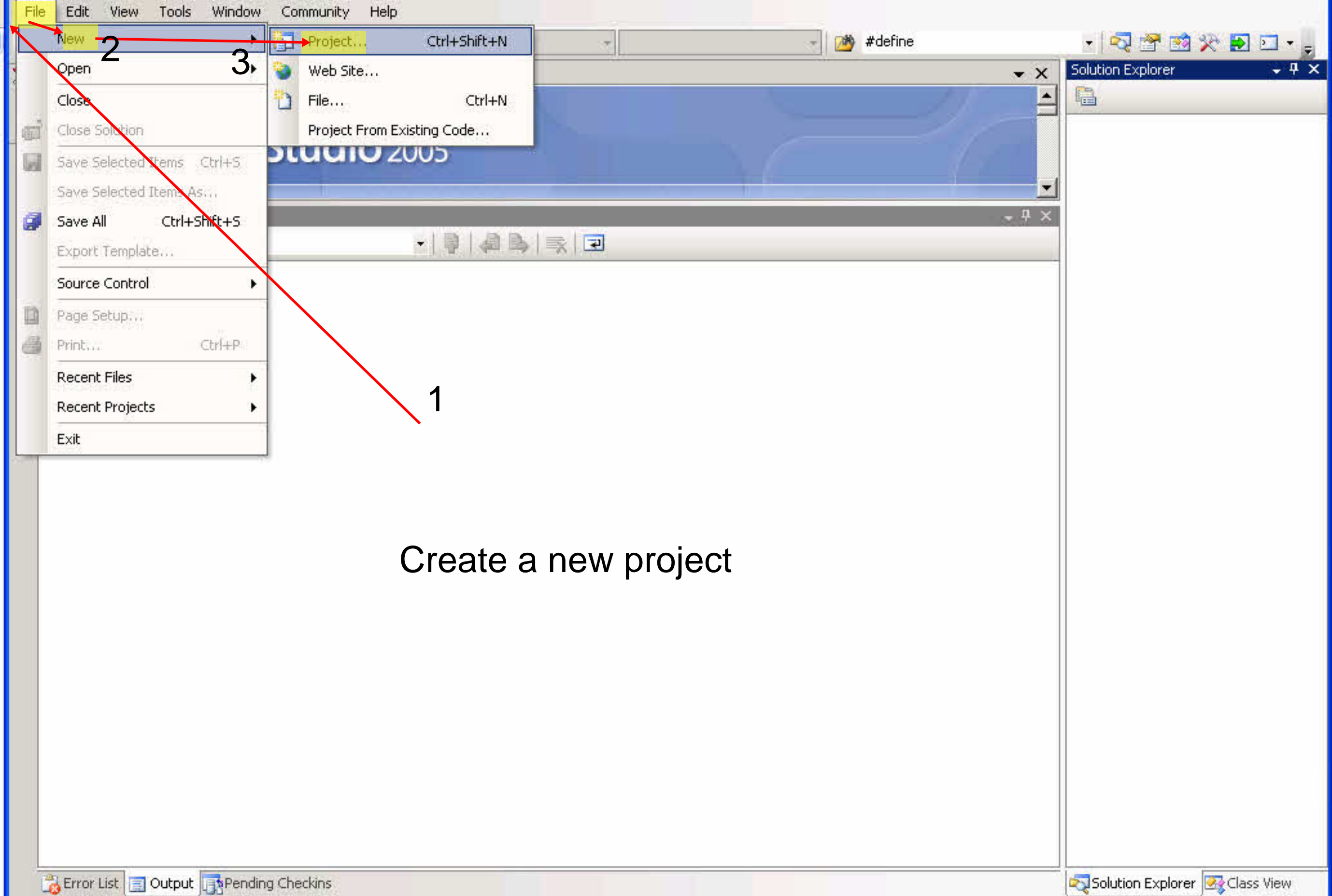


Start Microsoft Visual Studio



Create a new project

Microsoft Visual Studio

Recent Projects

- DDD1
- DDDMFC1
- ddd
- D3DX8BBC
- WindowsApplication2
- WindowsApplication1

Getting Started

What's new in Visual Studio 2005?

Output

Show output from:

New Project

Project types:

- Visual Basic
 - Windows
 - Smart Device
 - Database
 - Starter Kits
- Visual C#
- Visual J#
- Visual C++** (1)
 - ATL
 - CLR
 - General
 - MFC
 - Smart Device
 - Win32** (2)
- Other Project Types

Templates:

Visual Studio installed templates

- Win32 Console Application
- Win32 Project** (3)

My Templates

- Search Online Templates...

A project for creating a Win32 application, console application, DLL, or static library

Name: **MakeTempFilenameDLL2005** (4)

Location: **Z:\proj** (5)

Solution Name: **MakeTempFilenameDLL2005**

Create directory for solution

Add to Source Control

Create a DLL



Recent Projects

- DDD1
- DDDMFC1
- ddd
- D3DX8BBC
- WindowsApplication2
- WindowsApplication1

Open: Project... | Web Site...
Create: Project... | Web Site...

Getting Started

What's new in Visual Studio 2005?

Win32 Application Wizard - MakeTempFilenameDLL2005

Welcome to the Win32 Application Wizard

Overview
Application Settings

These are the current project settings:

- Windows application

Click **Finish** from any window to accept the current settings.

After you create the project, see the project's readme.txt file for information about the project features and files that are generated.

< Previous **Next >** Finish Cancel





Recent Projects

- DDD1
- DDDMFC1
- ddd
- D3DX8BBC
- WindowsApplication2
- WindowsApplication1

Open: [Project...](#) | [Web Site...](#)
Create: [Project...](#) | [Web Site...](#)

Getting Started

What's new in Visual Studio 2005?

Win32 Application Wizard - MakeTempFilenameDLL2005

Application Settings

Overview
Application Settings

Application type:

- Windows application
- Console application
- DLL**
- Static library

Additional options:

- Empty project
- Export symbols
- Precompiled header

Add common header files for:

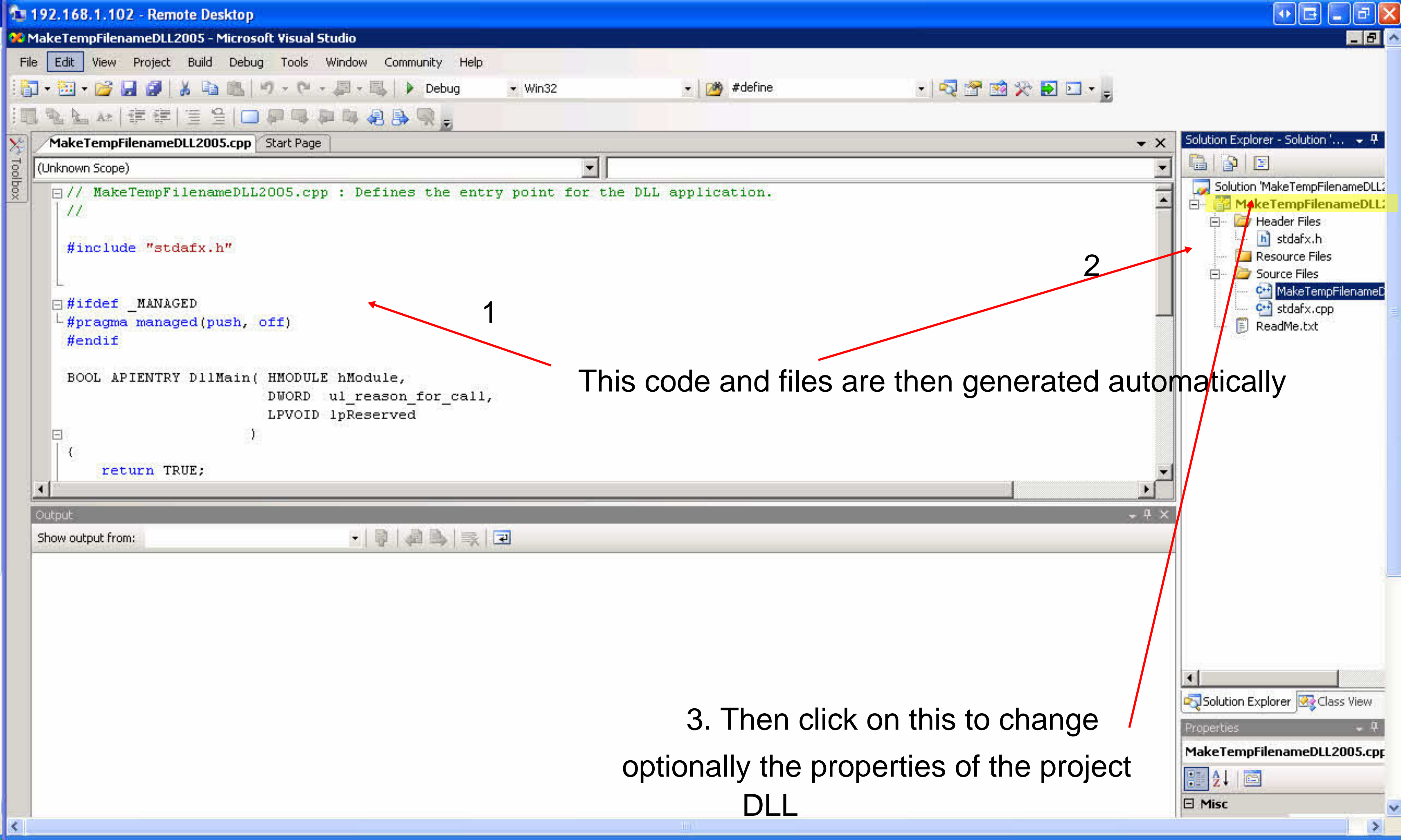
- ATL
- MFC

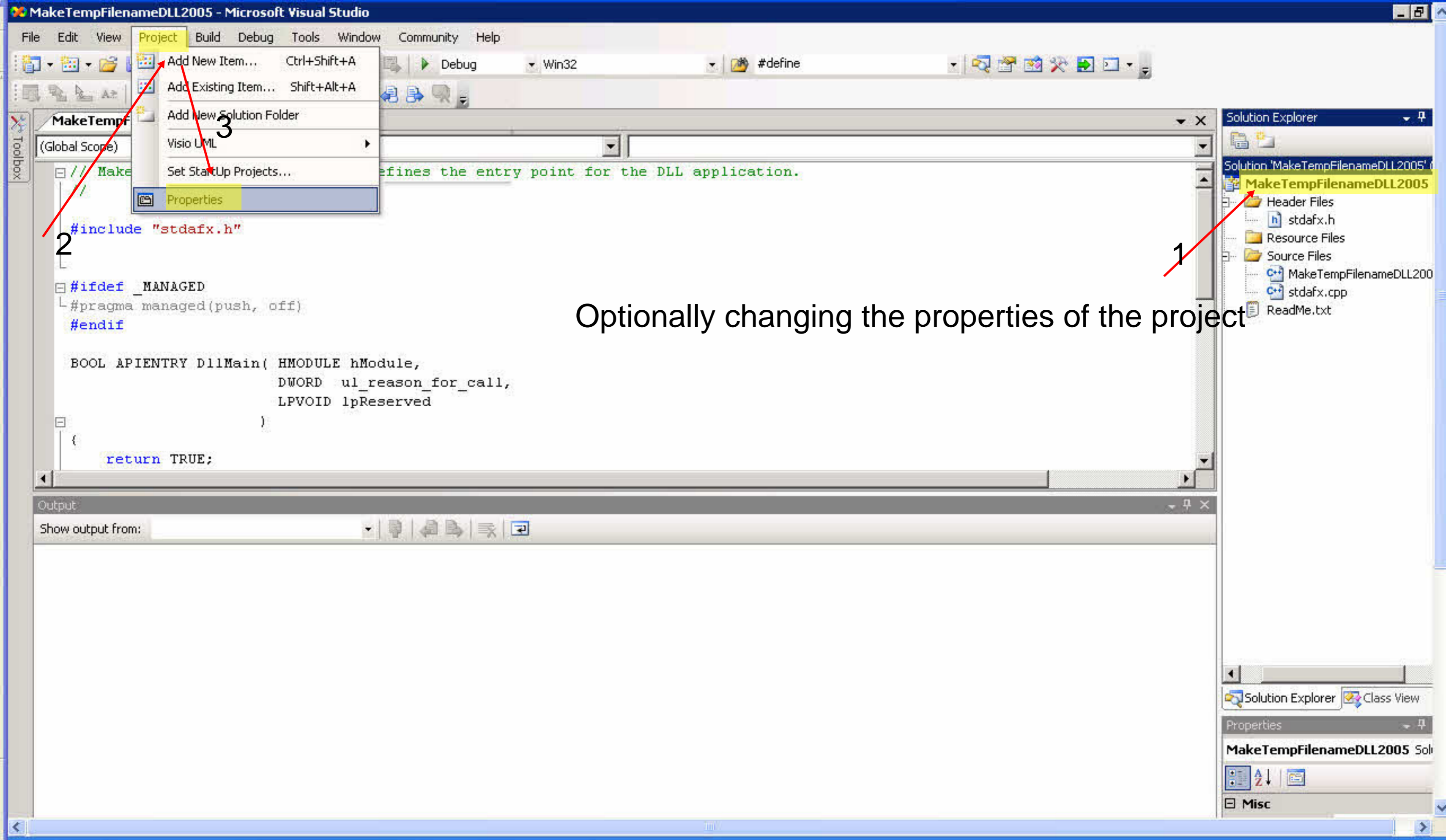
< Previous Next > **Finish** Cancel

1

2

Choose DLL





Optionally changing the properties of the project

```

MakeTempFilenameDLL2005.cpp Start P
(Global Scope)
// MakeTempFilenameDLL2005.
//
#include "stdafx.h"

#ifdef _MANAGED
#pragma managed(push, off)
#endif

BOOL APIENTRY DllMain( HMODULE
                    hModule,
                    DWORD
                    dwReason,
                    LPVOID
                    lpReserved)
{
    return TRUE;
}

```

MakeTempFilenameDLL2005 Property Pages

Configuration: Active(Debug) Platform: Active(Win32) Configuration Manager...

- Common Properties
 - References
- Configuration Properties
 - General**
 - Debugging
 - C/C++
 - Linker
 - Manifest Tool
 - XML Document Generator
 - Browse Information
 - Build Events
 - Custom Build Step
 - Web Deployment

General	
Output Directory	\$(SolutionDir)\$(ConfigurationName)
Intermediate Directory	\$(ConfigurationName)
Extensions to Delete on Clean	*.obj;*.ilk;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.pgd;\$(TargetDir)
Build Log File	\$(IntDir)\BuildLog.htm
Inherited Project Property Sheets	
Project Defaults	
Configuration Type	Dynamic Library (.dll)
Use of MFC	Use Standard Windows Libraries
Use of ATL	Not Using ATL
Minimize CRT Use in ATL	No
Character Set	Use Unicode Character Set
Common Language Runtime support	No Common Language Runtime support
Whole Program Optimization	No Whole Program Optimization

Output Directory
Specifies a relative path to the output file directory; can include environment variables.

OK Cancel Apply

Solution Explorer - MakeTemp...

- Solution 'MakeTempFilenameDLL2005' (1 project)
 - MakeTempFilenameDLL2005**
 - Header Files
 - stdafx.h
 - Resource Files
 - Source Files
 - MakeTempFilenameDLL2005.c
 - stdafx.cpp
 - ReadMe.txt

Solution Explorer Class View

Properties

MakeTempFilenameDLL2005 Pro

Misc

```
MakeTempFilenameDLL2005.cpp Start P  
(Global Scope)  
// MakeTempFilenameDLL2005.  
//  
#include "stdafx.h"  
  
#ifdef _MANAGED  
#pragma managed(push, off)  
#endif  
  
BOOL WINAPI DllMain( HMODULE  
                    DWORD  
                    LPVOID  
                    )  
{  
    return TRUE;  
}
```

MakeTempFilenameDLL2005 Property Pages

Configuration: Active(Debug) Platform: Active(Win32) Configuration Manager...

- Common Properties
- References
- Configuration Properties
 - General
 - Debugging
 - C/C++
 - General
 - Optimization
 - Preprocessor
 - Code Generation
 - Language
 - Precompiled Headers
 - Output Files
 - Browse Information
 - Advanced
 - Command Line
 - Linker
 - Manifest Tool
 - XML Document Generator
 - Browse Information
 - Build Events
 - Custom Build Step
 - Web Deployment

Enable String Pooling	No
Enable Minimal Rebuild	Yes (/Gm)
Enable C++ Exceptions	Yes (/EHsc)
Smaller Type Check	No
Basic Runtime Checks	Both (/RTC1, equiv. to /RTCsu)
Runtime Library	Multi-threaded Debug (/MTd)
Struct Member Alignment	Default
Buffer Security Check	Yes
Enable Function-Level Linking	No
Enable Enhanced Instruction Set	Not Set
Floating Point Model	Precise (/fp:precise)
Enable Floating Point Exceptions	No

Runtime Library
Specify runtime library for linking. (/MT, /MTd, /MD, /MDd)

OK Cancel Apply

Solution Explorer - MakeTemp...

