WebPost Overview

The components of the WebPost Software Development Kit (SDK) allow authoring tools to easily post web pages (files) to the user's Internet web site. The WebPost functions can be used to connect to the Internet Service Provider (ISP), determine the protocol needed to copy the files, and so on. Optionally, these functions can also display a wizard to guide the user through posting a file.

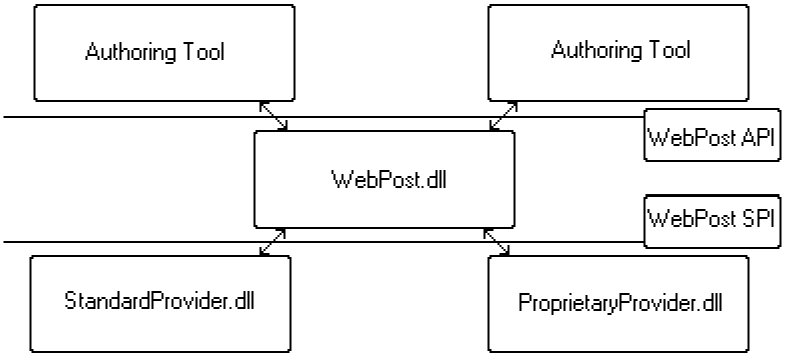
WebPost API

The WebPost application programming interface (API) makes it possible for authoring tools to post web pages to an Internet site with just one call to the WpPost function. In a typical scenario, an authoring tool's **File** menu would include a **Post to Web** button that, when clicked, causes the tool to call the WpPost function. This function starts a wizard that asks the user for a friendly name for the Internet site, the Uniform Resource Locator (URL) for the given site, and the name of a dial-up connection for accessing the Internet. Next, the function connects to the Internet server at the given URL, determines the protocol to use for posting the web pages, and then posts the requested files. For subsequent posting to the same site, this function remembers the details of how to connect to the site, and posts the files with little user intervention. Thus, the WebPost API maintains an association between a friendly site name, remembers all the details involved in posting to that site, and allows the applications to easily post web pages to a site or to the URL associated with that site.

As an alternative to the wizard, an authoring tool can let the user choose the web site before calling WpPost by displaying a list of sites (plus a New Site item) obtained by a call to the WpListSites function. The list of sites could be in a nested menu that appears when the user clicks the **Post to Web** button.

WebPost Service Providers

The WebPost dynamic-link library (DLL) can post web pages to some of the popular types of Internet servers, including the National Center for Supercomputing Applications' (NCSA's) *httpd* and Microsoft's Internet Information Server (IIS). To post to other types of Internet servers, the WebPost API uses the WebPost Service Provider Interface (SPI) to communicate with DLLs that "know how" to post web pages to those servers.



A WebPost service provider is implemented as an OLE Component Object Model (COM) server. An authoring tool can take advantage of the functionality available from a WebPost service provider by calling functions implemented by the provider. An authoring tool uses the WpBindToSite function to retrieve the addresses of the provider's functions. For descriptions of the functions that a provider implements, see WebPost SPI Interface Functions.

Posting Information File

Because of variations among Internet servers, Microsoft requests that Internet Service Providers (ISPs) include a posting information file named postinfo.html on the root URLs of their Internet servers. (Because filenames are case-sensitive on some servers, the name of the postinfo.html file should be in all lowercase letters.) This file contains details about the posting protocol and policy. The standard WebPost service provider uses the information in the file to help with the detection of the WebPost protocol.

