xposes methods that allow a client to retrieve the icon that is associated with one of the objects in a folder.

**Members**

The **IExtractIcon** interface inherits from the [**IUnknown**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms680509%28v=vs.85%29.aspx) interface. **IExtractIcon** also has these types of members:

**IExtractIcon::Extract method**

Extracts an icon image from the specified location.

**Syntax**

C++

HRESULT Extract(

[in]             PCTSTR pszFile,

UINT nIconIndex,

[out, optional]  HICON \*phiconLarge,

[out, optional]  HICON \*phiconSmall,

UINT nIconSize

);

**Parameters**

*pszFile* [in]

Type: **PCTSTR**

A pointer to a null-terminated string that specifies the icon location.

*nIconIndex*

Type: **UINT**

The index of the icon in the file pointed to by *pszFile*.

*phiconLarge* [out, optional]

Type: **HICON\***

A pointer to an **HICON** value that receives the handle to the large icon. This parameter may be **NULL**.

*phiconSmall* [out, optional]

Type: **HICON\***

A pointer to an **HICON** value that receives the handle to the small icon. This parameter may be **NULL**.

*nIconSize*

Type: **UINT**

The desired size of the icon, in pixels. The low word contains the size of the large icon, and the high word contains the size of the small icon. The size specified can be the width or height. The width of an icon always equals its height.

**Return value**

Type: **HRESULT**

Returns S\_OK if the function extracted the icon, or S\_FALSE if the calling application should extract the icon.

**IExtractIcon::GetIconLocation method**

Gets the location and index of an icon.

**Syntax**

C++

HRESULT GetIconLocation(

UINT uFlags,

[out]  PTSTR pszIconFile,

UINT cchMax,

[out]  int \*piIndex,

[out]  UINT \*pwFlags

);

**Parameters**

*uFlags*

Type: **UINT**

One or more of the following values. This parameter can also be **NULL**.

**GIL\_ASYNC** (0x0020)

Set this flag to determine whether the icon should be extracted asynchronously. If the icon can be extracted rapidly, this flag is usually ignored. If extraction will take more time, **GetIconLocation** should return E\_PENDING. See the Remarks for further discussion.

**GIL\_DEFAULTICON** (0x0040)

Retrieve information about the fallback icon. Fallback icons are usually used while the desired icon is extracted and added to the cache.

**GIL\_FORSHELL** (0x0002)

The icon is displayed in a Shell folder.

**GIL\_FORSHORTCUT** (0x0080)

The icon indicates a shortcut. However, the icon extractor should not apply the shortcut overlay; that will be done later. Shortcut icons are state-independent.

**GIL\_OPENICON** (0x0001)

The icon is in the open state if both open-state and closed-state images are available. If this flag is not specified, the icon is in the normal or closed state. This flag is typically used for folder objects.

**GIL\_CHECKSHIELD** (0x0200)

Explicitly return either GIL\_SHIELD or GIL\_FORCENOSHIELD in *pwFlags*. Do not block if GIL\_ASYNC is set.

*pszIconFile* [out]

Type: **PTSTR**

A pointer to a buffer that receives the icon location. The icon location is a null-terminated string that identifies the file that contains the icon.

*cchMax*

Type: **UINT**

The size of the buffer, in characters, pointed to by *pszIconFile*.

*piIndex* [out]

Type: **int\***

A pointer to an **int** that receives the index of the icon in the file pointed to by *pszIconFile*.

*pwFlags* [out]

Type: **UINT\***

A pointer to a **UINT** value that receives zero or a combination of the following values.

**GIL\_DONTCACHE** (0x0010)

The physical image bits for this icon are not cached by the calling application.

**GIL\_NOTFILENAME** (0x0008)

The location is not a file name/index pair. The values in *pszIconFile* and *piIndex* cannot be passed to [**ExtractIcon**](http://msdn.microsoft.com/en-us/library/windows/desktop/bb776416%28v=vs.85%29.aspx) or [**ExtractIconEx**](http://msdn.microsoft.com/en-us/library/windows/desktop/bb776417%28v=vs.85%29.aspx).

When this flag is omitted, the value returned in *pszIconFile* is a fully-qualified path name to either a .ico file or to a file that can contain icons. Also, the value returned in *piIndex* is an index into that file that identifies which of its icons to use. Therefore, when the GIL\_NOTFILENAME flag is omitted, these values can be passed to [**ExtractIcon**](http://msdn.microsoft.com/en-us/library/windows/desktop/bb776416%28v=vs.85%29.aspx) or [**ExtractIconEx**](http://msdn.microsoft.com/en-us/library/windows/desktop/bb776417%28v=vs.85%29.aspx).

**GIL\_PERCLASS** (0x0004)

All objects of this class have the same icon. This flag is used internally by the Shell. Typical implementations of [**IExtractIcon**](http://msdn.microsoft.com/en-us/library/windows/desktop/bb761854%28v=vs.85%29.aspx) do not require this flag because the flag implies that an icon handler is not required to resolve the icon on a per-object basis. The recommended method for implementing per-class icons is to register a DefaultIcon for the class.

**GIL\_PERINSTANCE** (0x0002)

Each object of this class has its own icon. This flag is used internally by the Shell to handle cases like Setup.exe, where objects with identical names can have different icons. Typical implementations of [**IExtractIcon**](http://msdn.microsoft.com/en-us/library/windows/desktop/bb761854%28v=vs.85%29.aspx) do not require this flag.

**GIL\_SIMULATEDOC** (0x0001)

The calling application should create a document icon using the specified icon.

**GIL\_SHIELD** (0x0200)

**Windows Vista only**. The calling application must stamp the icon with the UAC shield.

**GIL\_FORCENOSHIELD** (0x0400)

**Windows Vista only**. The calling application must not stamp the icon with the UAC shield.

**Return value**