WINSVS code notes

7/13/2106

WinSVS was built as a stand-alone executable (winsvs.exe in the distribution but winsvs32.exe in the compiler settings) and four DLLs. The main executable handles all of the user interface tasks and calls functions from the DLLs as needed. Source code is somewhat poorly organized but everything is contained under the “winsvs” parent folder. On my computer, this full path to the parent folder is “C:\Bob's Stuff\Code\winsvs”. There is a good chance that none of the components will compile if the code is installed using a different path.

The code builds with Microsoft Visual Studio 2006 (Visual C++ V6.0). I don’t know if the code will build with a newer version of visual studio. I have not built the programs for at least 10 years. I tried recently but the problem mentioned in the next paragraph prevented a successful link. Changing the path to the lz32.lib file (changed Program Files to Programs Files (x86)) solved the problem and resulted in a successful build. You will get several warnings related to unreferenced variables and references to the GDI32.dll. I am pretty sure you can ignore the warnings. The unreferenced are related to experimental code that is not used in the final programs.

There is one external dependency that is needed to build winSVS. WinSVS uses a compression library that is included with VC++ V6.0. The needed lz32.lib file is located in this folder on my computer: “C:\Program Files (x86)\Microsoft Visual Studio\VC98\Lib”. The path in the workspace file is “C:\Program Files\Microsoft Visual Studio\VC98\Lib”. This path will not exist on computer running current version of windows (WinVista+) so you will need to change the path to the file in the workspace.

Dialogs are defined in resource files accessible through Visual Studio. Many of the dialogs have small buttons so there may not be enough room for translated text. The dialogs were originally designed to run on low-resolution screens so you should be able to enlarge the dialogs to make enough room to enlarge individual controls and buttons to accommodate the translated text.

The help files are all rich-text format (RTF) files and there are separate files for each of the DLLs and winSVS. The help files for winSVS are in the “winsvs\svs help” folder and those for the DLLs are in a “help” folder in each DLL’s code folder. There is no help file for the svslib DLL (this DLL handles all the rendering). The help files are formatted for use with Microsoft’s old help compiler that produced .HLP files. Support for this tool was dropped when WinVista was released. The current help tools produce .CHM files (compiled HTML). I think you might be able to find the old help compiler or the new tools may provide the .HLP format as an option. In either case, you may have trouble building the help files so they link with winSVS. There are specific formatting codes (footnote symbols at the beginning of each topic) inserted in the RTF files to define specific help topics and allow programs to provide context-sensitive information. The function that calls the help is “CallHelp” in the winacc.c file. I don’t know if the WinHelp() system function will support the newer compiled HTML format but you will likely need to change this function to use any translated help files. The tag parameter in CallHelp() provides the link to specific topics in the help file.

The SVS manual is located in the winsvs\svs help folder. This is a Word document that was built from the help text (with additional information added).