To:
Tenouk

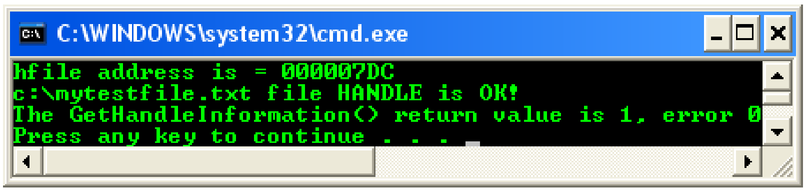
The Win32 Windows Programming
Configure And Using Windows PSDK- Part 4

Items in this page:

1. Building and running a Win32 Windows C program sample.
2. Windows Device Driver Kit, DDK/WDK.

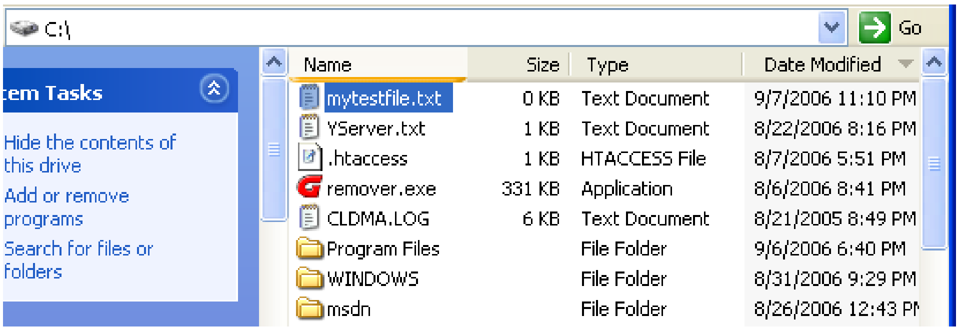
- Now we are ready to run our first an empty Windows console mode application using C and Microsoft C to test the PSDK. Create a new Win32 console application project named **win32prog**. Add a C++ source file named **win32progsrc**. Make sure you set your project to **Compile as C Code (/TC)**. Copy and paste the following sample code. Build and run the program.

```
// test program for Windows PSDK
#include <windows.h>
#include <stdio.h>

int main()
{
    // handle for file
    HANDLE hFile;
    // file and its path, the VC++ 2005 EE has been set to use unicode/wide character set...
    LPCWSTR fname = L"c:\\mytestfile.txt";
    // pointer to a variable that receives a set of bit flags that
    // specify properties of the object handle. Used in GetHandleInformation()
    DWORD lpdwFlags[100];
    BOOL test;

    // create a file with the given name
    hFile = CreateFile(fname,                  // file to be opened
                GENERIC_WRITE,                 // open for writing
                FILE_SHARE_WRITE,              // share for writing
                NULL,                          // default security
                CREATE_ALWAYS,                 // create new file only
                FILE_ATTRIBUTE_NORMAL |FILE_ATTRIBUTE_ARCHIVE
| SECURITY_IMPERSONATION,
                                               // normal file archive and impersonate client
                NULL);                 // no attribute template
    // check the return value...
    if(hFile == INVALID_HANDLE_VALUE)
        printf("Could not open %ls file, error %d\n", fname, GetLastError());
    // just to check the hfile address...
    printf("hfile address is = %p\n", hFile);
    // print some info...
    printf("%ls file HANDLE is OK!\n", fname);
    test = GetHandleInformation(hFile, lpdwFlags);
    printf("The GetHandleInformation() return value is %d, error %d\n", test, GetLastError());
    // don't forget to close the opened file handle
    CloseHandle(hFile);
    // optionally delete the created file. Try uncomment it
    // DeleteFile(fname);
    return 0;
}
```



```
hfile address is = 000007DC
c:\mytestfile.txt file HANDLE is OK!
The GetHandleInformation() return value is 1, error 0
Press any key to continue . . .
```

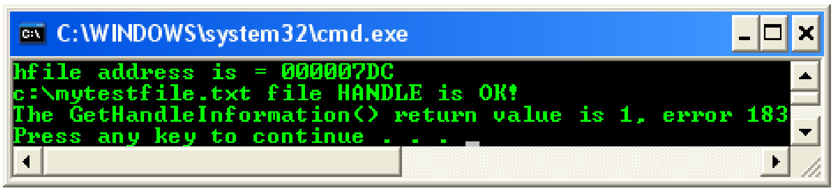
- If you check the root folder, **C:\**, the **mytestfile.txt** should be created.