- Solution 'MakeTempFilenameDLL2005' (1 p...)
 - MakeTempFilenameDLL2005
 - Header Files
 - stdafx.h
 - Resource Files
 - Source Files
 - MakeTempFilenameDLL2005.c...
 - stdafx.cpp
 - ReadMe.txt

Solution Explorer Class View

Properties

MakeTempFilenameDLL2005 Pro

Misc

directory that we want to generate the unique filename in. The second argument, `szPrefix` is a string that will be the prefix of the temporary filename. The "GetTempFileName" API function will use the first three characters of this string when generating the unique filename. The function will return the unique filename if successful, or the string "ERROR" if it fails.

Create (any) function to export, and put it in your .cpp file

For example:

Because we want to return a string, we need to make the returned string a global variable. This is necessary because if we declare the string inside of the function it will be destroyed when the function is finished. We need to create a string that will last for the life of the DLL.

Your global string declaration and function should look like this:

```
TCHAR szReturn[MAX_PATH]; __declspec(dllexport) LPCTSTR RetrieveTempFilename(LPCTSTR szDirectory, LPCTSTR
szPrefix)
{
    TCHAR szBuffer[MAX_PATH];
    if(GetTempFileName(szDirectory, szPrefix, 0, szBuffer) == 0)
    {
        // It failed - return the string "ERROR"
        lstrcpy(szReturn, TEXT("ERROR"));
    } else {
        // Success!
        lstrcpy(szReturn, szBuffer);
    }
    return szReturn;
}
```

Note that the code above is Unicode safe so it will compile properly regardless of which character set you chose. You can copy and paste the code above right into your file below the "DllMain" function.

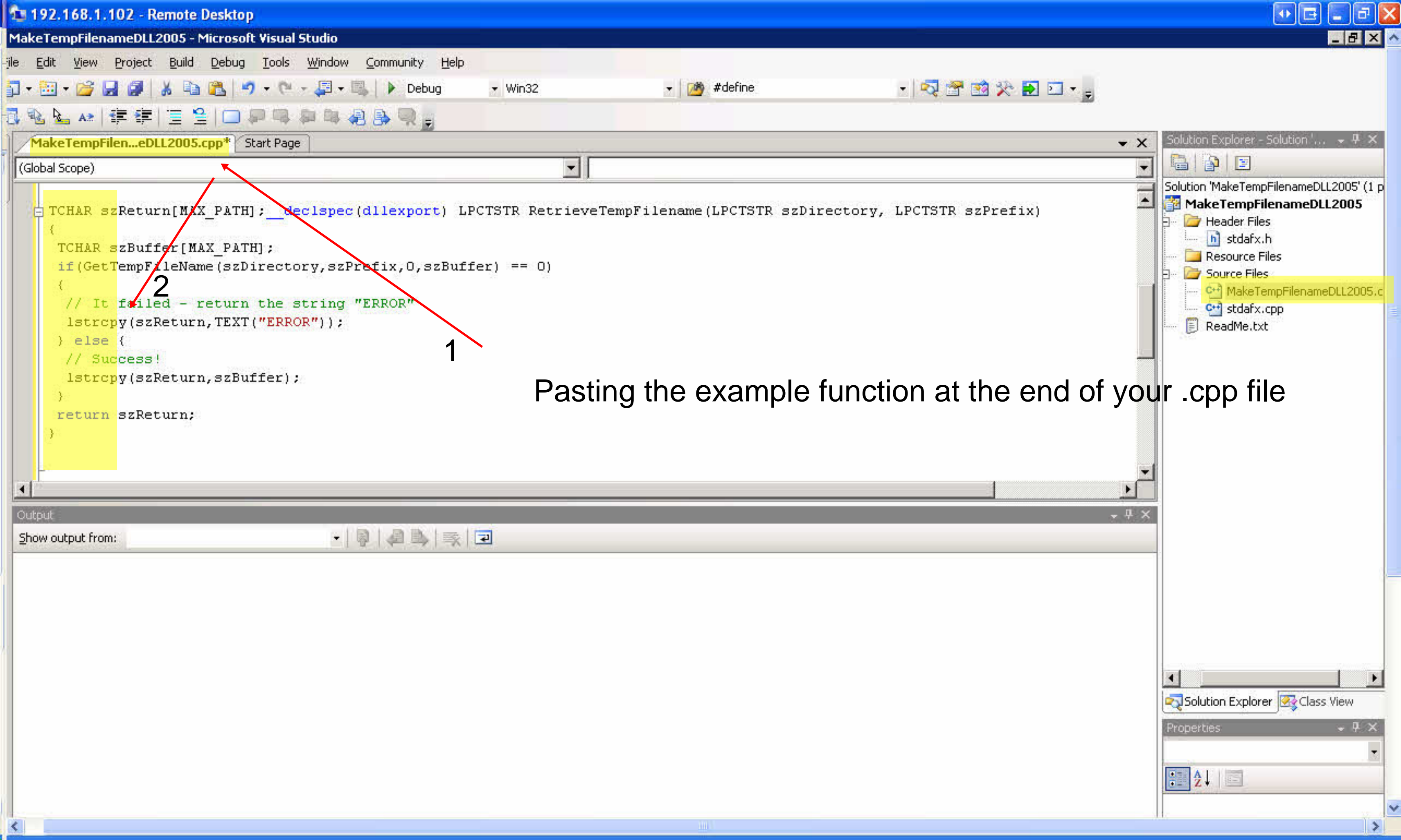
If you build your project now it will compile successfully but the function will still not be callable from outside programs. That is because the Visual C++ compiler "decorates" the function names so that they would not be recognizable to the calling program. Although there are several ways around this problem, we will use an "Export.def" file.

Add an "Export.def" File

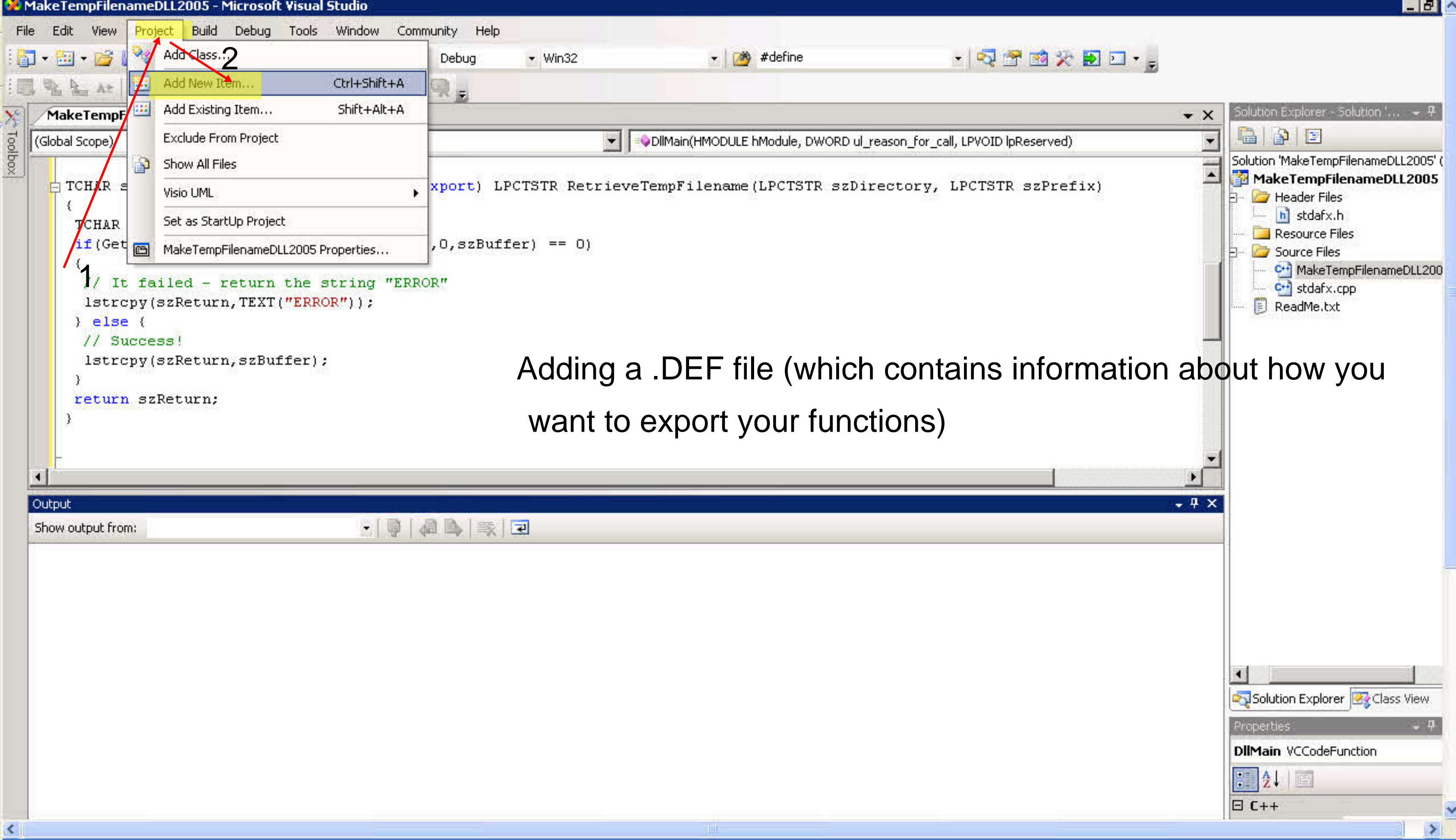
Adding an Export.def file will allow us to tell the compiler the proper names of our DLL functions that should be exposed.

