminates a thread.	**System.Threading.Thread.Abort**
ThreadProc	An application-defined function that serves as the starting address for a thread.	See **System.Threading.ThreadStart** delegate
TlsAlloc	Allocates a thread local storage (TLS) index.	**System.Threading.Thread.AllocateNamedDataSlot**
TlsFree	Releases a TLS index.	**System.Threading.Thread.FreeNamedDataSlot**
TlsGetValue	Retrieves the value in the calling thread's TLS slot for a specified TLS index.	**System.Threading.Thread.GetData**
TlsSetValue	Stores a value in the calling thread's TLS slot for a specified TLS index.	**System.Threading.Thread.SetData**
WaitForInputIdle	Waits until the specified process is waiting for user input with no input pending or until the time-out interval has elapsed.	**System.Diagnostics.Process.WaitForInputIdle**

Functions used in thread pooling:

Win32 function	**Description**	**.NET Framework API**
BindIoCompletionCallback	Binds the specified file handle to the I/O completion port for the thread pool.	**System.Threading.ThreadPool.BindHandle**
QueueUserWorkItem	Queues a work item to a worker thread in the thread pool.	**System.Threading.ThreadPool.QueueUserWorkItem** **System.Threading.ThreadPool.UnsafeQueueUserWorkItem**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

PSAPI Functions

Win32 function	**Description**	**.NET Framework API**
EnumProcesses	Retrieves the process identifier for each process object in the system.	**System.Diagnostics.Process.GetProcesses**
EnumProcessModules	Retrieves a handle for each	**System.Diagnostics.Process.Modules**

	module in the specified process.	
GetModuleBaseName	Retrieves the base name for the specified module.	**System.Diagnostics.ProcessModule.ModuleName**
GetModuleFileNameEx	Retrieves the fully qualified path for the specified module.	Either of the following: **System.Diagnostics.ProcessModule.FileName** **System.Reflection.Module.FullyQualifiedName**
GetModuleInformation	Retrieves information about the specified module.	**System.Diagnostics.ProcessModule**
GetProcessMemoryInfo	Retrieves information about the memory usage of the specified process.	**System.Diagnostics.Process.PagedMemorySize** **System.Diagnostics.Process.PeakPagedMemorySize** **System.Diagnostics.Process.PeakWorkingSet** **System.Diagnostics.Process.WorkingSet**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Rectangle Functions

Win32 function	Description	.NET Framework API
EqualRect	Determines whether the two specified rectangles are equal by comparing the coordinates of their upper-left and lower-right corners.	**System.Drawing.Rectangle.Equals**
InflateRect	Increases or decreases the width and height of the specified rectangle.	**System.Drawing.Rectangle.Inflate**
IntersectRect	Calculates the intersection of two source rectangles and places the coordinates of the intersection rectangle into the destination rectangle.	**System.Drawing.Rectangle.Intersect**
IsRectEmpty	Determines whether the specified rectangle is empty.	**System.Drawing.Rectangle.IsEmpty**
OffsetRect	Moves the specified rectangle by the specified offsets.	**System.Drawing.Rectangle.Offset**
PtInRect	Determines whether the specified point lies within the specified rectangle.	**System.Drawing.Rectangle.Contains**
SetRect	Sets the coordinates of the specified rectangle.	**System.Drawing.Rectangle.FromLTRB**
SetRectEmpty	Creates an empty rectangle in which all coordinates are set to zero.	**System.Drawing.Rectangle.FromLTRB**
UnionRect	Creates the union of two rectangles.	**System.Drawing.Rectangle.Union**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Region Functions

Win32 function	Description	.NET Framework API
CombineRgn	Combines two regions and stores the result in a third region.	**System.Drawing.Region.Clone** **System.Drawing.Region.Exclude** **System.Drawing.Region.Intersect** **System.Drawing.Region.Union** **System.Drawing.Region.Xor**
CreateEllipticRgn	Creates an elliptical region.	**System.Drawing.Drawing2D.GraphicsPath** constructor **System.Drawing.Drawing2D.GraphicsPath.AddEllipse** **System.Drawing.Region.Region (System.Drawing.Drawing2D.GraphicsPath)**
CreateEllipticRgnIndirect	Creates an elliptical region.	**System.Drawing.Drawing2D.GraphicsPath** constructor **System.Drawing.Drawing2D.GraphicsPath.AddEllipse** **System.Drawing.Region.Region (System.Drawing.Drawing2D.GraphicsPath)**
CreatePolygonRgn	Creates a polygonal region.	**System.Drawing.Drawing2D.GraphicsPath** constructor **System.Drawing.Drawing2D.GraphicsPath.AddPolygon**