The following is a sample postinfo.html file that the default provider uses. In the comments, the page contains configuration information for authoring tools that use the Microsoft WebPost functions:

```
<html>
<!-- postinfo.html version 0.100 -->

<head>
<title>
    Web Posting Information
</title>
<head>

<body>
<!--
    WebPost
    version="0.100"
    BaseURL="http://<servername>/~$USERNAME"
    BasePath="public_html"
    FtpServerName="<servername>"
    XferType="FTP"
    DefaultPage="default.htm"
    VerifyFiles="1"
    CreateRoot="1"
-->

<h1>
    Web Posting Information
</h1>
```

Provider DLL Installation

Each provider should create a subkey in the following registry location:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
    \WebPost\Providers
```

The subkey consists of the WpSite class identifier (CLSID) that the provider exports. Within this subkey, you should enter the following values:

Value	Meaning
Description	A string that contains the name of the provider. This name is used in the wizard; it appears in the list box that contains the names of the providers.
Path	The path of the WebPost service provider DLL.
Priority	A doubleword value that indicates the provider's priority. A value of 0 is high priority. The default provider has a priority of 8192.

The following example shows the registry entries for the default provider:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Providers
    \{3151E2E0-6C4C-11CF-86B1-00AA0060F86C}
        "Provider"="Other Internet Provider"
        "Path"="defwpp.dll"
        "Priority"=hex:00,10,00,00
```

All providers should also register their class identifiers at the following registry location, with the value name set to the class identifier. WebPost functions use the entries in this registry location to determine which providers to load.

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\WebPost\Providers
```

The default provider entry at this location is as follows:

```
"{3151E2E0-6C4C-11CF-86B1-00AA0060F86C}"=""
```

WebPost API Functions

This section describes the functions in the WebPost API.

The functions for simple usage include:

WpDeleteSite Deletes a friendly site name that has been configured.

WpListSites Retrieves a list of the friendly site names that the user has configured.

WpPost Posts a file name to the URL at a given site. If the URL and site name are set to NULL, the function starts a wizard that asks the user to choose or set up a URL and site.

The functions for advanced usage include:

WpBindToSite Returns a COM object to the WebPost service provider that supports the given site name or URL. With this object, the application can call into the provider functions listed in the WebPost SPI section.

A fifth WebPost API function, WpPostFile, is OLE Automation-enabled and very similar to WpPost. The major difference is that WpPostFile can only post one file or directory at a time.

WpBindToSite

```
LONG WpBindToSite(
    IN HWND    hwnd,
    IN LPTSTR  lpszSiteName,
    IN LPTSTR  lpszURL OPTIONAL,
    IN DWORD   fdwFlags,
    IN DWORD   dwReserved,
    OUT LPVOID *ppbObj
);
```

Retrieves the address of a WebPost service provider object that can post web pages to the specified site name or URL. The application can then call the provider functions directly. The function works only if the site exists.

- Returns ERROR_SUCCESS if successful or an error value otherwise.

hwnd
Handle of the window to which the focus returns when the wizard, if invoked, completes. This parameter can be NULL if the wizard is not invoked or if this function is called from a console application.

lpszSiteName
Address of a null-terminated string that contains the site name.

lpszURL
Address of a null-terminated string that contains a URL.

fdwFlags
Action flags. Reserved for future use.

dwReserved
Reserved, and must be set to 0.

ppbObj
Address of a WebPost service provider interface.

WpDeleteSite

```
LONG WpDeleteSite(
     IN LPTSTR lpszSiteName
);
```

Deletes a site name that has been configured.

- Returns ERROR_SUCCESS if successful or an error value otherwise.

lpszSiteName
Address of a null-terminated string that contains the site name.

WpListSites

```
LONG WpListSites(
    IN OUT LPDWORD   lpcbSites,
    OUT LPWPSITEINFO lpbSites,
    OUT LPDWORD      lpcSites
);
```

Retrieves information about the Internet sites that are already configured.

- Returns ERROR_SUCCESS if successful or an error value otherwise.

lpcbSites

Address of a variable that contains the size, in bytes, of the buffer pointed to by the *lpbSites* parameter. When this function returns, the variable contains the number of bytes in the *lpbSites* buffer. If the *lpbSites* value is NULL, the variable receives the size of the buffer required to contain all of the site structures.

lpbSites

Address of a buffer that receives an array of WPSITEINFO structures that contain the site information.

lpcSites

Address of a variable that receives the number of structures returned in the *lpbSites* array. If the *lpbSites* value is NULL, the variable receives the total number of sites.

WpPost