1) Add a new text file to your project called "Export.def"

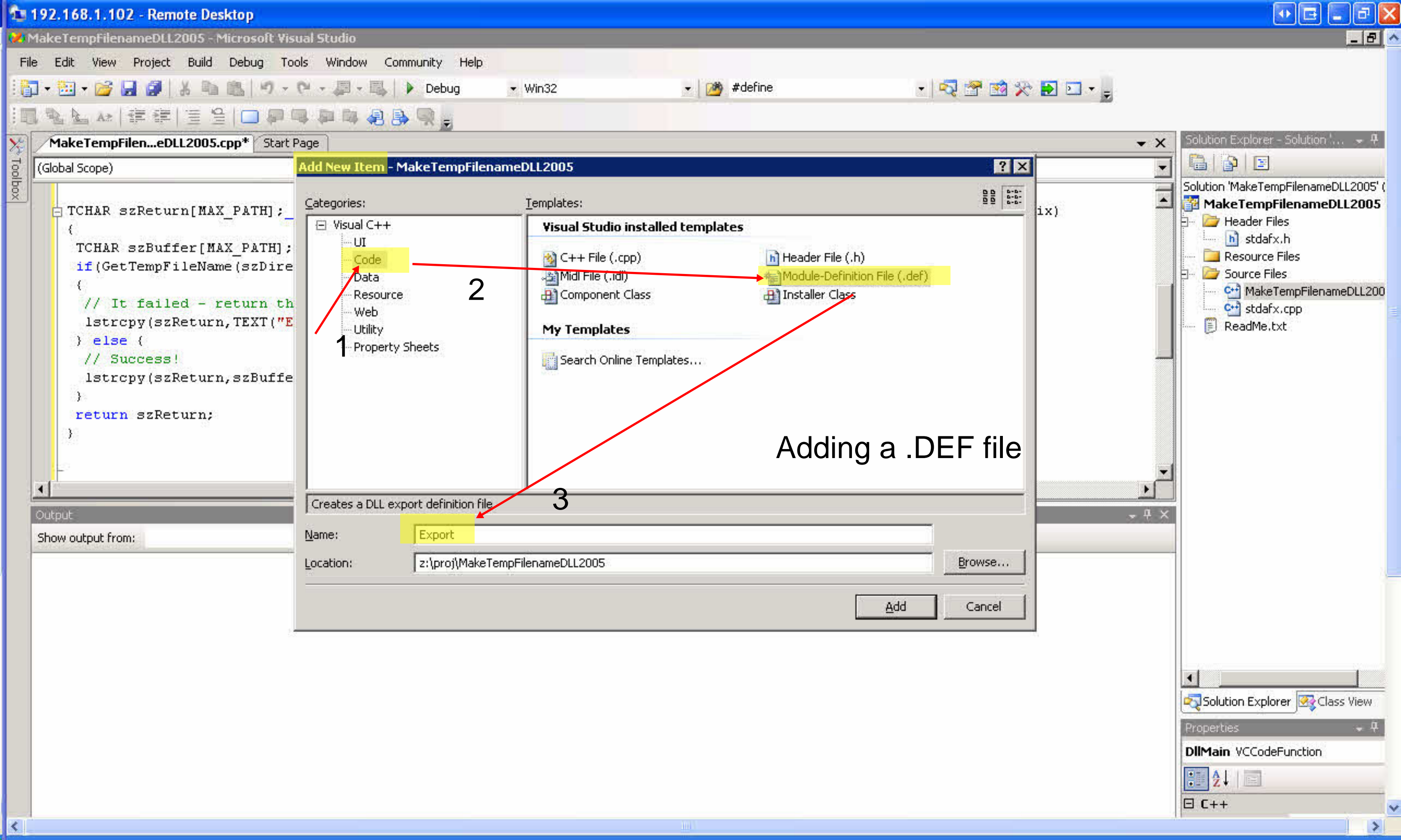
Copy/paste this
and paste it at the
bottom of the .cpp file
of your project

