

# GetTempFileName function

Creates a name for a temporary file. If a unique file name is generated, an empty file is created and the handle to it is released; otherwise, only a file name is generated.

## Syntax

**C++**

```
UINT WINAPI GetTempFileName(  
    _In_   LPCTSTR lpPathName,  
    _In_   LPCTSTR lpPrefixString,  
    _In_   UINT     uUnique,  
    _Out_  LPTSTR  lpTempFileName  
);
```

## Parameters

*lpPathName* [in]

The directory path for the file name. Applications typically specify a period (.) for the current directory or the result of the [GetTempPath](#) function. The string cannot be longer than **MAX\_PATH**–14 characters or **GetTempFileName** will fail. If this parameter is **NULL**, the function fails.

*lpPrefixString* [in]

The null-terminated prefix string. The function uses up to the first three characters of this string as the prefix of the file name. This string must consist of characters in the OEM-defined character set.

*uUnique* [in]

An unsigned integer to be used in creating the temporary file name. For more information, see Remarks.

If *uUnique* is zero, the function attempts to form a unique file name using the current system time. If the file already exists, the number is increased by one and the functions tests if this file already exists. This continues until a unique filename is found; the function creates a file by that name and closes it. Note that the function does not attempt to verify the uniqueness of the file name when *uUnique* is nonzero.

*lpTempFileName* [out]

A pointer to the buffer that receives the temporary file name. This buffer should be **MAX\_PATH** characters to accommodate the path plus the terminating null character.

## Return value

If the function succeeds, the return value specifies the unique numeric value used in the temporary file name. If the *uUnique* parameter is nonzero, the return value specifies that same number.

If the function fails, the return value is zero. To get extended error information, call [GetLastError](#).

The following is a possible return value.

Return value	Description
<b>ERROR_BUFFER_OVERFLOW</b>	The length of the string pointed to by the <i>lpPathName</i> parameter is more than <b>MAX_PATH</b> –14 characters.

## Remarks

The **GetTempFileName** function creates a temporary file name of the following form:

<path>\<pre><uuuu>.TMP

The following table describes the file name syntax.

Component	Meaning
<path>	Path specified by the <i>lpPathName</i> parameter
<pre>	First three letters of the <i>lpPrefixString</i> string
<uuuu>	Hexadecimal value of <i>uUnique</i>

If *uUnique* is zero, **GetTempFileName** creates an empty file and closes it. If *uUnique* is not zero, you must create the file yourself. Only a file name is created, because **GetTempFileName** is not able to guarantee that the file name is unique.

Only the lower 16 bits of the *uUnique* parameter are used. This limits **GetTempFileName** to a maximum of 65,535 unique file names if the *lpPathName* and *lpPrefixString* parameters remain the same.

Due to the algorithm used to generate file names, **GetTempFileName** can perform poorly when creating a large number of files with the same prefix. In such cases, it is recommended that you construct unique file names

based on **GUIDs**.

Temporary files whose names have been created by this function are not automatically deleted. To delete these files call [DeleteFile](#).

To avoid problems resulting when converting an ANSI string, an application should call the [CreateFile](#) function to create a temporary file.

In Windows 8 and Windows Server 2012, this function is supported by the following technologies.

Technology	Supported
Server Message Block (SMB) 3.0 protocol	Yes
SMB 3.0 Transparent Failover (TFO)	Yes
SMB 3.0 with Scale-out File Shares (SO)	Yes
Cluster Shared Volume File System (CsvFS)	Yes
Resilient File System (ReFS)	Yes

## Examples

For an example, see [Creating and Using a Temporary File](#).

## Requirements

<b>Minimum supported client</b>	Windows XP [desktop apps   Windows Store apps]
<b>Minimum supported server</b>	Windows Server 2003 [desktop apps   Windows Store apps]
<b>Header</b>	FileAPI.h (include Windows.h); WinBase.h on Windows Server 2008 R2, Windows 7, Windows Server 2008, Windows Vista, Windows Server 2003, and Windows XP (include Windows.h)

<b>Library</b>	Kernel32.lib
<b>DLL</b>	Kernel32.dll
<b>Unicode and ANSI names</b>	<b>GetTempFileNameW</b> (Unicode) and <b>GetTempFileNameA</b> (ANSI)

## See also

[File Management Functions](#)

[Naming Files, Paths, and Namespaces](#)

[CreateFile](#)

[DeleteFile](#)

[GetTempPath](#)