		System.Drawing.Region.Region (System.Drawing.Drawing2D.GraphicsPath)
CreatePolyPolygonRgn	Creates a region consisting of a series of polygons.	**System.Drawing.Drawing2D.GraphicsPath** constructor **System.Drawing.Drawing2D.GraphicsPath.AddPolygon** **System.Drawing.Region.Region (System.Drawing.Drawing2D.GraphicsPath)**
CreateRectRgnIndirect	Creates a rectangular region.	**System.Drawing.Region.Region (System.Drawing.Rectangle)**
CreateRoundRectRgn	Creates a rectangular region with rounded corners.	**System.Drawing.Drawing2D.GraphicsPath.AddLines** **System.Drawing.Pen.LineJoin = System.Drawing.Drawing2D.LineJoin.Round**
EqualRgn	Checks the two specified regions to determine whether they are identical.	**System.Drawing.Region.Equals**
ExtCreateRegion	Creates a region from the specified region and transformation data.	**System.Drawing.Region.Transform**
FillRgn	Fills a region by using the specified brush.	**System.Drawing.Graphics.FillRegion**
GetPolyFillMode	Retrieves the current polygon fill mode.	**System.Drawing.Drawing2D.GraphicsPath.FillMode**
GetRegionData	Fills the specified buffer with data describing a region.	**System.Drawing.Region.GetRegionData**
GetRgnBox	Retrieves the bounding rectangle of the specified region.	**System.Drawing.Region.GetBounds**
OffsetRgn	Moves a region by the specified offsets.	**System.Drawing.Region.Translate**
PtInRegion	Determines whether the specified point is inside the specified region.	**System.Drawing.Region.IsVisible**
RectInRegion	Determines whether any part of the specified rectangle is within the boundaries of a region.	**System.Drawing.Region.IsVisible**
SetPolyFillMode	Sets the polygon fill mode for functions that fill polygons.	**System.Drawing.Drawing2D.GraphicsPath.FillMode**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Registry Functions

Win32 function	**Description**	**.NET Framework API**
RegCloseKey	Releases a handle to the specified registry key.	**Microsoft.Win32.RegistryKey.Close**
RegConnectRegistry	Establishes a connection to a predefined registry handle on another computer.	**Microsoft.Win32.RegistryKey.OpenRemoteBaseKey**
RegCreateKeyEx	Creates the specified registry key.	**Microsoft.Win32.RegistryKey.CreateSubKey**
RegDeleteKey	Deletes a subkey.	**Microsoft.Win32.RegistryKey.DeleteSubKeyTree**
RegDeleteValue	Removes a named value from the specified registry key.	**Microsoft.Win32.RegistryKey.DeleteValue**
RegEnumKeyEx	Enumerates subkeys of the specified open registry key.	**Microsoft.Win32.RegistryKey.GetSubKeyNames**
RegEnumValue	Enumerates the values for the specified open registry	**Microsoft.Win32.RegistryKey.GetValueNames** **Microsoft.Win32.RegistryKey.GetValue**

	key.	
RegFlushKey	Writes all the attributes of the specified open registry key into the registry.	**Microsoft.Win32.RegistryKey.Flush**
RegOpenCurrentUser	Retrieves a handle to the **HKEY_CURRENT_USER** key for the user that the current thread is impersonating.	**Microsoft.Win32.Registry.CurrentUser**
RegOpenKeyEx	Opens the specified registry key.	**Microsoft.Win32.RegistryKey.OpenSubKey**
RegOpenUserClassesRoot	Retrieves a handle to the **HKEY_CLASSES_ROOT** key for the specified user.	**Microsoft.Win32.Registry.ClassesRoot**
RegQueryInfoKey	Retrieves information about the specified registry key.	**Microsoft.Win32.RegistryKey.SubKeyCount** **Microsoft.Win32.RegistryKey.ValueCount**
RegQueryValueEx	Retrieves the type and data for a specified value name associated with an open registry key.	**Microsoft.Win32.RegistryKey.GetValue**
RegSetValueEx	Sets the data and type of a specified value under a registry key.	**Microsoft.Win32.RegistryKey.SetValue**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Resource Functions

Win32 function	Description	.NET Framework API
CopyImage	Creates a new icon, cursor, or bitmap and copies the attributes of the old one to the new one.	**System.Drawing.Image.Clone**
EnumResNameProc	An application-defined callback function used with **EnumResourceNames**.	**System.Resources.ResourceReader.GetEnumerator**
EnumResourceNames	Searches for resources of a specified type and passes the name or the ID to a callback function.	**System.Resources.ResourceReader.GetEnumerator**
LoadImage	Loads an icon, cursor, or bitmap.	**System.Drawing.Bitmap.Bitmap(System.Type, System.String)** **System.Drawing.Icon.Icon(System.Type, System.String)** **System.Windows.Forms.Cursor.Cursor (System.Type, System.String)** **System.Resources.ResourceManager.GetObject** **System.Resources.ResourceSet.GetObject**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Rich Edit Control Functions

Win32 Funtion	Description	.NET Framework API
EditStreamCallback	An application-defined callback function used with the **EM_STREAMIN** and **EM_STREAMOUT** messages.	**System.Windows.Forms.RichTextBox.LoadFile** **System.Windows.Forms.RichTextBox.RichTextBox.SaveFile** **Note** Not a callback function.