```
LONG WpPost(
    IN HWND        hwnd  OPTIONAL,
    IN DWORD       cLocalPaths,
    IN LPTSTR      *lppszLocalPaths,
    IN OUT LPDWORD lpcbSiteName,
    IN OUT LPTSTR  lpszSiteName OPTIONAL,
    IN OUT LPDWORD lpcbURL,
    IN OUT LPTSTR  lpszURL OPTIONAL,
    IN DWORD       fdwFlags
);
```

Posts a list of files or directories to an Internet site identified by the given site name or URL. If the site name and URL are NULL, the function invokes a wizard to let the user choose an existing or create a new Internet site.

- Returns ERROR_SUCCESS if successful or an error value otherwise.

hwnd

Handle of the window to which the focus returns when the wizard, if invoked, completes. This parameter can be NULL if the wizard is not invoked or if this function is called from a console application.

cLocalPaths

Number of elements in the array specified by the *lppszLocalPaths* parameter.

lppszLocalPaths

Address of an array of null-terminated strings that contain the file names or directories to be posted on the Internet. If any of these strings point to a directory (and the **WPF_NO_RECURSIVE_POST** flag is not set in the *fdwFlags*parameter), all the files in that directory are posted.

lpcbSiteName

Address of a variable that contains the length, in bytes, of the buffer specified by the *lpszSiteName* parameter.

lpszSiteName

Address of a null-terminated string that contains the friendly Internet site name. If this parameter and the *lpszURL* parameter are NULL, the function invokes the wizard (unless **WPF_NO_WIZARD** is set in *fdwFlags*) to let the user choose or create a site name.

If this parameter is not null, on return it will contain the site name that the files were posted to.

lpcbURL

Address of a variable that receives the length, in bytes, of the buffer specified by *lpszURL*.

lpszURL

Address of a null-terminated string that contains the destination URL. If this parameter is NULL, the files are posted in the root URL for *lpszSiteName*.

If this parameter is not NULL, the URL of the file copied, or the URL of the directory that the files were copied to, are returned in the buffer pointed to by this parameter.

fdwFlags

Array of action flags. The following values can be combined:

Value	Meaning
WPF_FIRST_FILE_AS_DEFAULT	Take the first file specified in the *lppszLocalPaths* list as the file that will be shown as the default page.
WPF_MINIMAL_UI	Skip the pages where the input has been provided. For example, if *lpszSiteName* is specified, the wizard will not show the page for choosing the site name.
WPF_NO_RECURSIVE_POST	If any element in the *lppszLocalPaths* array points to a directory, do not post files recursively.

WPF_NO_WIZARD Do not prompt the user for any input. This is relevant only if *lpszSiteName* has been created before.

Suppose you have created a site named "My Web Site" that has a URL of "http://www.isp.com/~username", and you want to post a file named "abcd.tmp" to "http://www.isp.com/~username/personal" as person.htm. The parameters for **WpPost** should be as follows:

```
lpszSiteName = "My Web Site"
cLocalPaths = 1
lppszLocalPaths = {"c:\tmp\abcd.tmp"}
lpszURL = "http://www.isp.com/~username/personal/person.htm"
```

WpPostFile