C:\

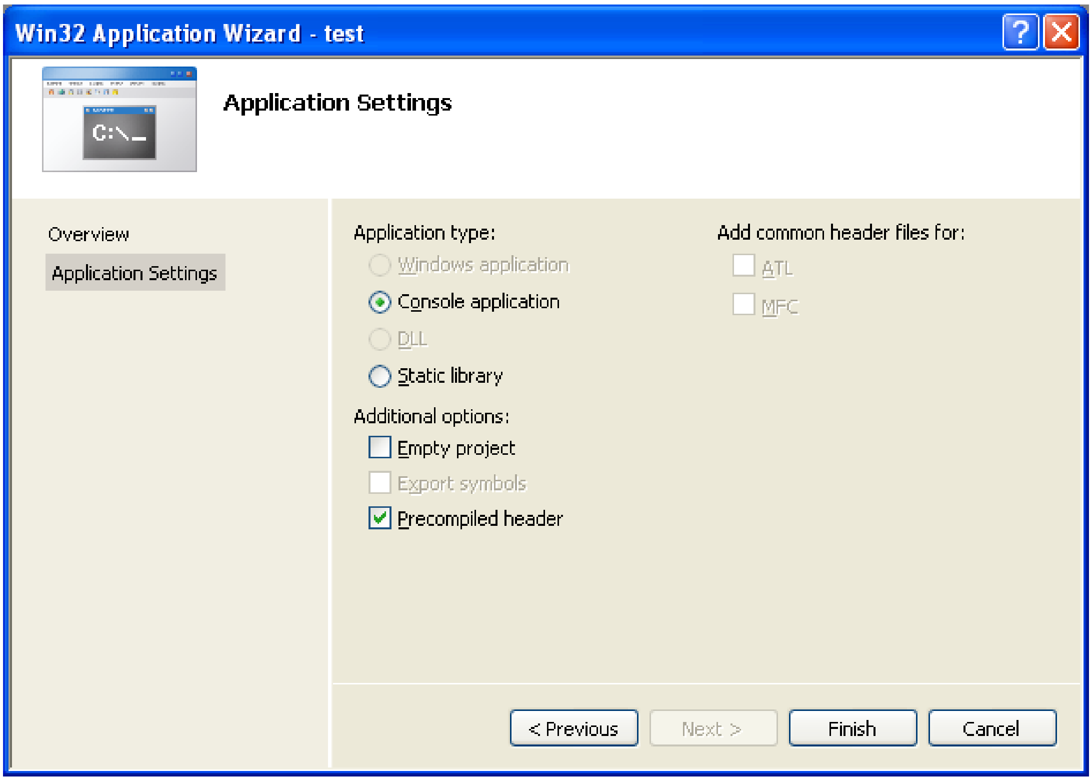
Name	Size	Type	Date Modified
mytestfile.txt	0 KB	Text Document	9/7/2006 11:10 PM
YServer.txt	1 KB	Text Document	8/22/2006 8:16 PM
.htaccess	1 KB	HTACCESS File	8/7/2006 5:51 PM
remover.exe	331 KB	Application	8/6/2006 8:41 PM
CLDMA.LOG	6 KB	Text Document	8/21/2005 8:49 PM
Program Files		File Folder	9/6/2006 6:40 PM
WINDOWS		File Folder	8/31/2006 9:29 PM
msdn		File Folder	8/26/2006 12:43 PI

- If you re-run the program, the output should return error 183 because the file was already exist. You can uncomment the DeleteFile(fname); line in order to avoid the error.



```
hfile address is = 000007DC
c:\mytestfile.txt file HANDLE is OK!
The GetHandleInformation() return value is 1, error 183
Press any key to continue . . .
```

- Seems our PSDK successfully installed and configured.
- In Visual C++ Express, the Win32 **Windows Application** type is disabled in the **Win32 Application Wizard** as shown below.