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Scroll Bar Functions

Win32 function	Description	.NET Framework API
EnableScrollBar	Enables or disables one or both scroll bar arrows.	**System.Windows.Forms.ScrollBar.Enabled** **Note** Cannot enable/disable built-in scrollbars (auto-scrolling).
GetScrollBarInfo	Retrieves information about the specified scroll bar.	**System.Windows.Forms.ScrollBar.Value** **System.Windows.Forms.ScrollEventArgs.Type**
GetScrollInfo	Retrieves the parameters of a scroll bar.	**System.Windows.Forms.ScrollBar.Maximum** **System.Windows.Forms.ScrollBar.Minimum** **System.Windows.Forms.ScrollBar.Value**
GetScrollPos	Retrieves the current position of the scroll box.	**System.Windows.Forms.ScrollBar.Value**
GetScrollRange	Retrieves the current minimum and maximum scroll box positions.	**System.Windows.Forms.ScrollBar.Minimum** **System.Windows.Forms.ScrollBar.Maximum**
SetScrollInfo	Sets the parameters of a scroll bar.	**System.Windows.Forms.ScrollBar.Maximum** **System.Windows.Forms.ScrollBar.Minimum** **System.Windows.Forms.ScrollBar.Value**
SetScrollPos	Sets the position of the scroll box.	**System.Windows.Forms.ScrollBar.Value**
SetScrollRange	Sets the minimum and maximum scroll box position.	**System.Windows.Forms.ScrollBar.Minimum** **System.Windows.Forms.ScrollBar.Maximum**
ShowScrollBar	Shows or hides the specified scroll bar.	Any of the following: **System.Windows.Forms.Form.AutoScroll** **System.Windows.Forms.HScrollBar** constructor **System.Windows.Forms.ListBox.HorizontalScrollBar** **System.Windows.Forms.ListBox.ScrollAlwaysVisible** **System.Windows.Forms.RichTextBox.ScrollBars** **System.Windows.Forms.ScrollableControl.HScroll** **System.Windows.Forms.ScrollableControl.VScroll** **System.Windows.Forms.TextBox.ScrollBars** **System.Windows.Forms.VScrollBar** constructor

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Service Functions

Functions implemented by services:

Win32 function	Description	.NET Framework API
Handler	An application-defined callback function used with **RegisterServiceCtrlHandler**.	All of the following: **System.ServiceProcess.ServiceBase.OnContinue** **System.ServiceProcess.ServiceBase.OnCustomCommand** **ServiceProcess.ServiceBase.OnPause** **System.ServiceProcess.ServiceBase.OnShutdown** **System.ServiceProcess.ServiceBase.OnStop**
HandlerEx	An application-defined callback function used with **RegisterServiceCtrlHandlerEx**.	All of the following: **System.ServiceProcess.ServiceBase.OnContinue** **System.ServiceProcess.ServiceBase.OnCustomCommand** **System.ServiceProcess.ServiceBase.OnPause** **System.ServiceProcess.ServiceBase.OnPowerEvent** **System.ServiceProcess.ServiceBase.OnShutdown** **System.ServiceProcess.ServiceBase.OnStop**
ServiceMain	An application-defined function that serves as the entry point for a service.	**System.ServiceProcess.ServiceBase.Run**

Functions used by programs that control or configure services:

Win32 function	Description	.NET Framework API
ChangeServiceConfig	Changes the configuration	**System.ServiceProcess.ServiceController.DisplayName**

	parameters of a service.	
CloseServiceHandle	Closes the specified handle to a service control manager object or a service object.	**System.ServiceProcess.ServiceController.Close**
ControlService	Sends a control code to a service.	**System.ServiceProcess.ServiceController.Continue** **System.ServiceProcess.ServiceController.Pause** **System.ServiceProcess.ServiceController.Stop**
CreateService	Creates a service object and adds it to the specified service control manager database.	**System.ServiceProcess.ServiceBase.ServiceName** **System.ServiceProcess.ServiceInstaller.Install**
DeleteService	Marks the specified service for deletion from the service control manager database.	**System.ServiceProcess.ServiceInstaller.Uninstall**
EnumDependentServices	Retrieves the name and status of each service that depends on the specified service.	**System.ServiceProcess.ServiceController.DependentServices**
EnumServicesStatus	Enumerates services in the specified service control manager database.	**System.ServiceProcess.ServiceController.GetServices**
GetServiceDisplayName	Retrieves the display name of the specified service.	**System.ServiceProcess.ServiceController.DisplayName**
GetServiceKeyName	Retrieves the name of the specified service.	**System.ServiceProcess.ServiceBase.ServiceName** **System.ServiceProcess.ServiceController.ServiceName**
OpenService	Opens an existing service.	**System.ServiceProcess.ServiceController** constructor
QueryServiceConfig	Retrieves the configuration parameters of the specified service.	**System.ServiceProcess.ServiceController.DisplayName** **System.ServiceProcess.ServiceController.ServicesDependedOn** **System.ServiceProcess.ServiceController.ServiceName** **System.ServiceProcess.ServiceController.ServiceType**
QueryServiceStatus	Retrieves the current status of the specified service.	**System.ServiceProcess.ServiceController.Status**
StartService	Starts a service.	**System.ServiceProcess.ServiceController.Start**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

String Functions

Win32 function	Description	.NET Framework API
CharLower	Converts a character string or a single character to lowercase.	**System.String.ToLower**
CharNext	Retrieves a pointer to the next	**System.String.GetEnumerator.MoveNext**

	character in a string.	
CharUpper	Converts a character string or a single character to uppercase.	**System.String.ToUpper**
CompareString	Compares two character strings, using the specified locale.	**System.String.Compare**
IsCharAlpha	Determines whether a character is an alphabetical character.	**System.Char.IsLetter**
IsCharAlphaNumeric	Determines whether a character is either an alphabetical or a numeric character.	**System.Char.IsLetterOrDigit**
IsCharLower	Determines whether a character is lowercase.	**System.Char.IsLower**
IsCharUpper	Determines whether a character is uppercase.	**System.Char.IsUpper**
LoadString	Loads a string resource from the executable file associated with a specified module.	**System.Resources.ResourceManager.GetString**
lstrcat	Appends one string to another.	**System.String.Concat** **Note** Strings are immutable in the .NET Framework. Methods that appear to modify a string actually return a new string.
lstrcmp	Compares two character strings.	**System.String.Compare**
lstrcmpi	Compares two character strings.	**System.String.Compare**
lstrcpy	Copies a string to a buffer.	**System.String.ToCharArray**
lstrcpyn	Copies a specified number of characters from a source string into a buffer.	**System.String.CopyTo** **System.String.ToCharArray**
lstrlen	Retrieves the length of the specified string (not including the terminating null character).	**System.String.Length**
wsprintf	Formats and stores a series of characters and values in a buffer.	**System.String.Format**
wvsprintf	Formats and stores a series of characters and values in a buffer.	**System.String.Format**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Structured Exception-Handling Functions

