

Installing Windows Shell version of LPJ-GUESS in Microsoft Visual C++ / Developer Studio Version 6

Follow these instructions carefully. Experience shows that failure to do so can cause much pain and result in hours to days of wasted work. Take particularly careful note of the items below marked **IMPORTANT**. A list of the most frequently encountered problems and their solutions appears at the end of this document.

1. Ensure that binary libraries **gutil.lib** and **plib.lib**, and their associated header files **gutil.h** and **plib.h** are installed. This only needs to be done once on each computer (different versions of guess can access the libraries in the same directory). These instructions assume the libraries are stored as follows:

C:\lib

gutil.lib

plib.lib

C:\lib\include

gutil.h

plib.h

2. Create a directory under which to install GUESS as a Visual C++ project. These instructions assume the path **C:\myguess**
3. Start **Developer Studio**. If this is the *first time* you are installing guess on this computer, add the path to the library header files to the list of directories in which header files are automatically sought:
 - **Tools|Options ...**
 - Select the **Directories** tab.
 - In the box labelled **Show directories for:** choose **Include files**
 - Add the path to the library header files (**C:\lib\include**) to the list
 - Press **OK**
4. In **Developer Studio**, create a new project called **Guess**. This will be stored in a subdirectory (**Guess**) of the directory you created in step 2. **IMPORTANT:** the Visual C++ project **MUST** be called **Guess** and **NOTHING ELSE**, otherwise the Windows Shell (the executable file **guesswin.exe**) will not find the model DLL. If you have several versions of the model, you can distinguish them by storing them under different directories at the level of step 2.
 - **File|New**
 - On **Projects** tab, select the item **Win32 Dynamic Link Library**
 - In the box labelled **Location**, enter (or browse to) the directory you created in step 2.
 - Under **Project name** enter **guess (NOTHING ELSE!!!!)**
 - **Check:** the **Location** box should now be updated to read "C:\myguess\guess" or equivalent

- Press **OK**. This brings up a window entitled **Win32 Dynamic Link Library – Step 1 of 1**
 - Choose **An empty DLL project**
 - Press **Finish**, then **OK**
- 5. In **Windows Explorer** or equivalent, move all of the module and framework source (.cpp) files and header (.h) files of GUESS to the project directory (**C:\myguess\guess**). Make sure that **main.cpp** and **main.h** are the Windows Shell versions of these files. All of the other files are common to the Windows Shell and command-line (Unix and Windows) versions.
- 6. In **Developer Studio**, add the source and header files to the project **guess**:
 - Choose **Project|Add To Project|Files ...**
 - Browse to the project directory (**C:\myguess\guess**) and choose (highlight) all of the GUESS source and header files (if you have followed the instructions above, this should be all of the .cpp and .h files in the directory)
 - Press **OK**
 - **Project|Add To Project|Files ...**
 - Choose **Library Files (.lib)** under **Files of type**
 - Browse to the library directory (**C:\lib**) and add **gutil.lib** and **plib.lib** to the project
- 7. Build guess as a dynamic-link library: **Build|Build guess.dll**
- 8. In **Windows Explorer**, place a copy of the Windows Shell executable file (**guesswin.exe**) in the **Debug** directory under the **guess** project directory
- 9. In **Developer Studio**, choose **Build|Execute**. The first time only, a window will appear, asking you to specify the name and location of the executable file. This is the Windows Shell file, **guesswin.exe**. In the box labelled **Executable file name**, enter the full pathname to **guesswin.exe** in the **Debug** directory of the current project (i.e. **C:\myguess\guess\debug\guesswin.exe**). **IMPORTANT:** make sure to specify this particular copy of **guesswin.exe** and **NO OTHER COPY**.
- 10. Press **OK** to run the model.
- 11. You can also run the model directly from outside Developer Studio by executing (e.g. by double-clicking) **C:\myguess\guess\debug\guesswin.exe**.

Note on project configurations:

By default, Developer Studio supplies two different project configurations, Debug and Release. These determine the level of optimisations and other features of the DLL created by **Build**. Release mode has the disadvantage of not including debugging information, but produces binary code that runs **considerably faster**.

To switch between Debug and Release modes choose **Build|Set Active Configuration ...** and choose the mode you want in the box labelled **Project Configuration**.

IMPORTANT: the first time you use Release mode, a subdirectory called **Release** will be created under the project directory (i.e. **C:\myguess\guess\Release**). In this directory, you

should place a copy of the Windows Shell, **guesswin.exe**. The first time you run the model from Developer Studio in Release mode, make sure to specify **EXACTLY THIS COPY** of **guesswin.exe** as the executable file.

Frequent problems and their solutions:

GUESS does not compile properly. Error messages like "Unresolved external symbol WinMain"

You have set up the project as a "Windows application" or something instead of a DLL. See Step 4.

GUESS does not compile properly. Error messages like "Unresolved external symbol xtring::..."

You have forgotten to add the libraries **gutil** or **plib** to the project. See Step 6.

GUESS does not compile properly. Error messages like "Cannot open include file"

You have forgotten to add the directory containing the library header files **gutil.h** and **plib.h** to the list of default directory searched by Developer Studio. See Step 3.

*GUESS seems to compile successfully, but when I try to run **guesswin.exe** an error message appears saying that a required DLL is missing, or a device attached to the system is missing, or that the program could not initialise properly.*

Are you using the Windows Shell version of **main.cpp** and **main.h**? See Step 5.
Is the project called **Guess**? See Step 4.

The model runs but it seems to ignore the changes I have made

It is probably running a different configuration or different version. The usual reason for this is that you have specified the wrong copy of **guesswin.exe** as the executable file for running the model from within Visual Studio. The correct version to use is the one in the **Debug** or **Release** directory of the **current** project (depending on whether Debug or Release mode is active). You can change the path to the executable on the **Debug** tab of the **Project Settings** dialogue (**Project|Settings ...**).

*The model runs but it seems to ignore the changes I have made. There is a copy of **guesswin.exe** in the Debug/Release directory and I have specified this version as the executable.*

Is the project called **Guess**?? **Guesswin.exe** looks for a DLL called **guess.dll**. If you have named the project something other than **Guess**, the DLL will have the wrong name, and **guesswin.exe** will not "see" it.