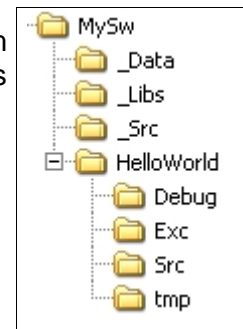


## Einstellungen bei der Arbeit mit MS Visual Studio 2010 (VS2010):

Die u.a. Vorgehensweise erzeugt ein VS-Projekt (intern geführt unter dem Namen Debug), das unterhalb eines bereits bestehenden Verzeichnisses [ ~\MySw\ ] HelloWorld eine Verzeichnisstruktur nutzt und

- die Projekt-Dateien unter HelloWorld\Debug,
- das lauffähige Test.exe unter HelloWorld\Exc,
- alle C/C++ - Quellen unter HelloWorld\Src,
- temporäre Dateien unter HelloWorld\tmp ablegt.



Das so implementierte Projekt nutzt (evtl. gemeinsam mit anderen Projekten)

- Daten unter [~\MySw\\_Data,
- Bibliotheken unter [~\MySw\\_Libs und
- Quelldateien unter [~\MySw\\_Src.

Test.exe kann sowohl von der Entwicklungsumgebung als auch von HelloWorld\Exc aus (Doppelklick) gestartet werden und in beiden Fällen zur Laufzeit auf Daten im \_Data-Verzeichnis zugreifen. Aktuelles Arbeitsverzeichnis ist HelloWorld\Debug.

Bereits compilierte Quellen (\*.obj) können ins Verzeichnis [~\MySw\\_Libs kopiert, per Drag&Drop im Projektmappen-Explorer des VS2010 als Quellcodedateien eingebunden und (auch ohne Vorhandensein des Quellcodes) vom Linker berücksichtigt werden.

Die Unterverzeichnisse von HelloWorld können auch als Grundlage für weitere Projekte verwendet werden – etwa für ein Projekt Hello2. Dazu sollte man

- das Verzeichnis HelloWorld2 erzeugen,
- die 4 Unterverzeichnisse von HelloWorld kopieren,
- mit der „Suchen“-Funktion des Explorers alle Dateien ermitteln und alle diejenigen darunter löschen, die man unter Hello2 nicht mehr verwenden möchte.

### Anmerkung:

Die unten aufgeführte Vorgehensweise mit dem VS ist so aufgezeichnet, daß die Eingaben aus der elektronischen Form dieses Dokumentes direkt in die Entwicklungsumgebung kopiert werden können.

Hinweise auf mögliche Irrtümer oder Mißverständnisse werden dankbar angenommen.

Leeres Projekt erstellen:	Datei ⇒ Neu ⇒ Projekt ⇒ Installierte Vorlagen: Visual C++ ⇒ Leeres Projekt ⇒ Name: Debug ⇒ Ort: [Pfad zum Fach] \SwProjekte ⇒ Projektmappenname (Häkchen bei „Verzeichnis für Lösung erstellen“): Test ⇒ OK
Verzeichnisstruktur erzeugen:	Zum erzeugten Ordner Debug hinzufügen: Exc, Src, tmp; in den Ordner Src Test.c und Test.h einer „Hello World“-Anwendung o.a. kopieren (ggf. Editor benutzen)
main u.a. einfügen:	Im Projektmappen-Explorer (Drag&Drop o. re. Maustaste) bzw. Projekt ⇒ [Neues / Vorhandenes] Element hinzufügen ⇒ ... Zu erstellende Dateien: Datei ⇒ Neu ⇒ Datei ⇒ C++-Datei ...
Konfiguration Ein-/Ausgabe:	Projekt ⇒ Debug-Eigenschaften... ⇒ Konfigurationseigenschaften ⇒ Allgemein: ⇒ Ausgabeverzeichnis: ..\Exc\ ⇒ Zwischenverzeichnis: ..\tmp\ ⇒ Zielname: Test
Bibliotheken und/oder Obj-Dateien einbinden:	Projekt ⇒ Debug-Eigenschaften... ⇒ Konfigurationseigenschaften ⇒ C/C++: ⇒ Allgemein ⇒ Zusätzliche Includeverzeichnisse: ..\Src;..\..\_Data;..\..\_Libs;..\..\_Src ⇒ Präprozessor ⇒ Präprozessordefinitionen: WIN32;_DEBUG;_CONSOLE;% (PreprocessorDefinitions) ⇒ Linker ⇒ Allgemein ⇒ Zusätzliche Bibliotheksverzeichnisse: ....\_Data;..\..\_Libs;..\..\_Src; ....\_Dat ⇒ Eingabe ⇒ Zusätzliche Abhängigkeiten (nur bei Bedarf!): OpenGL32.Lib GLU32.Lib GLUT32.lib Falls benötigt, auch: mui.lib ..\tmp\App.obj
Compilieren:	Erstellen ⇒ Debug neu erstellen
<b>Optionale Zusätze:</b>	
Multithreading erlauben:	Projekt ⇒ Debug-Eigenschaften ... ⇒ Konfigurationseigenschaften ⇒ C/C++ ⇒ Codegenerierung ⇒ Laufzeitbibliothek: Multithreaded-Debug (/MTd) (Voreinstellung: Multithreaded-Debug-DLL (/MDd))
Bei Bedarf Unterdrückung des „Console window“: – alternativ dazu im Code: Problem unter Win 7:	Projekt ⇒ Debug-Eigenschaften... ⇒ Konfigurationseigenschaften ⇒ Linker ⇒ Befehlszeile ⇒ Weitere Optionen: /entry:"mainCRTStartup" /subsystem:windows #pragma comment( linker, "/subsystem:\"windows\"" /entry:\"mainCRTStartup\" " ) Konsole-Fenster wird immer kurz eingeblendet

**Achtung:** Bei Nutzung der GLUT-Library ist es ratsam, eine Kopie von GLUT32.DLL dort abzulegen, wo auch opengl32.dll liegt – das ist i.d.R. unter C:\WINDOWS\system32.

<b>Verbleibende Probleme:</b>	<ul style="list-style-type: none"> <li>• Beim ersten Compilieren Frage, wo Debug.sln zu speichern ist (⇒ OK)</li> <li>• Bei jedem Start des entwickelten Programms (F5 o.ä.) Meldung, das Projekt sei veraltet</li> </ul>
<b>Evtl. zu ändern:</b>	Projekt ⇒ Debug-Eigenschaften... ⇒ Konfigurationseigenschaften ⇒ Linker ⇒ Allgemein ⇒ Inkrementelles Verknüpfen aktivieren: Nein
<b>Änderung des Arbeitsverzeichnisses (meist unnötig):</b>	Projekt ⇒ Debug-Eigenschaften... ⇒ Konfigurationseigenschaften ⇒ Debuggen ⇒ Arbeitsverzeichnis
<b>Vermeidung des Verzeichnisses Debug\ipch:</b>	Extras ⇒ Optionen ⇒ Text-Editor ⇒ C/C++ ⇒ Erweitert ⇒ Immer Ausweichpfad verwenden: False
<b>Seltenes Kompatibilitätsproblem:</b>	Projekt ⇒ Debug-Eigenschaften ... ⇒ Konfigurationseigenschaften ⇒ C/C++ ⇒ Codegenerierung: Überprüfen von kleinen Typen: Ja (/RTCc)