Win32 function	Description	.NET Framework API
GetExceptionCode	Retrieves a code that identifies the type of exception that occurred.	**System.Exception.GetType**
GetExceptionInformation	Retrieves a machine-independent description of an exception and information about the machine state that existed for the thread when the exception occurred.	**System.Exception** class
RaiseException	Raises an exception in the calling thread.	Throw an instance of a **System.Exception**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Synchronization Functions

CriticalSection functions:

Win32 function	Description	.NET Framework API
EnterCriticalSection	Waits for ownership of the specified critical section object.	**System.Threading.Monitor.Enter**
LeaveCriticalSection	Releases ownership of the specified critical section object.	**System.Threading.Monitor.Exit**

TryEnterCriticalSection	Attempts to enter a critical section without blocking.	**System.Threading.Monitor.TryEnter**

Event functions:

Win32 function	Description	.NET Framework API
CreateEvent	Creates or opens a named or an unnamed event object.	Either of the following: **System.Threading.AutoResetEvent** constructor **System.Threading.ManualResetEvent** constructor
ResetEvent	Sets the specified event object to the nonsignaled state.	Either of the following: **System.Threading.AutoResetEvent.Reset** **System.Threading.ManualResetEvent.Reset**
SetEvent	Sets the specified event object to the signaled state.	Either of the following: **System.Threading.AutoResetEvent.Set** **System.Threading.ManualResetEvent.Set**

Interlocked functions:

Win32 function	Description	.NET Framework API
InterlockedCompareExchange	Performs an atomic comparison of the specified values and exchanges the values, based on the outcome of the comparison.	**System.Threading.Interlocked.CompareExchange**
InterlockedCompareExchangePointer	Performs an atomic comparison of the specified values and exchange of the values, based on the outcome of the comparison.	**System.Threading.Interlocked.CompareExchange**
InterlockedDecrement	Decrements (decreases by one) the value of the specified variable and checks the resulting value.	**System.Threading.Interlocked.Decrement**
InterlockedExchange	Atomically exchanges a pair of values.	**System.Threading.Interlocked.Exchange**
InterlockedIncrement	Increments (increases by one) the value of the specified variable and checks the resulting value.	**System.Threading.Interlocked.Increment**

Mutex functions:

Win32 function	Description	.NET Framework API
CreateMutex	Creates or opens a named or an unnamed mutex object.	**System.Threading.Mutex** constructor
OpenMutex	Opens an existing named mutex object.	**System.Threading.Mutex(false, string, name)** constructor
ReleaseMutex	Releases ownership of the specified mutex object.	**System.Threading.Mutex.ReleaseMutex**

Wait functions:

Win32 function	Description	.NET Framework API
RegisterWaitForSingleObject	Directs a wait thread in the thread pool to wait on the object.	**System.Threading.ThreadPool.RegisterWaitForSingleObject** **System.Threading.ThreadPool.UnsafeRegisterWaitForSingleObjec**
UnregisterWait	Cancels a registered wait operation.	**System.Threading.RegisteredWaitHandle.Unregister**
WaitForMultipleObjects	Returns when the specified criteria for the specified objects is met.	Either of the following: **System.Threading.WaitHandle.WaitAll** **System.Threading.WaitHandle.WaitAny**
WaitForSingleObject	Returns when the specified criteria for the specified object is met.	**System.Threading.WaitHandle.WaitOne**
WaitOrTimerCallback	Returns when the specified criteria is met.	See **System.Threading.WaitOrTimerCallback** delegate

WaitableTimer functions:

Win32 function	Description	.NET Framework API
CancelWaitableTimer	Sets the specified waitable timer to the inactive state.	**System.Threading.Timer.Change (System.Threading.Timeout.Infinite, System.Threading.Timeout.Infinite)**
CreateWaitableTimer	Creates or opens a waitable timer object.	**System.Threading.Timer** constructor
SetWaitableTimer	Activates the specified waitable timer.	**System.Threading.Timer** constructor
TimerAPCProc	Application-defined timer completion routine used with **SetWaitableTimer**.	See **System.Threading.TimerCallback** delegate

Alphabetical Win32 Category List
Hierarchical Win32 Category List

System Information Functions

Win32 function	Description	.NET Framework API
ExpandEnvironmentStrings	Replaces environment-variable strings with their defined values.	**System.Environment.ExpandEnvironmentVariables**
GetComputerName	Retrieves the NetBIOS name of the local computer.	**System.Environment.MachineName** **System.Windows.Forms.SystemInformation.ComputerName**
GetCurrentHwProfile	Retrieves the current hardware profile for the local computer.	Use **System.Management** and **Microsoft.Win32.RegistryKey**