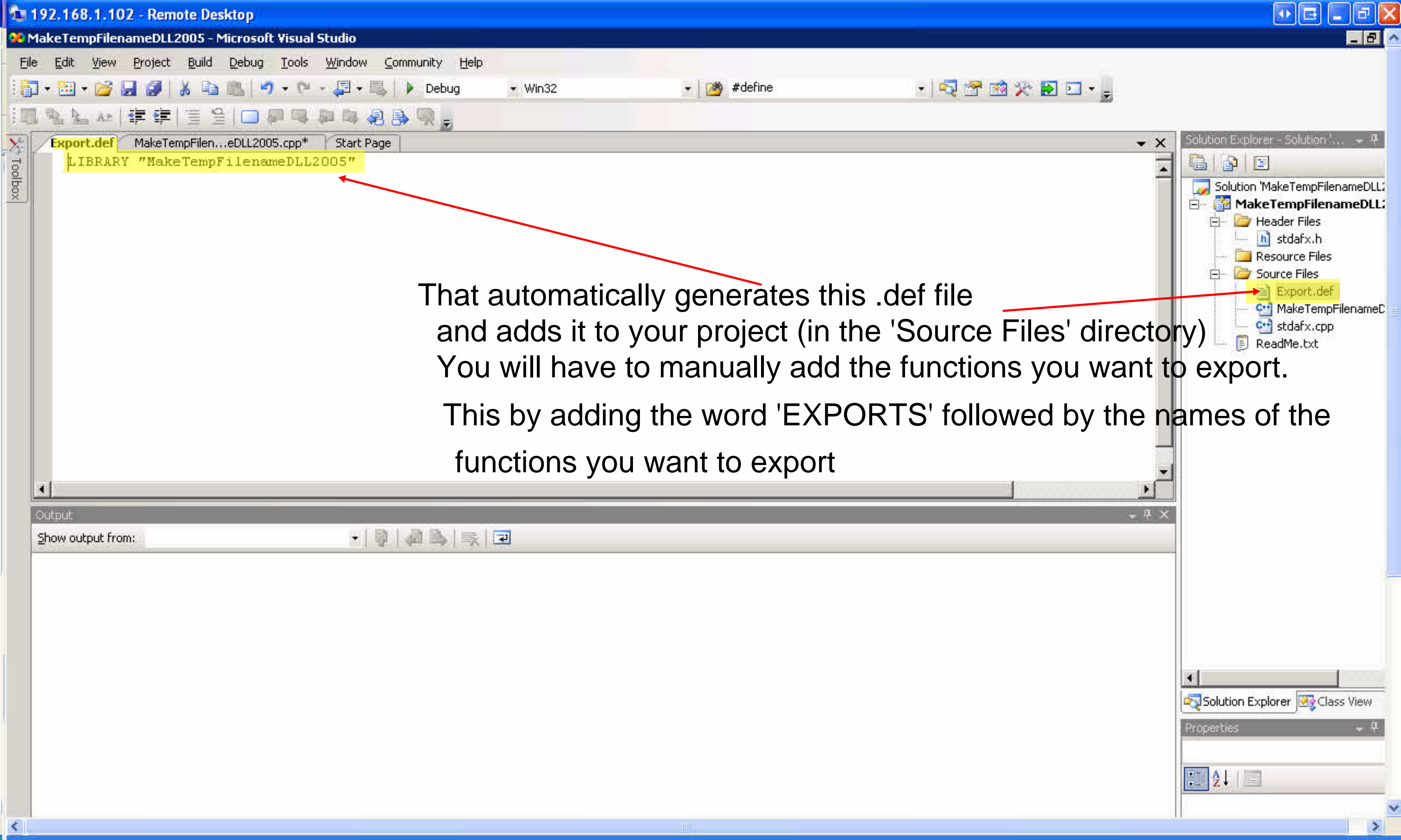


Pasting the example function at the end of your .cpp file

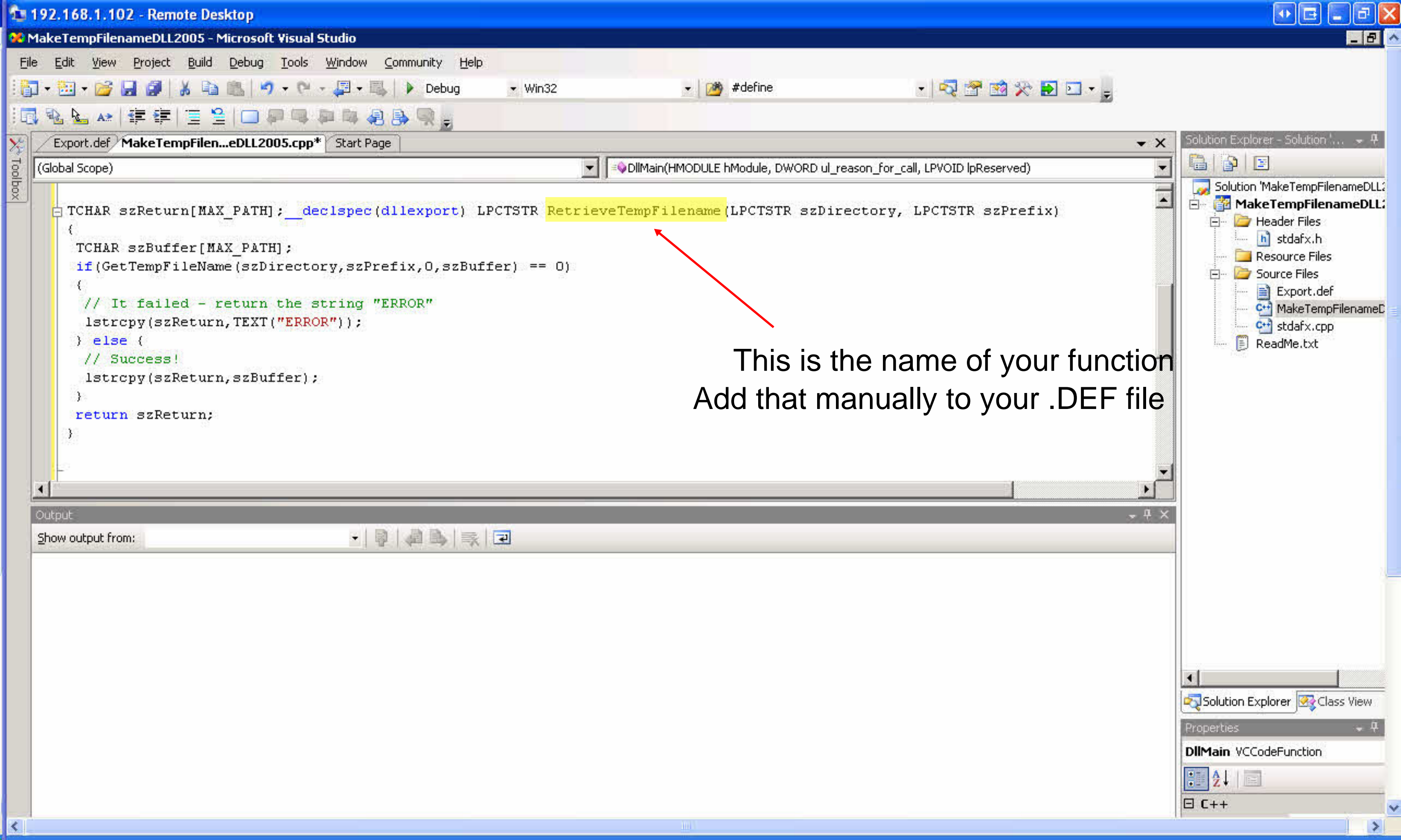


Adding a .DEF file (which contains information about how you want to export your functions)

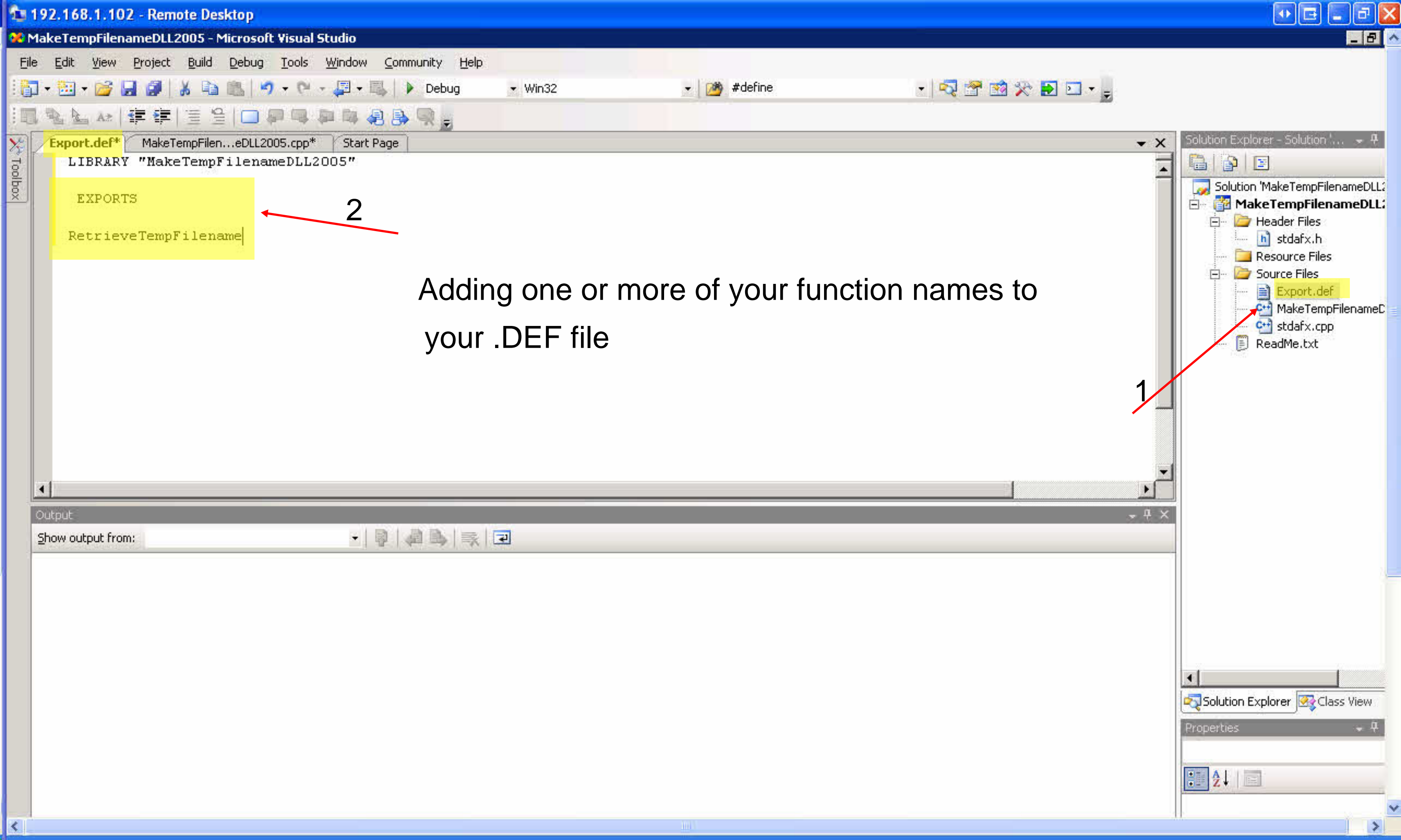




That automatically generates this .def file and adds it to your project (in the 'Source Files' directory) You will have to manually add the functions you want to export. This by adding the word 'EXPORTS' followed by the names of the functions you want to export