```
HRESULT WpPostFile(
    [in] long hwnd,
    [in] BSTR lpszLocalPaths,
    [in, out]long * lpcbSiteName,
    [in, out]BSTR * lpszSiteName,
    [in, out]long * lpcbURL,
    [in, out]BSTR * lpszURL,
    [in] long   fdwFlag,
    [out] long *lpRetCode
);
```

This function is for use with OLE Automation. It posts a file or directory to an Internet site identified by the given site name or URL. If the site name and URL are NULL, the function invokes a wizard to let the user choose an existing or create a new Internet site.

- Returns ERROR_SUCCESS if successful or an error value otherwise.

hwnd

Handle of the window to which the focus returns when the wizard, if invoked, completes. This parameter can be NULL if the wizard is not invoked or if this function is called from a console application.

lpszLocalPaths

Address of a null-terminated string that contains the file name or directory to be posted on the Internet.

lpcbSiteName

Address of a variable that contains the length, in bytes, of the buffer specified by the *lpszSiteName* parameter.

lpszSiteName

Address of a null-terminated string that contains the friendly Internet site name. If this parameter and the *lpszURL* parameter are NULL, the function invokes the wizard (unless **WPF_NO_WIZARD** is set in *fdwFlags*) to let the user choose or create a site name.

If this parameter is not NULL, on return it will contain the site name that the files were posted to.

lpcbURL

Address of a variable that receives the length, in bytes, of the buffer specified by *lpszURL*.

lpszURL

Address of a null-terminated string that contains the destination URL. If this parameter is NULL, the files are posted in the root URL for *lpszSiteName*.

If this parameter is not NULL, the URL of the file copied or the URL of the directory that the files were copied to, return in the buffer pointed to by this parameter.

fdwFlags

Array of action flags. The following values can be combined:

Value	Meaning
WPF_FIRST_FILE_AS_DEFAULT	Take the first file specified in the *lpszLocalPaths* as the file that will be shown as the default page.
WPF_MINIMAL_UI	Skip the pages where input has been provided. For example, when this flag is set and *lpszSiteName* is specified, the wizard will not show the page for choosing the site name.
WPF_NO_RECURSIVE_POST	If *lpszLocalPaths* points to a directory, do not post files recursively.
WPF_NO_WIZARD	Do not prompt the user for any input. This is relevant only if *lpszSiteName* points to a site that has been created before.

lpRetCode

Address of a variable that receives the function's return code.

This function can post only one file or directory at a time.

Here are two examples that show how to use this function with Visual Basic.

Example 1:

```
Dim wpo as New WPObj.WPObj
wpo.WpPostFile  <etc.>
```

Example 2:

```
Dim wpo as Object
Set wpo = CreateObject("WPObj.Application")
wpo.WpPostFile  <etc.>
```

WPSITEINFO

```
typedef struct tagWPSITEINFO {
    DWORD  dwSize;
    DWORD  fdwFlags;
    LPTSTR lpszSiteName;
    LPTSTR lpszSiteURL;
} WPSITEINFO, *LPWPSITEINFO;
```

Contains information about a site on the World Wide Web.

dwSize

Size, in bytes, of the structure.

fdwFlags

Array of flags that indicate the state of the site and provide information about the site. The following values can be combined:

Value	**Meaning**
WPSF_NEEDS_COMMIT	Posting occurs during the Commit function.
WPSF_CAN_BROWSE_DIR	Supports browsing directories.
WPSF_CONNECTED_TO_NETWORK	Connected to the network.
WPSF_LOGGED_IN_TO_SERVER	Logged in to the server.

lpszSiteName

Address of a null-terminated string that specifies a friendly name for the URL pointed to by *lpszSiteURL*.

lpszSiteURL

Address of a null-terminated string that specifies the root URL for this site.

WebPost SPI Functions

WebPost service provider DLLs are OLE COM servers that do the actual work of posting files to Internet sites. They provide the functions described in this section. The WebPost functions, and the wizard, route the request to one of the providers based on the site name or URL and use the provider functions to post the files.

Each provider DLL exports the WppBindToSite function, which takes a site name or URL and an interface identifier, and returns a pointer to the interface. The WpBindToSite function uses the WppBindToSite function to obtain the address of an interface, and then passes the address back to the caller. Each provider also exports the WppListSites function, which lists all the Internet sites that the provider can post to.

WppBindToSite