GetKeyboardType	Retrieves information about the current keyboard.	**System.Management.ManagementClass ("Win32_Keyboard")** **System.Management.ManagementClass.GetInstances** **System.Management.ManagementObject.Properties**
GetNativeSystemInfo	Retrieves information about the current system for an application running under WOW64.	**System.Management.ManagementClass ("Win32_Processor")** **System.Management.ManagementClass.GetInstances** **System.Management.ManagementObject.Properties**
GetSysColor	Retrieves the current color of a display element.	**System.Drawing.SystemColors**
GetSystemDirectory	Retrieves the path of the system directory.	**System.Environment.SystemDirectory**
GetSystemInfo	Retrieves information about the current system.	**System.Management.ManagementClass ("Win32_Processor")** **System.Management.ManagementClass.GetInstances** **System.Management.ManagementObject.Properties**
GetSystemMetrics	Retrieves system metrics and configuration settings.	**System.Windows.Forms.SystemInformation** properties
GetUserName	Retrieves the user name of the current thread.	**System.Environment.UserName**
GetVersion	Retrieves the version number of the operating system.	**System.Environment.OSVersion**
GetVersionEx	Retrieves the version number of the current operating system.	**System.Environment.OSVersion**
GetWindowsDirectory	Retrieves the path of the Windows directory.	**System.Management.ManagementClass ("Win32_OperatingSystem")** **System.Management.ManagementClass.GetInstances** **System.Management.ManagementObject ["WindowsDirectory"].ToString**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Tape Backup Functions

Win32 function	Description	.NET Framework API
GetTapeParameters	Retrieves information that describes the tape or the tape drive.	**System.Management.ManagementClass ("MSTapeMediaCapacity")** **System.Management.ManagementClass ("Win32_TapeDrive")** **System.Management.ManagementClass.GetInstances** **System.Management.ManagementObject.Properties**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Text Object Model Functions

Win32 function	Description	.NET Framework API
ITextDocument	Retrieves the active selection and range objects for any story in the document.	**System.Windows.Forms.RichTextBox**

ITextFont	Encapsulates the functionality of the Microsoft Word **Font** dialog box.	**System.Windows.Forms.RichTextBox**
ITextPara	Encapsulates the functionality of the Microsoft Word **Paragraph** dialog box.	**System.Windows.Forms.RichTextBox**
ITextRange	Enables an application to select text in a story, and then examine or change that text.	**System.Windows.Forms.RichTextBox**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Time Functions

Win32 function	Description	.NET Framework API
GetSystemTime	Retrieves the current system date and time in UTC format.	**System.DateTime.UtcNow**
GetTimeFormat	Formats a system time as a time string for a specified locale.	**System.DateTime.ToString**
SystemTimeToFileTime	Converts a system time to a file time.	**System.DateTime.ToFileTime**
FileTimeToLocalFileTime	Converts a UTC file time to a local file time.	**System.DateTime.ToFileTime**
GetLocalTime	Retrieves the current local date and time.	**System.DateTime.Now**
GetTimeZoneInformation	Retrieves the current time-zone parameters.	**System.TimeZone.CurrentTimeZone**
CompareFileTime	Compares two file times.	**System.DateTime.Compare**
FileTimeToLocalFileTime	Converts a UTC file time to a local file time.	**System.DateTime.ToLocalTime**
FileTimeToSystemTime	Converts a file time to system time format.	**System.DateTime.ToUniversalTime**
GetFileTime	Retrieves the date and time that a file was created, last accessed, and last modified.	Any of the following: **System.IO.Directory.GetCreationTime** **System.IO.Directory.GetLastAccessTime** **System.IO.Directory.GetLastWriteTime** **System.IO.DirectoryInfo.CreationTime** **System.IO.DirectoryInfo.LastAccessTime** **System.IO.DirectoryInfo.LastWriteTime** **System.IO.File.GetCreationTime** **System.IO.File.GetLastAccessTime** **System.IO.File.GetLastWriteTime** **System.IO.FileInfo.CreationTime** **System.IO.FileInfo.LastAccessTime** **System.IO.FileInfo.LastWriteTime**
GetSystemTimeAsFileTime	Retrieves the current system date and time in UTC format.	**System.DateTime.UtcNow.ToFileTimeUtc**
LocalFileTimeToFileTime	Converts a local file time to a file time based on UTC.	**System.DateTime.ToFileTimeUtc**
SetFileTime	Sets the date and time that a file was created, last accessed, or last modified.	Any of the following: **System.IO.Directory.SetCreationTime** **System.IO.Directory.SetLastAccessTime** **System.IO.Directory.SetLastWriteTime** **System.IO.DirectoryInfo.CreationTime** **System.IO.DirectoryInfo.LastAccessTime** **System.IO.DirectoryInfo.LastWriteTime** **System.IO.File.SetCreationTime** **System.IO.File.SetLastAccessTime** **System.IO.File.SetLastWriteTime** **System.IO.FileInfo.CreationTime** **System.IO.FileInfo.LastAccessTime** **System.IO.FileInfo.LastWriteTime**
SystemTimeToFileTime	Converts a system time to a file time.	**System.DateTime.ToFileTime**
GetTickCount	Retrieves the number of milliseconds	**System.Environment.TickCount**

	that have elapsed since the system was started.	