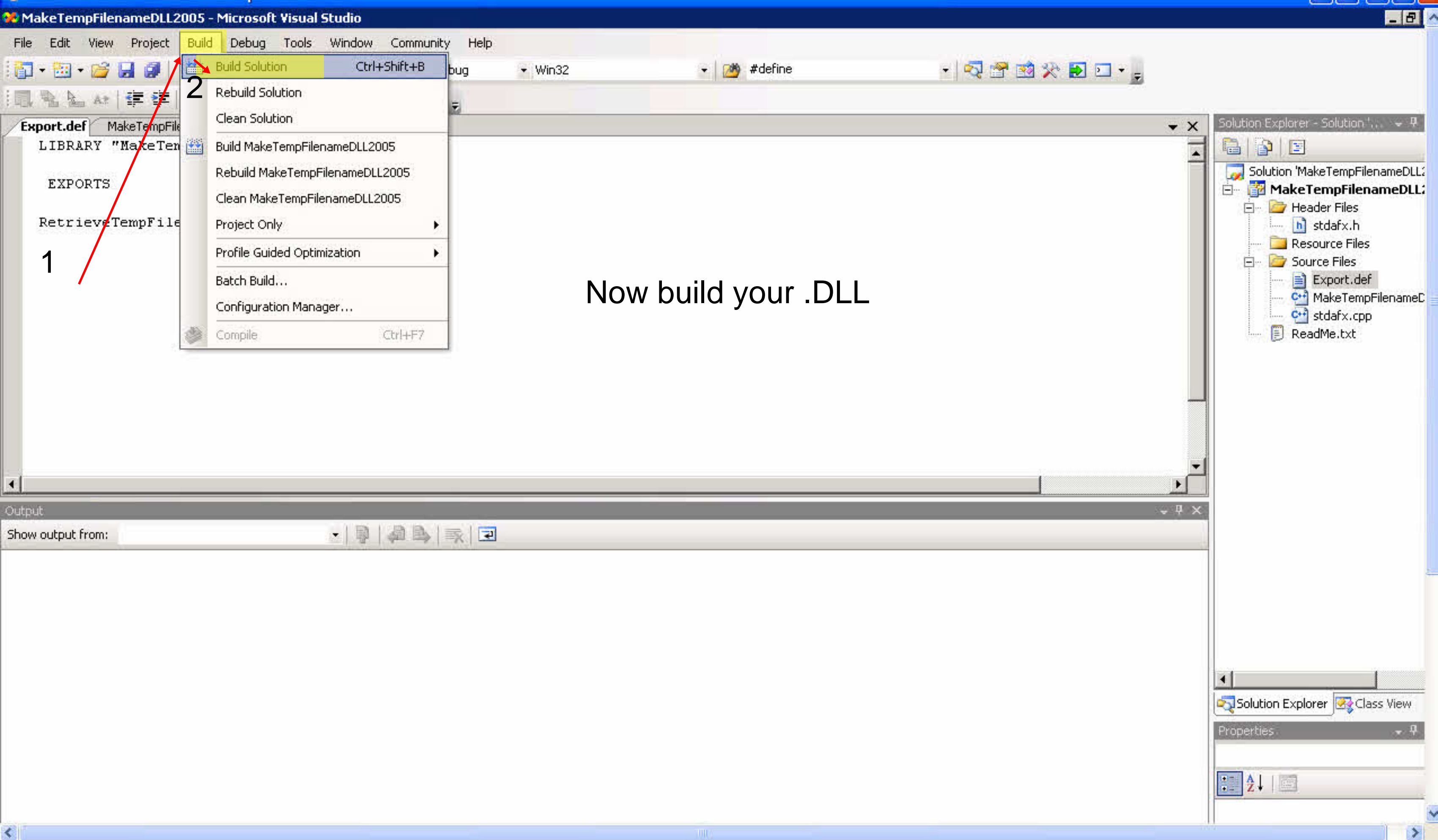
This is the name of your function
Add that manually to your .DEF file



2

1

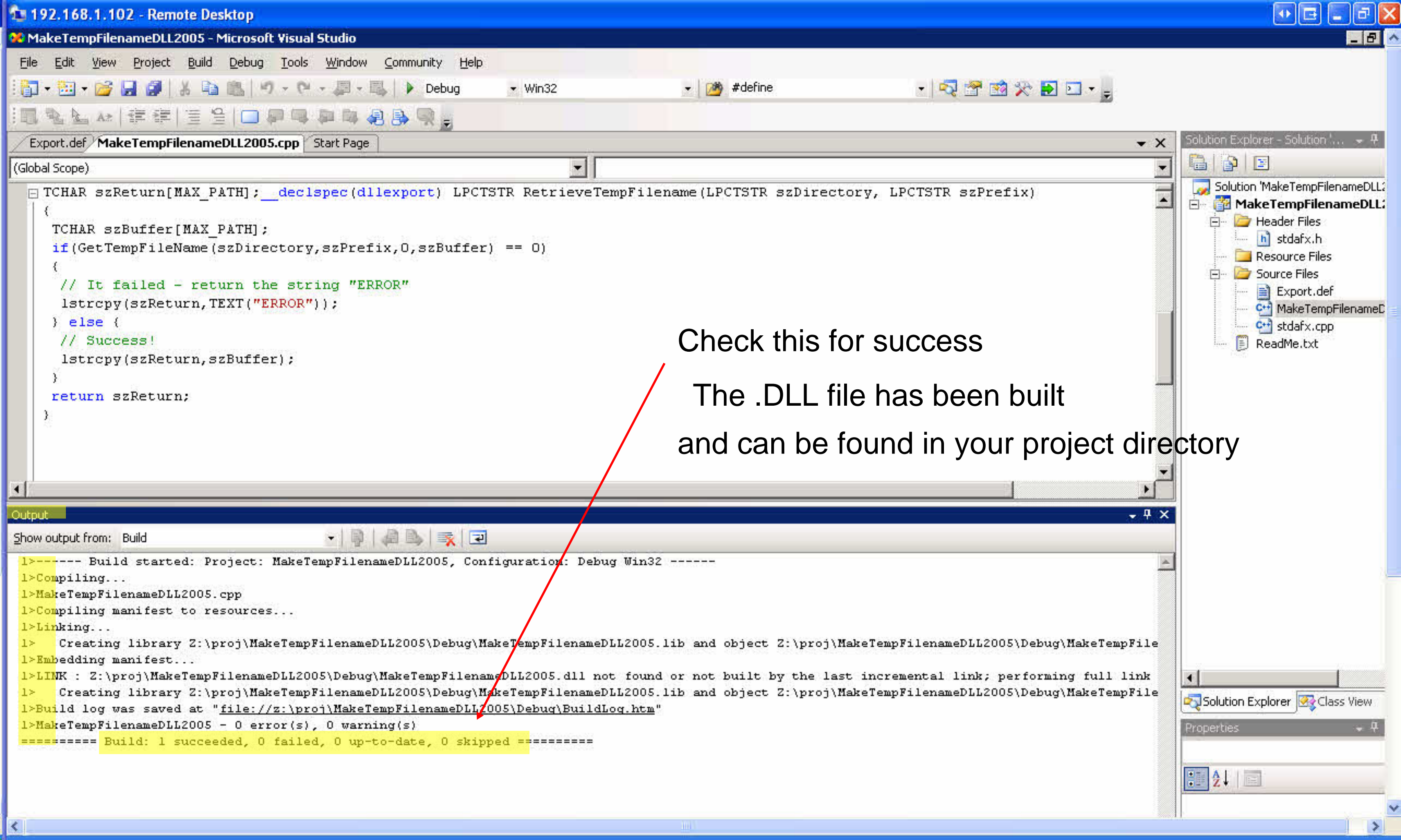
Adding one or more of your function names to your .DEF file



2

1

Now build your .DLL



Check this for success

The .DLL file has been built
and can be found in your project directory

```
(Global Scope)
TCHAR szReturn[MAX_PATH]; __declspec(dllexport) LPCTSTR RetrieveTempFilename(LPCTSTR szDirectory, LPCTSTR szPrefix)
{
    TCHAR szBuffer[MAX_PATH];
    if(GetTempFileName(szDirectory, szPrefix, 0, szBuffer) == 0)
    {
        // It failed - return the string "ERROR"
        strcpy(szReturn, TEXT("ERROR"));
    } else {
        // Success!
        strcpy(szReturn, szBuffer);
    }
    return szReturn;
}
```

