

## Creating a Name for Temporary Files

- [Alyce](#)

[Creating a Name for Temporary Files](#) | [Writing Temporary Files](#) | [GetTempPathA](#) | [GetTempFileNameA](#) | [Demo](#)

---

## Writing Temporary Files

Security measures in Windows prevents programs from writing files to all areas. Use the Windows API to retrieve the user's temporary directory and to create a temporary filename that is unlikely to be used by another application.

### GetTempPathA

To insure that your program can successfully write a temporary file on the user's computer, use the function [GetTempPathA](#) to retrieve the path to a directory where temporary files should be created.

### GetTempFileNameA

The **GetTempFileNameA** function allows you to create a unique temporary filename. The name is in this format:

TemporaryPath\$ + ThreeCharPrefix + HexadecimalString\$ + ".TMP"

An example that uses GetTempPathA for the temporary path and "xyz" for the prefix might look like this:

C:\DOCUME~1\MYNAME~1\LOCALS~1\Temp\xyzDA7.tmp

The function looks like this:

```
call dll #kernel32, "GetTempFileNameA", _  
    TempPath$ as ptr, _    'directory for temp file  
    prefix$ as ptr, _      'desired prefix for temp filename  
    unique as ulong, _     '0=file created, nonzero=you must create file  
    TempFile$ as ptr, _  
'string buffer to hold qualified path and filename  
    result as ulong        'nonzero=success
```

The TempPath\$ string can contain any directory. It can be the DefaultDir\$, for instance. Best practice dictates that you use the [GetTempPathA](#) function to retrieve the user's temporary directory.

The prefix\$ argument should be one to three characters long. Some examples:

```
abc
rgf
aa
b
xyz
zy
```

The "unique" argument tells the function whether to create the file, or to simply return the filename. If it is 0, the function creates the file, thus assuring its existence. If the value for unique is nonzero, the programmer must create the file himself.

TempFile\$ should be a string buffer of blank spaces that will receive the temporary filename returned by the function. This is a fully qualified path and filename, as in the example above.

Best practice dictates that you should delete your temporary file when your program closes, or when it is no longer needed. Use the **KILL** statement to do this.

## Demo

This demo program retrieves the path to the temporary directory. It created a prefix of "xyz". It creates a string variable full of blank spaces that will hold the fully qualified path and filename to the temporary file when the function returns.

The demo does not write anything to the file, nor does it delete the file on exit. You will do both of these things when you include the code in your program.

```
TempPath$=GetTempPath$( )
prefix$="xyz"          'up to 3 characters for desired prefix
TempFile$ = space$(256)+chr$(0)

callDll #kernel32, "GetTempFileNameA", _
    TempPath$ as ptr, _  'directory for temp file
    prefix$ as ptr, _    'desired prefix for temp filename
    0 as ulong, _        '0=file created, nonzero=you must create file
    TempFile$ as ptr, _
'string buffer to hold qualified path and filename
    result as ulong      'nonzero=success
```

```
'TempFile$ holds complete path and filename info
print TempFile$
end
```

```
Function GetTempPath$( )
    CallDLL #kernel32, "GetTempPathA", _
    0 as long, _
    _NULL as long, _
    length as long

    buf$ = space$(length)

    CallDLL #kernel32, "GetTempPathA", _
    length as long, _
    buf$ as ptr, _
    ret as long

    GetTempPath$ = buf$
End Function
```

---

See also: [GetTempPathA](#)

---

[Creating a Name for Temporary Files](#) | [Writing Temporary Files](#) | [GetTempPathA](#) | [GetTempFileNameA](#) | [Demo](#)