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Timer Functions

Win32 function	Description	.NET Framework API
KillTimer	Destroys the specified timer.	**System.Threading.Timer.Dispose** **System.Timers.Timer.Dispose** **System.Windows.Forms.Timer.Dispose**
QueryPerformanceCounter	Retrieves the current value of the high-resolution performance counter.	**System.Diagnostics.PerformanceCounter.NextValue** **System.Diagnostics.PerformanceCounter.RawValue**
QueryPerformanceFrequency	Retrieves the frequency of the high-resolution performance counter.	**System.Diagnostics.PerformanceCounter.CounterType**
SetTimer	Creates a timer with the specified time-out value.	Any of the following: **System.Threading.Timer** constructor **System.Timers.Timer** constructor **System.Windows.Forms.Timer** constructor **System.Windows.Forms.Timer.Interval**
TimerProc	An application-defined callback function that processes **WM_TIMER** messages.	Either of the following: **System.Threading.TimerCallback** delegate **System.Timers.ElapsedEventHandler** delegate

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Tool Help Functions

Win32 function	Description	.NET Framework API
CreateToolhelp32Snapshot	Takes a snapshot of the processes and heaps, modules, and threads used by the processes.	**System.Diagnostics.Process.GetProcesses** **System.Diagnostics.Process.Modules** **System.Diagnostics.Process.Threads**
Module32First	Retrieves information about the first module associated with a process.	**System.Diagnostics.Process.Modules**
Module32Next	Retrieves information about the next module associated with a process or thread.	**System.Diagnostics.Process.Modules**
Process32First	Retrieves information about the first process encountered in a system snapshot.	**System.Diagnostics.Process.GetProcesses**
Process32Next	Retrieves information about the next process recorded in a system snapshot.	**System.Diagnostics.Process.GetProcesses**
Thread32First	Retrieves information about the first thread of any process encountered in a system snapshot.	**System.Diagnostics.Process.Threads**
Thread32Next	Retrieves information about the next thread of any process encountered in the system memory snapshot.	**System.Diagnostics.Process.Threads**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Unicode and Character Set Functions

Win32 function	Description	.NET Framework API
GetTextCharset	Retrieves a character-set identifier for the font that is currently selected into a specified device context.	**System.Drawing.Font.GdiCharSet**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

User Profile Functions

Win32 function	Description	.NET Framework API
CreateEnvironmentBlock	Retrieves the environment variables for the specified user.	**Microsoft.Win32.Registry** class **Microsoft.Win32.RegistryKey.OpenSubKey(name).GetValue**
DeleteProfile	Deletes the user profile and all user-related settings from the specified computer.	**System.IO.Directory.Delete** **Microsoft.Win32.Registry** **Microsoft.Win32.RegistryKey.DeleteSubKey**
ExpandEnvironmentStringsForUser	Expands the source string by using the environment block established for the specified user.	**Microsoft.Win32.Registry** class **Microsoft.Win32.RegistryKey.OpenSubKey(name).GetValueNames**, and then call **GetValue** for each name
GetAllUsersProfileDirectory	Retrieves the path to the root of the All Users profile.	**Microsoft.Win32.Registry** class **Microsoft.Win32.RegistryKey.OpenSubKey(name).GetValue**
GetDefaultUserProfileDirectory	Retrieves the path to the root of the Default User profile.	**Microsoft.Win32.Registry** class **Microsoft.Win32.RegistryKey.OpenSubKey(name).GetValue**
GetProfilesDirectory	Retrieves the path to the root directory where all of the user profiles are stored.	**Microsoft.Win32.Registry** class **Microsoft.Win32.RegistryKey.OpenSubKey(name).GetValue**
GetUserProfileDirectory	Retrieves the path to the root directory of the profile for the specified user.	**Microsoft.Win32.Registry** class **Microsoft.Win32.RegistryKey.OpenSubKey(name).GetValue**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Version Information Functions

Win32 function	Description	.NET Framework API
GetFileVersionInfo	Retrieves version information for the specified file.	**System.Diagnostics.FileVersionInfo.GetVersionInfo**
GetFileVersionInfoSize	Determines whether the operating system can retrieve version information for a specified file. If version information is available, the function returns the size, in bytes, of that information.	**System.Diagnostics.FileVersionInfo.GetVersionInfo**
VerLanguageName	Retrieves a description string for the language associated with a specified binary Microsoft language identifier.	**System.Globalization.CultureInfo.Name**
VerQueryValue	Retrieves specified version information from the specified version-information resource.	**System.Diagnostics.FileVersionInfo** properties

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Window Functions

Win32 function	Description	.NET Framework API
AdjustWindowRect	Calculates the required size of the window rectangle, based on the desired client-rectangle size.	**System.Windows.Forms.Form.ClientSize**
AdjustWindowRectEx	Calculates the required size of the window rectangle, based on the desired client-rectangle size.	**System.Windows.Forms.Form.ClientSize**
ArrangeIconicWindows	Arranges all minimized (iconic) child windows of the specified parent window.	**System.Windows.Forms.Form.LayoutMdi(MdiLayout.Arra**
BringWindowToTop	Brings the specified window to the top of the Z order.	**System.Windows.Forms.Form.BringToFront**
CascadeWindows	Cascades the specified child windows of the specified parent window.	**System.Windows.Forms.Form.LayoutMdi(MdiLayout.Cas**
ChildWindowFromPoint	Determines which, if any, of the child windows belonging to a parent window contains the specified point.	**System.Windows.Forms.Form.GetChildAtPoint**
CloseWindow	Minimizes (but does not destroy) the specified window.	**System.Windows.Forms.Form.WindowState = System.Windows.Forms.FormWindowState.Minimized**
CreateWindow	Creates an overlapped, pop-up, or child window.	**System.Windows.Forms.Form** constructor
CreateWindowEx	Creates an overlapped, pop-up, or child window with an extended window style.	**System.Windows.Forms.Form** constructor
DestroyWindow	Destroys the specified window.	**System.Windows.Forms.Form.Close**
EnumChildProc	An application-defined callback function used with **EnumChildWindows**.	Same as **EnumChildWindows**
EnumChildWindows	Enumerates the child windows that belong to the specified parent window.	**System.Windows.Forms.Form.Controls**
GetAncestor	Retrieves the handle to the ancestor of the specified window.	**System.Windows.Forms.Form.FindForm** **System.Windows.Forms.Form.Owner** **System.Windows.Forms.Form.Parent**
GetClientRect	Retrieves the coordinates of the client area for a window.	**System.Windows.Forms.Form.ClientRectangle**
GetForegroundWindow	Returns a handle to the foreground window.	**System.Windows.Forms.Form.ActiveForm**
GetLayeredWindowAttributes	Retrieves the opacity	**System.Windows.Forms.Form.Opacity**