```
LONG WppBindToSite(
    IN HWND hwnd,
    IN LPCTSTR lpszSiteName OPTIONAL,
    IN LPCTSTR lpszURL OPTIONAL,
    IN REFIID riid,
    IN DWORD fdwFlags,
    IN DWORD dwReserved,
    OUT PVOID *ppvObj
);
```

Retrieves the address of an interface if the provider DLL owns the site name or the URL.

- Returns ERROR_SUCCESS if successful or an error value otherwise.

hwnd
Handle of the window to which the focus returns when the wizard, if invoked, completes. This parameter can be NULL if the wizard is not invoked or if this function is called from a console application.

lpszSiteName
Address of a null-terminated string that contains an Internet site name.

lpszURL
Address of a null-terminated string that contains a URL.

riid
Interface identifier. All providers should support the IID_IWPSite interface identifier.

fdwFlags
Action flags. Can be a combination of these values:

Value	**Meaning**
WPF_FORCE_BIND	The provider should return successfully for the fucntion no matter what the *lpszURL* value is.

dwReserved
Reserved, and must be set to 0.

ppvObj
Address of a variable to receive the interface address. For more information about the interfaces, see WebPost SPI Interface Functions.

WppDeleteSite

```
LONG WppDeleteSite(
    IN LPTSTR lpszSiteName
);
```

Deletes a site name that has been configured.

- Returns ERROR_SUCCESS if successful or an error value otherwise.

lpszSiteName
Address of a null-terminated string that contains the site name.

WppListSites

```
LONG WppListSites(
    IN OUT LPDWORD   lpcbSites,
    OUT LPWPSITEINFO lpbSites,
    OUT LPDWORD      lpcSites
);
```

Retrieves information about the Internet sites managed by this provider. The WpListSites function calls this function for each provider, aggregates the results, and returns them to the caller.

- Returns ERROR_SUCCESS if successful or an error value otherwise.

lpcbSites
Address of a variable that receives the number of bytes copied to the buffer pointed to by the *lpbSites* parameter. If *lpbSites* is NULL, the variable receives the required size, in bytes, of the buffer.

lpbSites
Address of a buffer that receives an array of WPSITEINFO structures that contain the site information

lpcSites
Address of a variable that receives the number of structures copied to the *lpbSites* buffer. If *lpbSites* is NULL, the variable receives the total number of sites.

WebPost SPI Interface Functions

A WebPost SPI provides the interface functions described in this section. There should be an ANSI version and a Unicode version of each interface. Please refer to the Wpspi.h file for details.

AddWizardPages

```
AddWizardPages(
    IN LPVOID               lpv,
    IN LPFNADDPROPSHEETPAGE lpfnAdd,
    IN OUT LPARAM           lParam
);
```

Allows the provider DLLs to plug pages into the wizard invoked by the WpPost function.

lpv

Not currently used.

lpfnAdd

Address of a function that this WebPost service provider should use to add wizard pages.

lParam

Address of a page identifier, an unsigned integer.

As input to this function, the low-order word is the dialog identifier of the next page of the last page of the provider, and the high-order word is the dialog identifier of the previous page of the first page of the provider. If the high-order word is 0, the provider's first page is the first page of the wizard (for the WPF_MINIMAL_UI case). The **Back** button should be disabled using **PropSheet_SetWizButtons**. If the low-order word is 0, the provider's last page is the last page of the wizard. The **Next** button should be changed to the **Finish** button using **PropSheet_SetWizButtons**.

As output from this function, the low-order word is the dialog identifier of the provider's last page, and the high-order word is the dialog identifier of the provider's first page. If the provider has no pages, they should return 0 as the page identifier.

As an example, in setting up a new site the wizard would provide the friendly site name and the site's URL and the provider DLLs would handle the rest of the site set up. The dialog identifier of the provider wizard page should be a value between the value of IDD_WEBPOST_PROVIDER_FIRST and IDD_WEBPOST_PROVIDER_LAST.

Commit

```
Commit(void);
```

Ensures that all the files posted to this server with the PostFiles function are actually written to the Internet server.

DeleteFile

```
DeleteFile(IN LPCTSTR lpszFile);
```

Deletes the given file from the destination site.

lpszFile
Address of a null-terminated string that contains the name of the file to delete.

FindClose