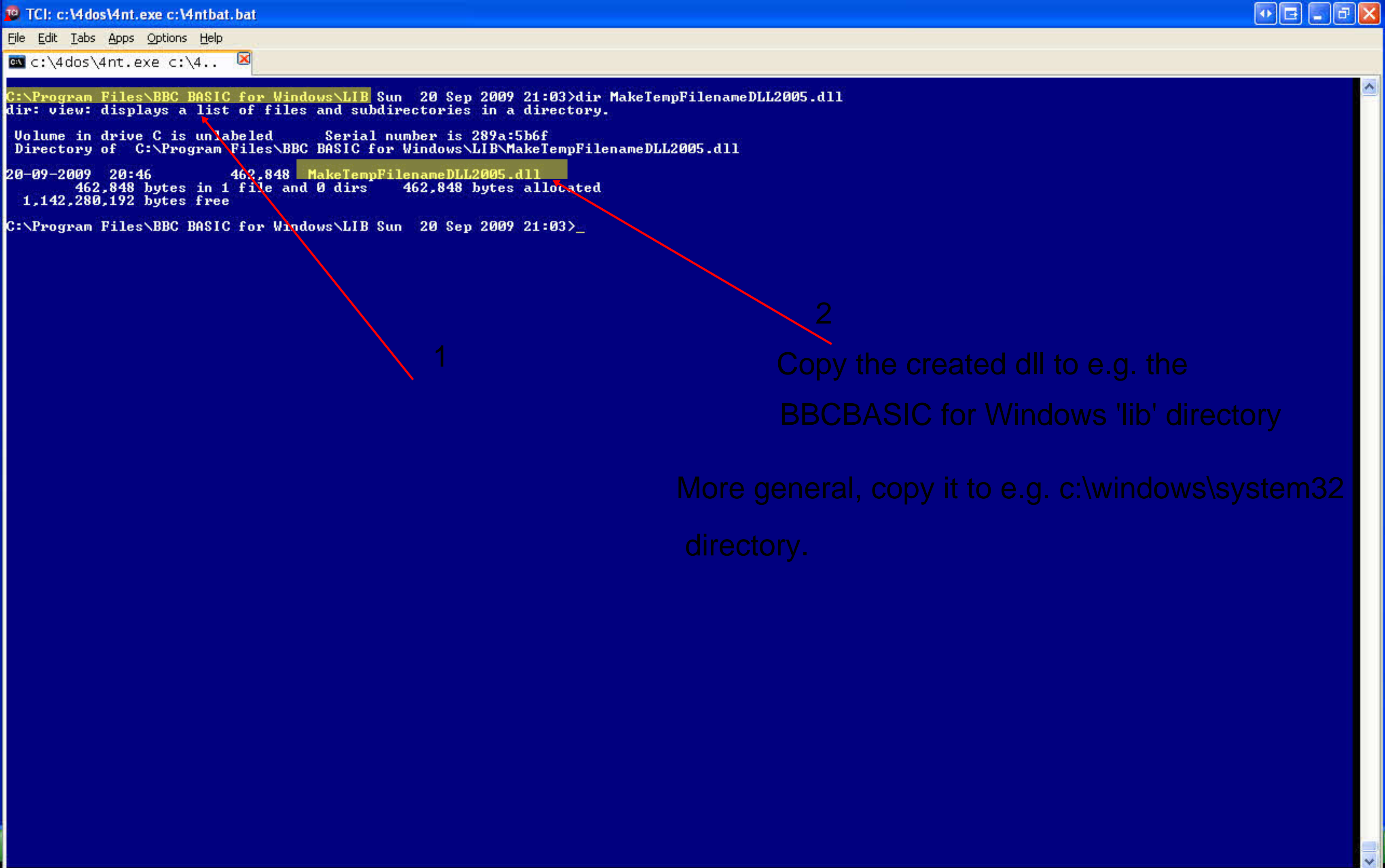
```
Output
Show output from: Build
1>----- Build started: Project: MakeTempFilenameDLL2005, Configuration: Debug Win32 -----
1>Compiling...
1>MakeTempFilenameDLL2005.cpp
1>Compiling manifest to resources...
1>Linking...
1> Creating library Z:\proj\MakeTempFilenameDLL2005\Debug\MakeTempFilenameDLL2005.lib and object Z:\proj\MakeTempFilenameDLL2005\Debug\MakeTempFile
1>Embedding manifest...
1>LINK : Z:\proj\MakeTempFilenameDLL2005\Debug\MakeTempFilenameDLL2005.dll not found or not built by the last incremental link; performing full link
1> Creating library Z:\proj\MakeTempFilenameDLL2005\Debug\MakeTempFilenameDLL2005.lib and object Z:\proj\MakeTempFilenameDLL2005\Debug\MakeTempFile
1>Build log was saved at "file:///z:/proj/MakeTempFilenameDLL2005/Debug/BuildLog.htm"
1>MakeTempFilenameDLL2005 - 0 error(s), 0 warning(s)
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

Solution Explorer - Solution '...

- Solution 'MakeTempFilenameDLL2005'
- MakeTempFilenameDLL2005
 - Header Files
 - stdafx.h
 - Resource Files
 - Source Files
 - Export.def
 - MakeTempFilenameD
 - stdafx.cpp
 - ReadMe.txt

Solution Explorer Class View

Properties



```
C:\Program Files\BBC BASIC for Windows\LIB Sun 20 Sep 2009 21:03>dir MakeTempFilenameDLL2005.dll
dir: view: displays a list of files and subdirectories in a directory.

Volume in drive C is unlabeled      Serial number is 289a:5b6f
Directory of  C:\Program Files\BBC BASIC for Windows\LIB\MakeTempFilenameDLL2005.dll

20-09-2009  20:46          462,848  MakeTempFilenameDLL2005.dll
             462,848 bytes in 1 file and 0 dirs   462,848 bytes allocated
1,142,280,192 bytes free

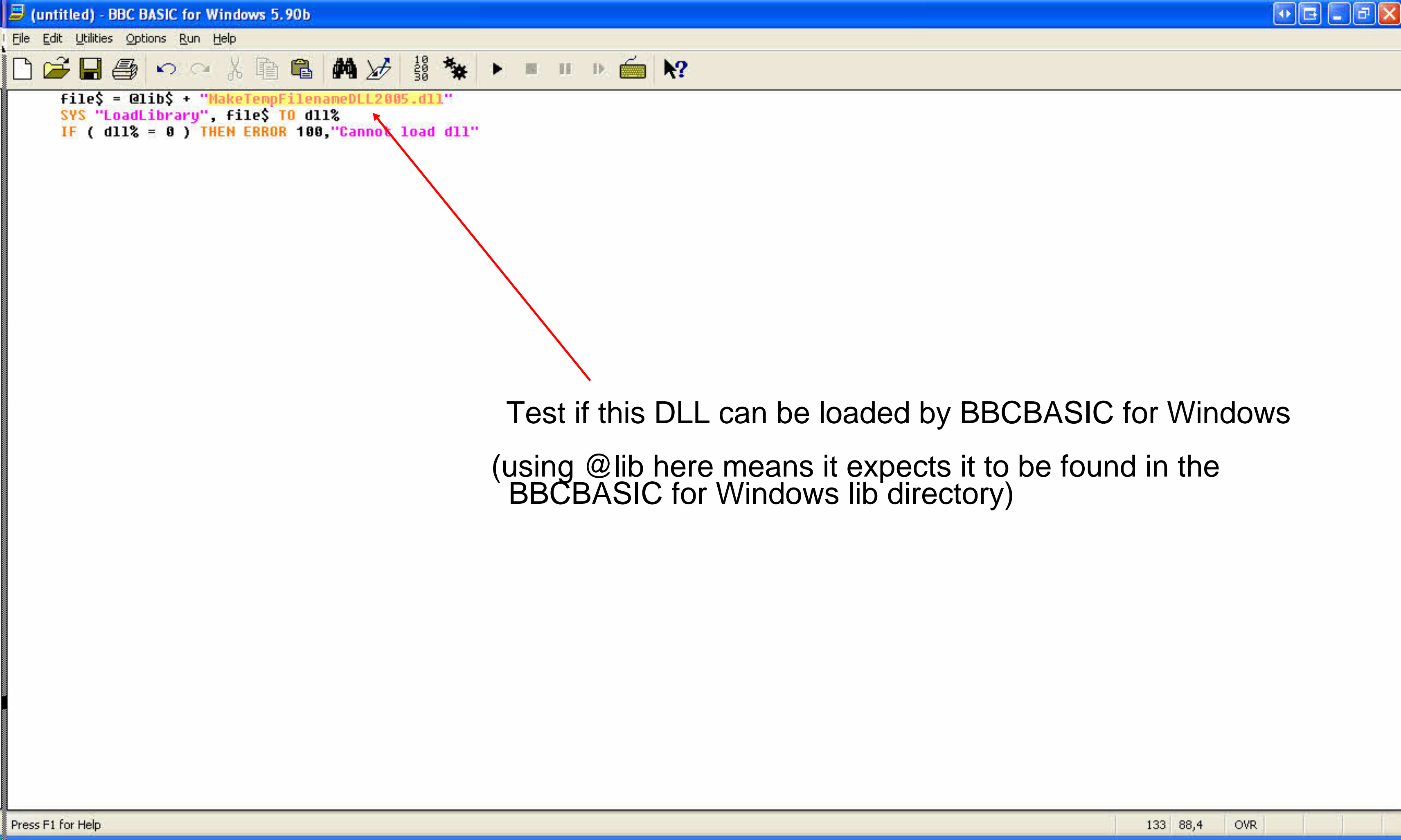
C:\Program Files\BBC BASIC for Windows\LIB Sun 20 Sep 2009 21:03>_
```

1

2

Copy the created dll to e.g. the BBCBASIC for Windows 'lib' directory

More general, copy it to e.g. c:\windows\system32 directory.

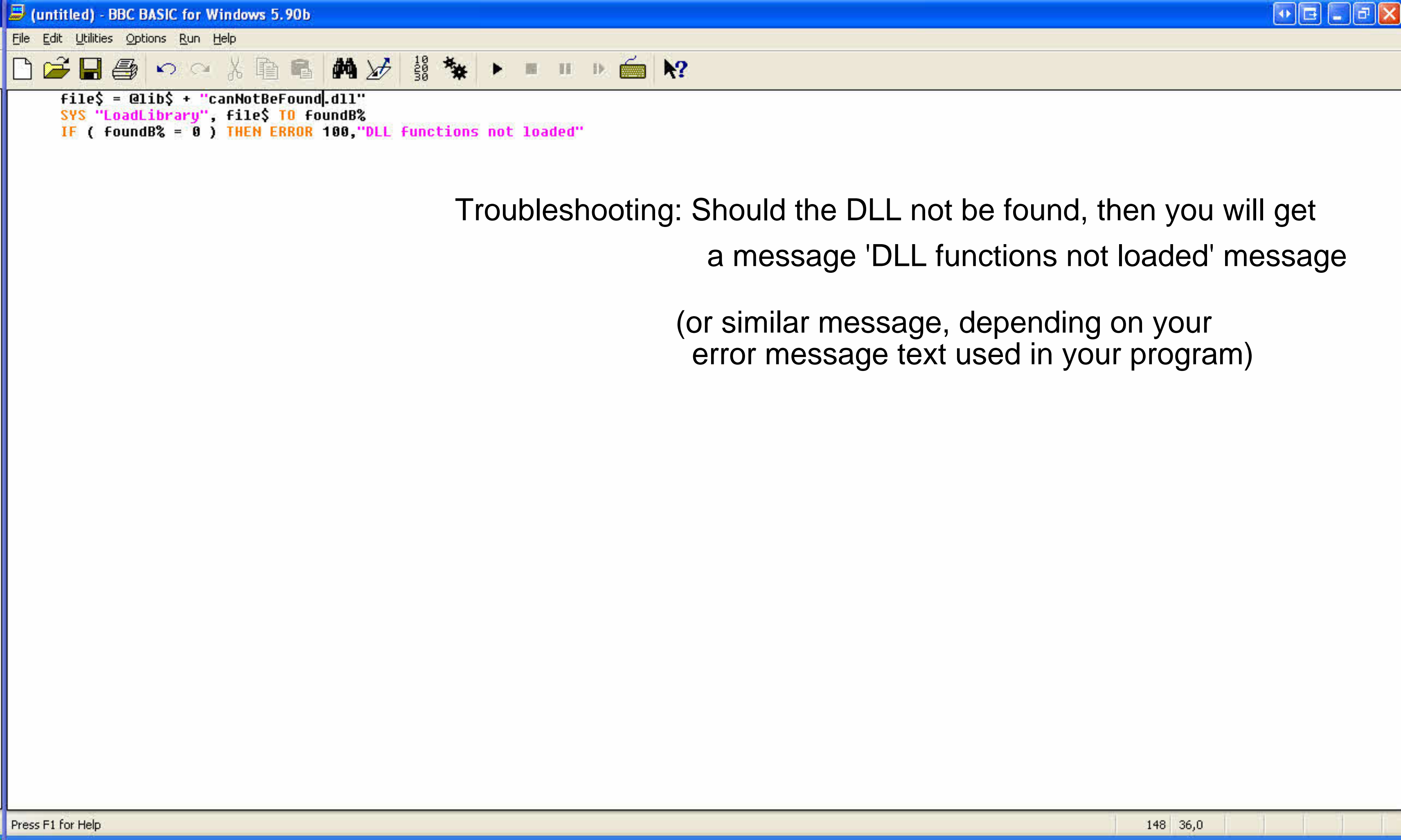