	and transparency color key of a layered window.	**System.Windows.Forms.Form.TransparencyKey**
GetNextWindow	Retrieves a handle to the next or previous window in Z order.	**System.Windows.Forms.Form.Controls[1 +/- Control.In**
GetParent	Retrieves a handle to the parent window for the specified child window.	**System.Windows.Forms.Form.FindForm** **System.Windows.Forms.Form.Parent**
GetTitleBarInfo	Retrieves information about the specified title bar.	**System.Windows.Forms.Form.SystemInformation.Captio** **System.Windows.Forms.Form.SystemInformation.Captio** **System.Windows.Forms.Form.HelpButton** **System.Windows.Forms.Form.MaximizeBox** **System.Windows.Forms.Form.MinimizeBox**
GetTopWindow	Retrieves a handle to the child window at the top of the Z order.	**System.Windows.Forms.Form.Controls[0]**
GetWindow	Retrieves a handle to a window that has the specified relationship to the specified window.	**System.Windows.Forms.Form.Controls**
GetWindowInfo	Retrieves information about the specified window.	**System.Windows.Forms.Form.ClientSize** **System.Windows.Forms.Form.CreateParams.ExStyle** **System.Windows.Forms.Form.CreateParams.Style** **System.Windows.Forms.Form.Name** **System.Windows.Forms.Form.ProductVersion** **System.Windows.Forms.Form.Size** **System.Windows.Forms.SystemInformation.BorderSize**
GetWindowPlacement	Retrieves the show state and the restored, minimized, and maximized positions of the specified window.	**System.Windows.Forms.Form.Bounds** **System.Windows.Forms.Form.MaximizedBounds** **System.Windows.Forms.Form.Visible** **System.Windows.Forms.Form.WindowState**
GetWindowRect	Retrieves the dimensions of the bounding rectangle of the specified window.	**System.Windows.Forms.Form.Bounds**
GetWindowText	Copies the text of the title bar of the specified window.	**System.Windows.Forms.Form.Text**
GetWindowTextLength	Retrieves the length of the title bar text for the specified window.	**System.Windows.Forms.Form.Text.Length**
IsChild	Tests whether a window is a child window or descendant window of a specified parent window.	**System.Windows.Forms.Form.Contains** **System.Windows.Forms.Form.IsMdiChild**
IsIconic	Determines whether the specified window is minimized (iconic).	**System.Windows.Forms.Form.WindowState**
IsWindowVisible	Retrieves the visibility state of the specified window.	**System.Windows.Forms.Form.Visible**
IsZoomed	Determines whether a window is maximized (zoomed).	**System.Windows.Forms.Form.WindowState**
MoveWindow	Changes the position and dimensions of the specified window.	**System.Windows.Forms.Form.SetBounds**

OpenIcon	Restores a minimized (iconic) window to its previous size and position.	**System.Windows.Forms.Form.WindowState = System.Windows.Forms.FormWindowState.Normal**
SetForegroundWindow	Puts the thread that created the specified window into the foreground and activates the window.	**System.Windows.Forms.Form.Activate**
SetLayeredWindowAttributes	Sets the opacity and transparency color key of a layered window.	**System.Windows.Forms.Form.Opacity** **System.Windows.Forms.Form.TransparencyKey**
SetParent	Changes the parent window of the specified child window.	**System.Windows.Forms.Form.Parent**
SetWindowPlacement	Sets the show state and the restored, minimized, and maximized positions of the specified window.	**System.Windows.Forms.Form.Bounds** **System.Windows.Forms.Form.MaximizedBounds** **System.Windows.Forms.Form.SetBounds** **System.Windows.Forms.Form.Visible** **System.Windows.Forms.Form.WindowState**
SetWindowPos	Changes the size, position, and Z order of a window.	**System.Windows.Forms.Form.BringToFront** **System.Windows.Forms.Form.Enabled** **System.Windows.Forms.Form.SendToBack** **System.Windows.Forms.Form.SetBounds** **System.Windows.Forms.Form.TopMost** **System.Windows.Forms.Form.Visible**
SetWindowText	Changes the text of the title bar for the specified window.	**System.Windows.Forms.Form.Text**
ShowWindow	Sets the show state of the specified window.	**System.Windows.Forms.Form.Show** **System.Windows.Forms.Form.Visible** **System.Windows.Forms.Form.WindowState**
TileWindows	Tiles the specified child windows of the specified parent window.	**System.Windows.Forms.Form.LayoutMdi(MdiLayout.Tile** **System.Windows.Forms.Form.LayoutMdi(MdiLayout.Tile**
WindowFromPoint	Retrieves a handle to the window that contains the specified point.	**System.Windows.Forms.Form.GetChildAtPoint**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Window Class Functions