Application type:

○ Windows application
● Console application
○ DLL
○ Static library

Additional options:

□ Empty project
□ Export symbols
☑ Precompiled header

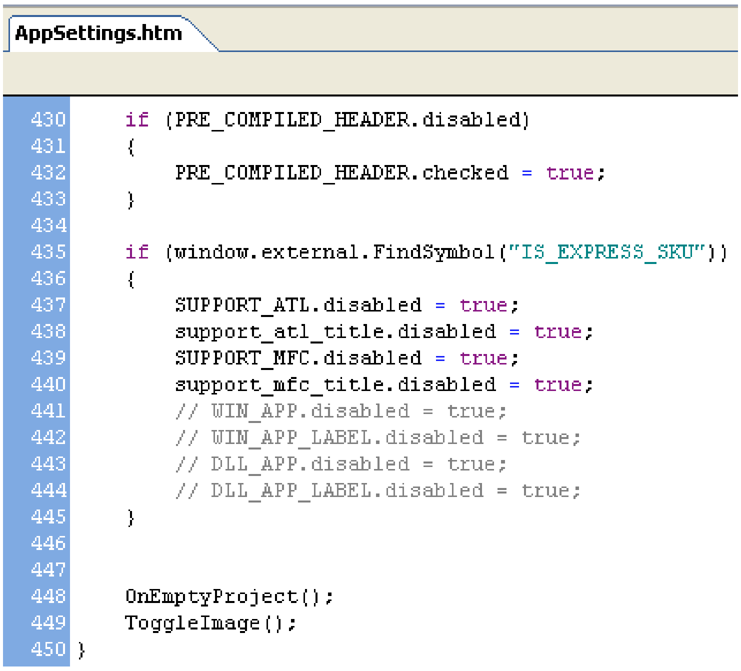
Add common header files for:

□ ATL
□ MFC

- To enable that type, you need to edit the file **AppSettings.htm** file located in the folder "**%ProgramFiles%\Microsoft Visual Studio 8\VC\VCWizards\AppWiz\Generic\Application \html\1033\**".
- In a text or html editor, comment out lines 441 - 444 by putting a // in front of them as shown here:

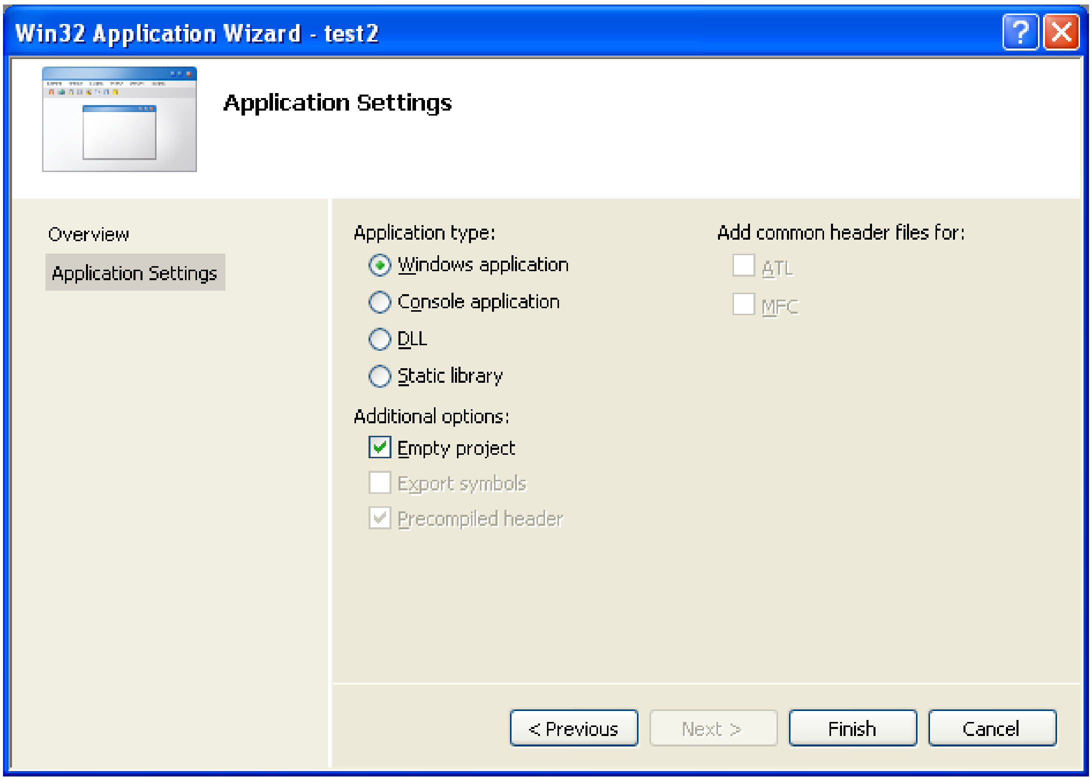
```
// WIN_APP.disabled = true;
// WIN_APP_LABEL.disabled = true;
// DLL_APP.disabled = true;
// DLL_APP_LABEL.disabled = true;
```