```
FindClose(IN HANDLE hSearchHandle);
```

Closes the specified search handle.

hSearchHandle
Search handle returned by a previous call to the FindFirstFile function.

The FindFirstFile and FindNextFile functions use the search handle to locate files with names that match a given name.

FindFirstFile

```
FindFirstFile(
    IN LPCTSTR            lpszSearchFile,
    OUT LPWIN32_FIND_DATA lpFindFileData,
    OUT LPHANDLE          lpSearchHandle
);
```

Searches a directory for a file whose name matches the specified file name on the destination site identified by this object. It examines subdirectory names as well as file names.

lpszSearchFile
Address of a null-terminated string that contains the file name to find.

lpFindFileData
Address of a **WIN32_FIND_DATA** structure that receives information about the file or subdirectory that has been found.

lpSearchHandle
Address of a variable that receives a handle that can be used for subsequent calls to the FindNextFile and FindClose functions.

FindNextFile

```
FindNextFile(
    IN HANDLE hSearchHandle,
    OUT LPWIN32_FIND_DATA lpFindFileData,
);
```

Continues a file search from a previous call to the FindFirstFile function.

hSearchHandle
Search handle returned by a previous call to the FindFirstFile function.

lpFindFileData
Address of a **WIN32_FIND_DATA** structure that receives information about the file or subdirectory that has been found.

GetError

```
GetError(
    OUT LPDWORD    lpdwErrorType,
    OUT LPDWORD    lpdwErrorCode,
    IN OUT LPDWORD lpcbError,
    OUT LPTSTR     lpszError
);
```

Retrieves additional information about an error.

lpdwErrorType
Indicates the error type. To be defined.

lpdwErrorCode
Indicates the error code.

lpcbError
Indicates the size, in bytes, of the *lpszError* buffer.

lpszError
Address of a buffer that receives the error information.

GetParam

```
GetParam(
    IN LPCTSTR     lpszParameter,
    IN OUT LPDWORD lpcbValue,
    OUT LPTSTR     lpszValue
);
```

Retrieves the configuration parameters for this site.

lpszParameter

Address of a null-terminated string that contains the configuration parameter name, such as **WPCP_HOMEPAGE_URL**. For more information about configuration parameters, see Configuration Parameters.

lpcbValue

Address of a variable that receives the size of the *lpszValue* buffer. If *lpszValue* is NULL, or if the buffer length is short, this variable receives the size, in bytes, of the parameter name.

lpszValue

Address of a buffer that receives the value of the parameter queried.

GetSiteInfo

```
GetSiteInfo(
    OUT LPWPSITEINFO lpbSite,
    IN OUT LPDWORD   lpcbSite
);
```

Retrieves the site information for the current object.

lpbSite
Address of a buffer that receives a WPSITEINFO structure containing the site information.

lpcbSite
Address of a variable that contains the size, in bytes, of the *lpbSite* buffer.

NetworkConnect

```
NetworkConnect(
    IN LPCTSTR lpszUserName,
    IN LPCTSTR lpszPassword
);
```

Connects to the Internet. If the site uses a dial-up connection, this function starts the dial-up connection (configured during the site setup).

lpszUserName

Address of a null-terminated string that contains the user name for the dial-up connection. If this parameter is NULL, the user name created during the setup of this site is reused.

lpszPassword

Address of a null-terminated string that contains the password for the dial-up connection. If this parameter is NULL, the behavior depends on the provider.

NetworkDisconnect

```
NetworkDisconnect(void);
```

Disconnects from the Internet (hang-up if a dial-up connection was used).

PostFiles