```
file$ = @lib$ + "MakeTempFilenameDLL2005.dll"  
SYS "LoadLibrary", file$ TO d11%  
IF ( d11% = 0 ) THEN ERROR 100, "Cannot load dll"
```

Test if this DLL can be loaded by BBCBASIC for Windows
(using @lib here means it expects it to be found in the
BBCBASIC for Windows lib directory)

```
>_
```

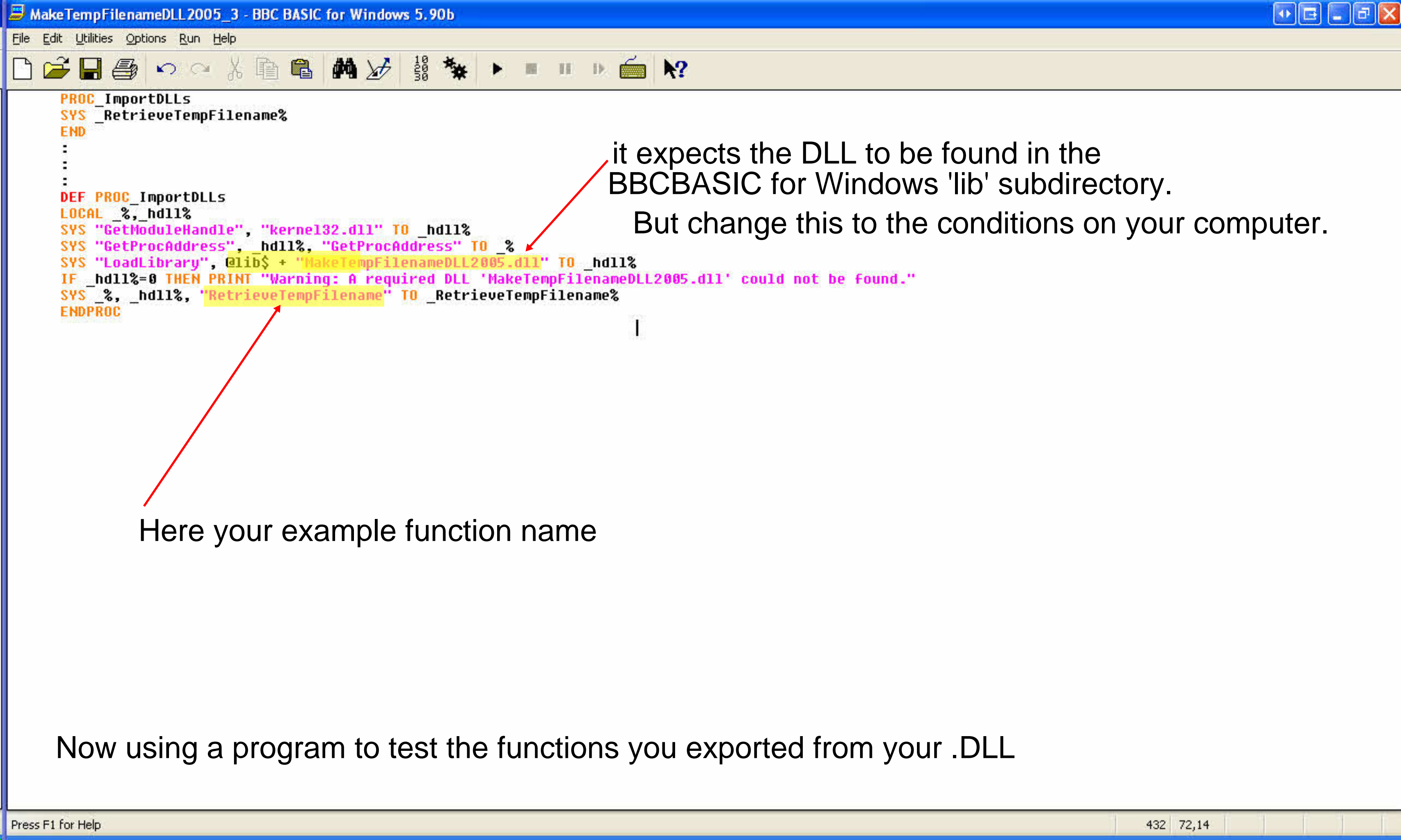
And if you see a blank screen, the DLL is recognized OK



Troubleshooting: Should the DLL not be found, then you will get a message 'DLL functions not loaded' message

(or similar message, depending on your error message text used in your program)

```
DLL functions not loaded  
>
```



it expects the DLL to be found in the BBCBASIC for Windows 'lib' subdirectory. But change this to the conditions on your computer.

Here your example function name

Now using a program to test the functions you exported from your .DLL

Address out of range
>

Running this DLL gives error 'out of range'
(but calling the DLL export function was successful)

This result was just because of the function used,
its address range was outside of the BBCBASIC for
Windows memory range.