- Or without commenting those lines, you can change the **true** value to **false** value to serve the same purpose.



```
    if (PRE_COMPILED_HEADER.disabled)
    {
        PRE_COMPILED_HEADER.checked = true;
    }

    if (window.external.FindSymbol("IS_EXPRESS_SKU"))
    {
        SUPPORT_ATL.disabled = true;
        support_atl_title.disabled = true;
        SUPPORT_MFC.disabled = true;
        support_mfc_title.disabled = true;
        // WIN_APP.disabled = true;
        // WIN_APP_LABEL.disabled = true;
        // DLL_APP.disabled = true;
        // DLL_APP_LABEL.disabled = true;
    }


    OnEmptyProject();
    ToggleImage();
}
```

- Save and close the file and open Visual C++ Express.
- Do some testing. From the **File** menu, click **New Project**. In the **New Project** dialog box, expand the **Visual C++** node in the **Product Types** tree and then click **Win32**. Click on the **Win32 Console Application** template and then give your project a name, any name and click **OK**. In the **Win32 Application Wizard** dialog box, **Windows application** under the **Application type** is enabled. Click the **Cancel** button.



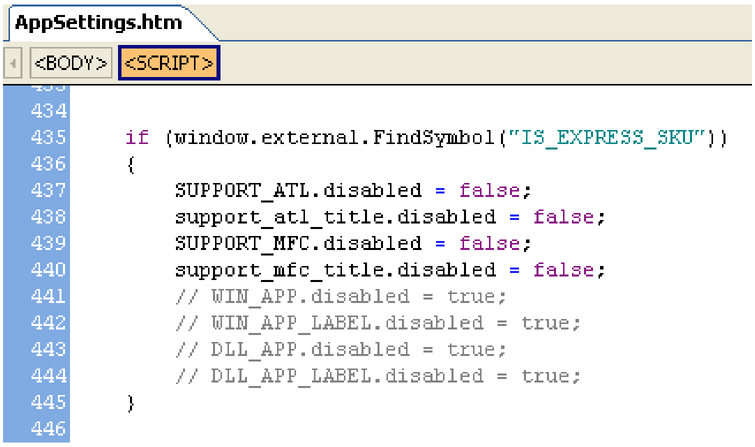
Application Settings

Application type:
● Windows application
○ Console application
○ DLL
○ Static library