```
PostFiles(
    IN DWORD cLocalPaths,
    IN LPTSTR *lppszLocalPaths,
    IN OUT LPDWORD lpcbURL,
    IN OUT LPTSTR lpszURL OPTIONAL,
    IN DWORD fdwFlags
);
```

Posts files to the specified URL on the destination site identified by this object.

cLocalPaths

Number of elements in the *lppszLocalPaths* array.

lppszLocalPaths

Address of an array of null-terminated strings that contain the file names or directories to be posted on the Internet. If any of these strings point to a directory and the **WPF_NO_RECURSIVE_POST** flag is not set in *fdwFlags*, all the files in that directory are posted.

lpcbURL

Address of a variable that indicates the length of the *lpszURL* buffer. When the function returns, the variable contains the length, in bytes, of the that buffer.

lpszURL

Address of a null-terminated string that contains the destination URL. If this parameter is NULL, the files are posted in the root URL for the site name specified by *lpszSiteName*.

If this parameter is not NULL, the URL of the copied file, or the URL of the directory that the files were copied to, is returned in the buffer pointed to by this parameter.

fdwFlags

Action flags. The following values can be combined:

Value	**Meaning**
0	Post files, invoking the wizard if necessary.
WPF_FIRST_FILE_AS_DEFAULT	Take the first file specified in the *lppszLocalPaths* list as the file that will be shown as the default page.
WPF_MINIMAL_UI	Skip the introduction page in the Wizard.
WPF_NO_RECURSIVE_POST	If any element in the *lppszLocalPaths* array points to a directory, do not post files recursively.
WPF_NO_WIZARD	Do not prompt the user for any input.

See the description of WpPost for more information.

ServerLogin

```
ServerLogin(
    IN LPCTSTR lpszUserName,
    IN LPCTSTR lpszPassword
);
```

Logs the user on to the Internet server. The given user name and password are for the Internet server. They may or may not be the same as those used for the NetworkConnect function.

lpszUserName

Address of a null-terminated string that contains the user name for the Internet server. If this parameter is NULL, the user name created during the setup of this site is reused.

lpszPassword

Address of a null-terminated string that contains the password for the Internet server. If this parameter is NULL, the behavior depends on the provider.

ServerLogout

```
ServerLogout(void);
```

Logs the user out of the Internet server.

SetParam

```
SetParam(
    IN LPCTSTR lpszParameter,
    IN LPCTSTR lpszValue
);
```

Sets a configuration parameter for a given site.

lpszParameter

Address of a null-terminated string that contains a configuration parameter name, such as **WPCP_DEFAULT_URL**. For more information about configuration parameters, see Configuration Parameters.

lpszValue

Address of a null-terminated string that contains the configuration parameter value.

Configuration Parameters

Every WebPost service provider must support the following parameters:

```
#define WPCP_HOMEPAGE_URL "HomepageURL"
```

A query of this parameter should return the home page for the given site. For example, for www.provider.com running NCSA's httpd server, the home page will usually be *"http://www.provider.com/~username/index.htm"*.

Here are the parameters supported by the default service provider (Defwpp.dll):

```
#define WPCP_BASEURL    "BaseURL"
```

A query of this parameter should return the root URL for the given user. For example, for www.provider.com running NCSA's httpd server, it will usually be "http://www.provider.com/~username".

```
#define WPCP_BASEPATH    "BasePath"
```

A query of this parameter should return corresponding file path for the above WPCP_BASEURL parameter. For example, for www.provider.com running NCSA's httpd server,it will usually be "public_html".

More parameters are to be defined at a later date.

Beta Release Notes

The WebPost SDK files include Wpapi.h, Wpspi.h, Wpguid.h, Webpost.lib, and this document, Werbpost.doc. Wpobj.h and Wpobj.tlb are for OLE Automation programmers.

The WebPost executable files in the SDK CD include Webpost.dll, Defwpp.dll, and Wpwiz.exe. The Wpwiz.exe file is a simple utility that calls the WpPost function. The Webpost.dll file provides the WebPost API functions. The Defwpp.dll file provides the Service Provider Interface (SPI) for the default provider that can post files to NCSA's httpd server. By default, it posts files using the *ftp* protocol to the *public_html* subdirectory under the user's home directory and verifies them with the *http* protocol using the *wininet* interface. It also lets the user identify a dial-up connection to the web site and can connect to the Internet server over this dial-up connection.