Win32 function	Description	.NET Framework API
GetClassInfo	Retrieves information about a window class.	**System.ComponentModel.Component.Events** **System.Windows.Forms.Control.Cursor** **System.Windows.Forms.Control.Name** **System.Windows.Forms.Form.BackColor** **System.Windows.Forms.Form.Icon** **System.Windows.Forms.Form.Menu** **System.Windows.Forms.Form.CreateParams.ClassStyle**
GetClassLong	Retrieves the specified 32-bit value from the **WNDCLASSEX** structure associated with the specified window.	**System.ComponentModel.Component.Events** **System.Windows.Forms.Control.Cursor** **System.Windows.Forms.Control.Name** **System.Windows.Forms.Form.BackColor** **System.Windows.Forms.Form.Icon** **System.Windows.Forms.Form.Menu** **System.Windows.Forms.Form.CreateParams.ClassStyle**
GetClassLongPtr	Retrieves the specified value	**System.ComponentModel.Component.Events**

	from the **WNDCLASSEX** structure associated with the specified window.	**System.Windows.Forms.Control.Cursor** **System.Windows.Forms.Control.Name** **System.Windows.Forms.Form.BackColor** **System.Windows.Forms.Form.Icon** **System.Windows.Forms.Form.Menu** **System.Windows.Forms.Form.CreateParams.ClassStyle**
GetClassName	Retrieves the name of the class to which the specified window belongs.	**System.Windows.Forms.Control.Name**
GetWindowLong	Retrieves information about the specified window.	**System.ComponentModel.Component.Events** **System.Windows.Forms.Control.Parent** **System.Windows.Forms.Form.CreateParams.ExStyle** **System.Windows.Forms.Form.CreateParams.Style** **System.Windows.Forms.Form** properties
GetWindowLongPtr	Retrieves information about the specified window.	**System.ComponentModel.Component.Events** **System.Windows.Forms.Control.Parent** **System.Windows.Forms.Form.CreateParams.ExStyle** **System.Windows.Forms.Form.CreateParams.Style** **System.Windows.Forms.Form** properties
RegisterClass	Registers a window class.	**System.Windows.Forms.Control.Cursor** **System.Windows.Forms.Control.Name** **System.Windows.Forms.Form.BackColor** **System.Windows.Forms.Form.Icon** **System.Windows.Forms.Form.Menu** **System.Windows.Forms.Form.CreateParams.ClassStyle** **System.Windows.Forms.Form** events
SetClassLong	Replaces the specified 32-bit value at the specified offset into the extra class memory or the **WNDCLASSEX** structure.	**System.Windows.Forms.Control.Cursor** **System.Windows.Forms.Control.Name** **System.Windows.Forms.Form.BackColor** **System.Windows.Forms.Form.Icon** **System.Windows.Forms.Form.Menu** **System.Windows.Forms.Form.CreateParams.ClassStyle** **System.Windows.Forms.Form** events
SetClassLongPtr	Replaces the specified value at the specified offset into the extra class memory or the **WNDCLASSEX** structure.	**System.Windows.Forms.Control.Cursor** **System.Windows.Forms.Control.Name** **System.Windows.Forms.Form.BackColor** **System.Windows.Forms.Form.Icon** **System.Windows.Forms.Form.Menu** **System.Windows.Forms.Form.CreateParams.ClassStyle** **System.Windows.Forms.Form** events
SetWindowLong	Changes an attribute of the specified window.	**System.Windows.Forms.Control.Parent** **System.Windows.Forms.Form.CreateParams.ExStyle** **System.Windows.Forms.Form.CreateParams.Style** **System.Windows.Forms.Form** properties **System.Windows.Forms.Form** events
SetWindowLongPtr	Changes an attribute of the specified window.	**System.Windows.Forms.Control.Parent** **System.Windows.Forms.Form.CreateParams.ExStyle** **System.Windows.Forms.Form.CreateParams.Style** **System.Windows.Forms.Form** properties **System.Windows.Forms.Form** events

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Window Procedure Functions

Win32 function	Description	.NET Framework API
DefWindowProc	Calls the default window procedure to provide default processing for any window messages that an application does not process.	Any of the following: **System.Windows.Forms.Form.DefWndProc** **System.Windows.Forms.Control.DefWndProc** **System.Windows.Forms.NativeWindow.DefWndProc**
WindowProc	An application-defined function that processes messages sent to a window.	Any of the following: **System.Windows.Forms.Form.WndProc** **System.Windows.Forms.Control.WndProc**

		System.Windows.Forms.NativeWindow.WndProc

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Window Property Functions

Win32 function	Description	.NET Framework API
GetProp	Retrieves a data handle from the property list of the specified window.	**System.Windows.Forms.Control.Tag**
RemoveProp	Removes an entry from the property list of the specified window.	**System.Windows.Forms.Control.Tag**
SetProp	Adds a new entry or changes an existing entry in the property list of the specified window.	**System.Windows.Forms.Control.Tag**

Alphabetical Win32 Category List
Hierarchical Win32 Category List

Print E-Mail Add to Favorites

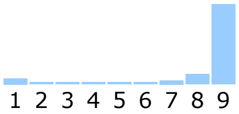
How would you rate the quality of this content?

	1	2	3	4	5	6	7	8	9	
Poor	○	○	○	○	○	○	○	○	○	Outstanding

Tell us why you rated the content this way. (optional)

Submit

Average rating:
8 out of 9



881 people have rated this page

Manage Your Profile | Legal | Contact Us | MSDN Flash Newsletter