Additional options:
☑ Empty project
☐ Export symbols
☑ Precompiled header

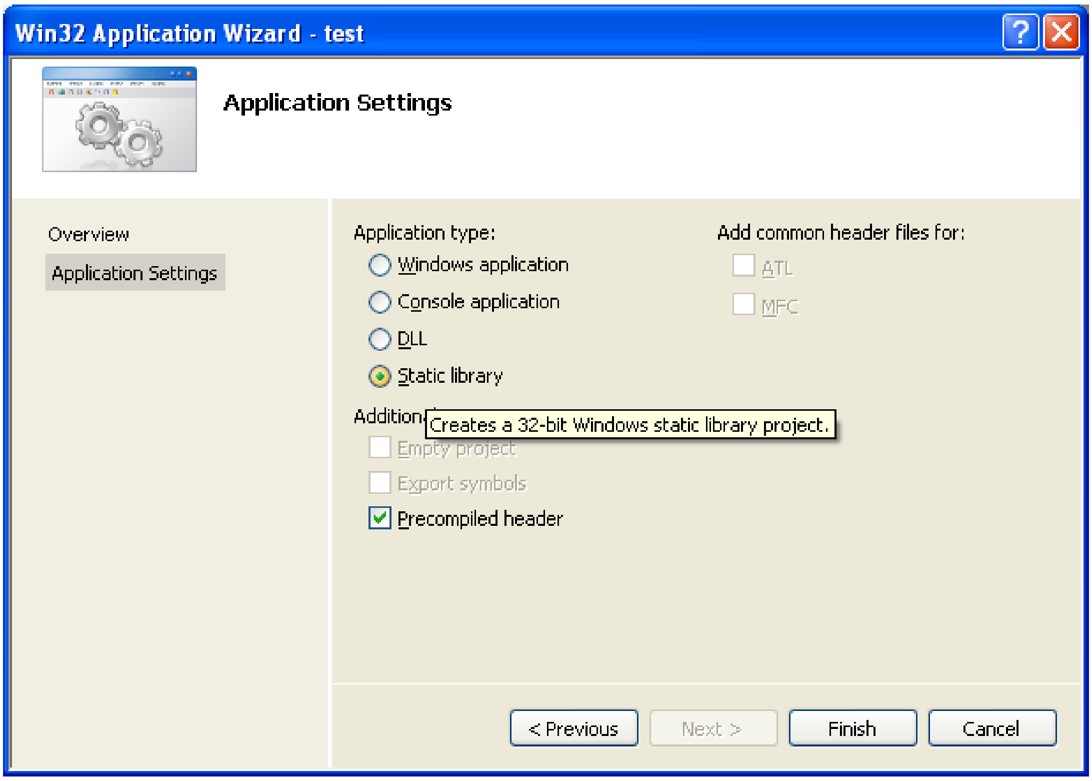
Add common header files for:
☐ ATL
☐ MFC

- Tenouk also enable other disabled options in the VC++ 2005 EE **Application wizard** as shown below by setting the **true** to **false** values.



```
    if (window.external.FindSymbol("IS_EXPRESS_SKU"))
    {
        SUPPORT_ATL.disabled = false;
        support_atl_title.disabled = false;
        SUPPORT_MFC.disabled = false;
        support_mfc_title.disabled = false;
        // WIN_APP.disabled = true;
        // WIN_APP_LABEL.disabled = true;
        // DLL_APP.disabled = true;
        // DLL_APP_LABEL.disabled = true;
    }
```

- When we check the **Application Settings** in the **Win32 Application Wizard** for example, the related options are enabled. An example is shown in the following Figures.



Application type:

○ Windows application
○ Console application
○ DLL
● Static library

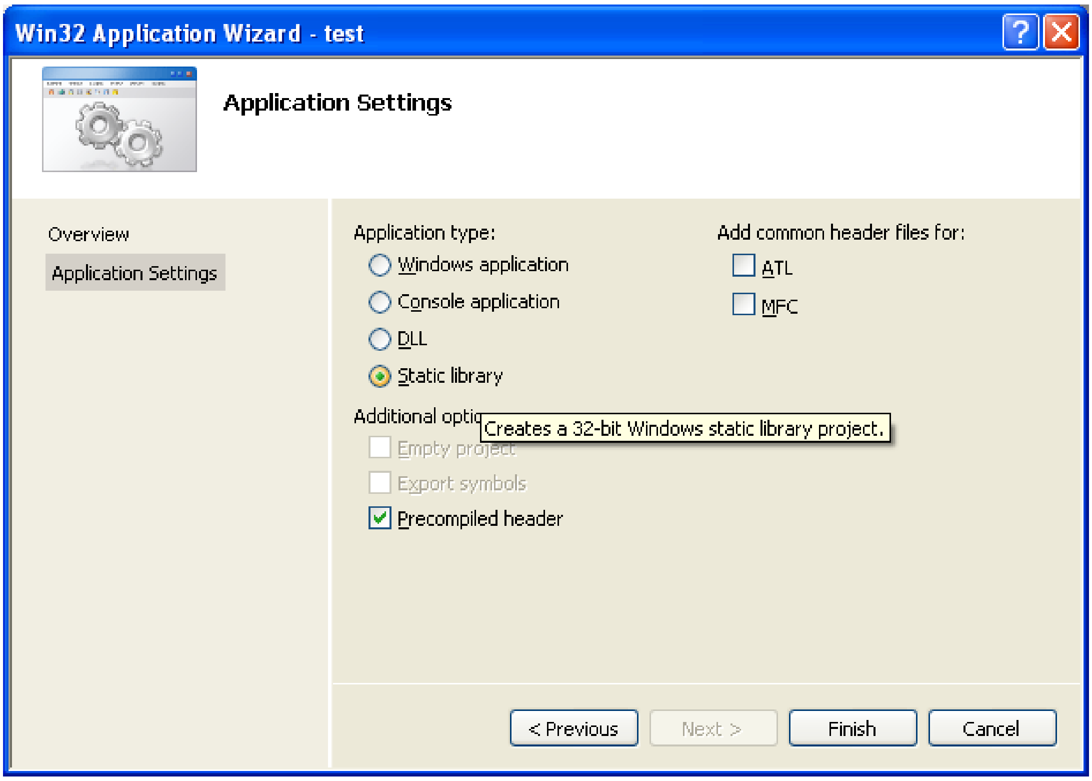
Add common header files for:

□ ATL
□ MFC

Addition

Creates a 32-bit Windows static library project.

□ Empty project
□ Export symbols
☑ Precompiled header



Win32 Application Wizard - test

Application Settings

Overview

Application Settings

Application type:

○ Windows application

○ Console application

○ DLL

● Static library

Additional optio

Creates a 32-bit Windows static library project.

□ Empty project

□ Export symbols

☑ Precompiled header

Add common header files for:

□ ATL

□ MFC

- From the previous sample Win32 (Microsoft) C code, you can see that for implementation specific, in this case Windows system, you need to find more information in order to build useful programs. Though our C knowledge provides a very good foundation, it is not so useful when we need to do platform or application specific coding. We need to dig more information. You can try compiling other Win32 C codes from Win32 Tutorial using VC++ 2005 EE but expect many warnings and/or errors! Previously, those codes compiled using VC++ 2003 .Net.

Windows Device Driver Kit (DDK)

- Other than the PSDK/SDK, sometime and somewhere you may find Windows DDK, Device Driver Kit. DDK is needed if you want to program Windows device drivers. At the moment this worksheet is prepared, to download Windows DDK you need to subscribe to MSDN or you can buy the CD. DDK also used when you want to program Windows Root Kit or if you want to access Windows kernel. Well, let summarize what we have done so far.

Item needed	What we can do?
Visual C++ Express Edition/ Visual Studio Express.	Building C/C++ console mode applications, standard C and Microsoft C.
PSDK/ SDK	Building C/C++ Windows based applications that include Win32 (Console mode Applications), Windows GUIs such as Windows forms, MFC, ATL, COM etc. PSDK/SDK should coming with Visual Studio 2003/2005 that includes Visual C++ 2003/2005 .Net.
DDK/ WDK	Building C/C++ Windows device drivers.

-------------------------End Windows pSDK: install, configure and use----------------------

| Main |< Configure And Use Windows PSDK 1 | C/C++ Array Data Type 1 >| Site Index | Download |

Windows SDK-Download, Install, Configure and Use: Part 1 | Part 2 | Part 3 | Part 4