# יצירת קבצי משטחים (DLL)

זימקס מחפש את הקבצים שמתארים משטחים בתיקיית המשטחים המקומית.

במחשב שלי מדובר בתיקייה:

C:\Users\ypokorec\Documents\Zemax\DLL\Surfaces

כדי לכתוב קובץ DLL, מומלץ להיעזר באתרים:

<https://customers.zemax.com/os/resources/learn/knowledgebase/how-to-compile-a-user-defined-surface>  
Which moved to the following URL:  
<https://my.zemax.com/en-US/Knowledge-Base/kb-article/?ka=KA-01787>

<https://docs.microsoft.com/en-us/cpp/build/walkthrough-creating-and-using-a-dynamic-link-library-cpp?view=vs-2019>

או בקיצור:

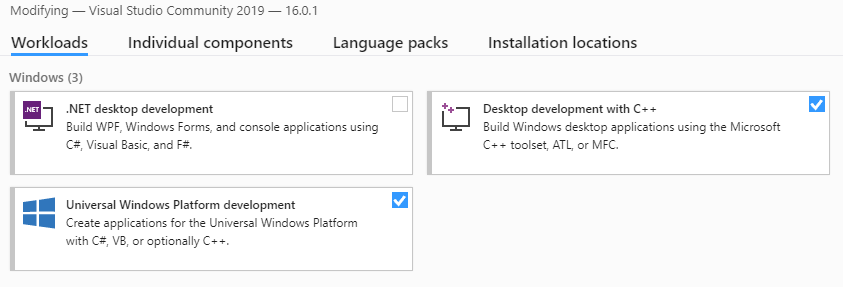
צריך להתקין מהאינטרנט ויזואל סטודיו <https://visualstudio.microsoft.com/downloads/>

אפשר לבחור גרסה חינמית (community)

להלן פרטים משלימים למידע שבאתרים הנ"ל:

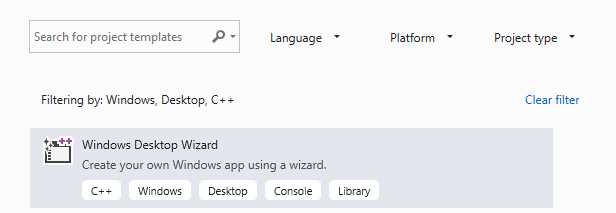
1. צריך להתקין ב-visual studio את הכלים:
2. Desktop development with C++
3. Universal Windows Platform development

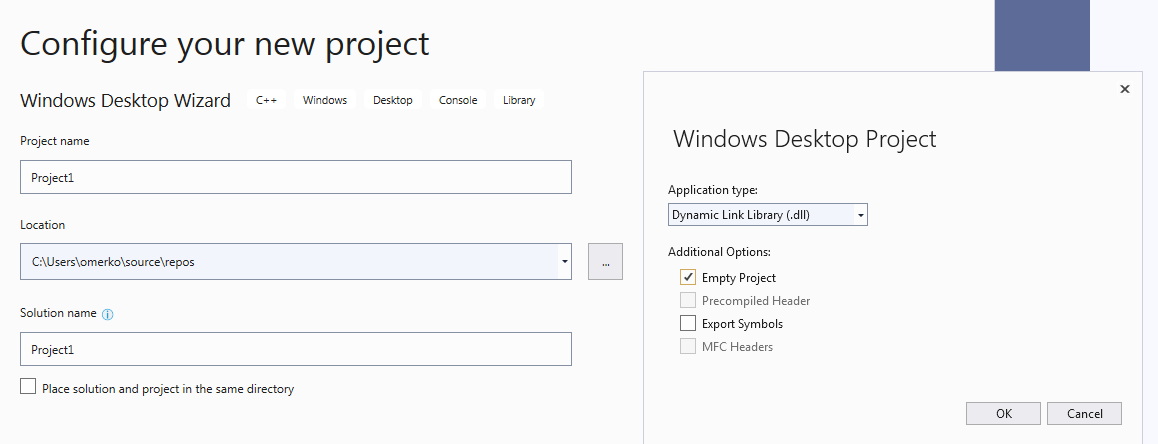
ייתכן שאחד מהם מספיק, אבל בהחלט אפשר להתקין את שניהם



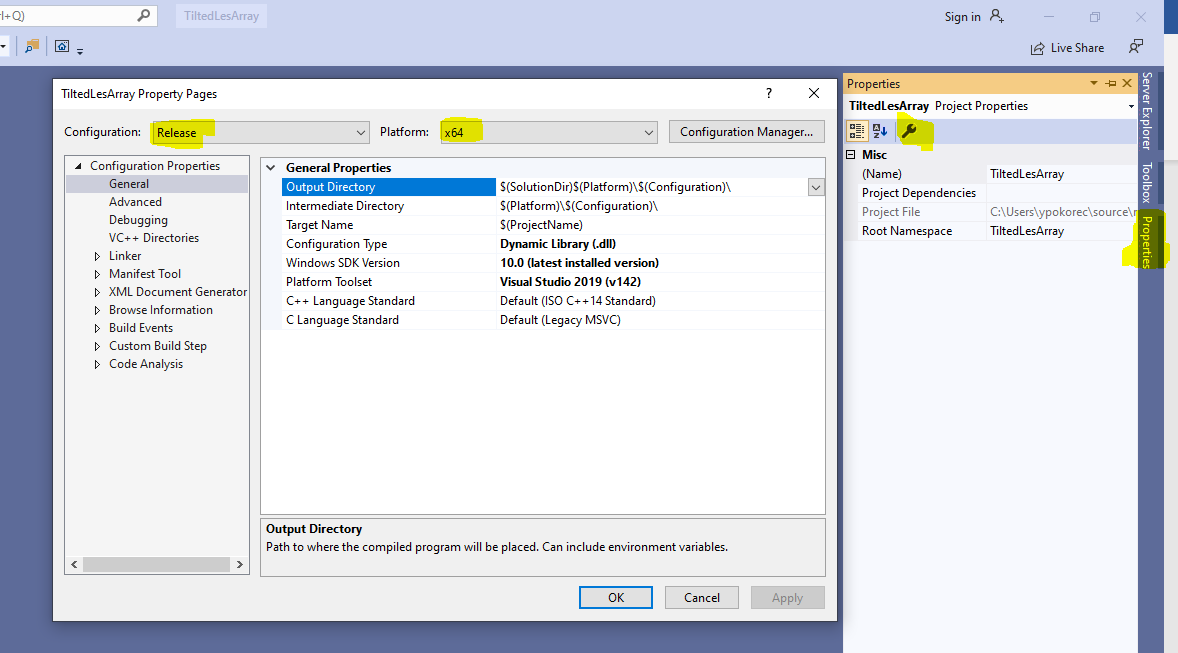
1. בפתיחה ראשונה של visual studio

אני בחרתי ב:  
מראה של סביבת פיתוח C++  
צור פרוייקט חדש  
Windows Desktop Wizard (מתאפשר רק לאחר התקנת החבילות בשלב הקודם)  
Application type: Dynamic Link Library  
Empty project

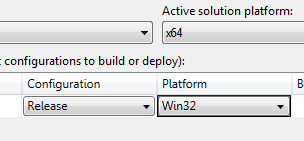
בפעם הקודמת שהתקנתי (לפני שנה במפרולייט) השקול של הנ"ל היה:



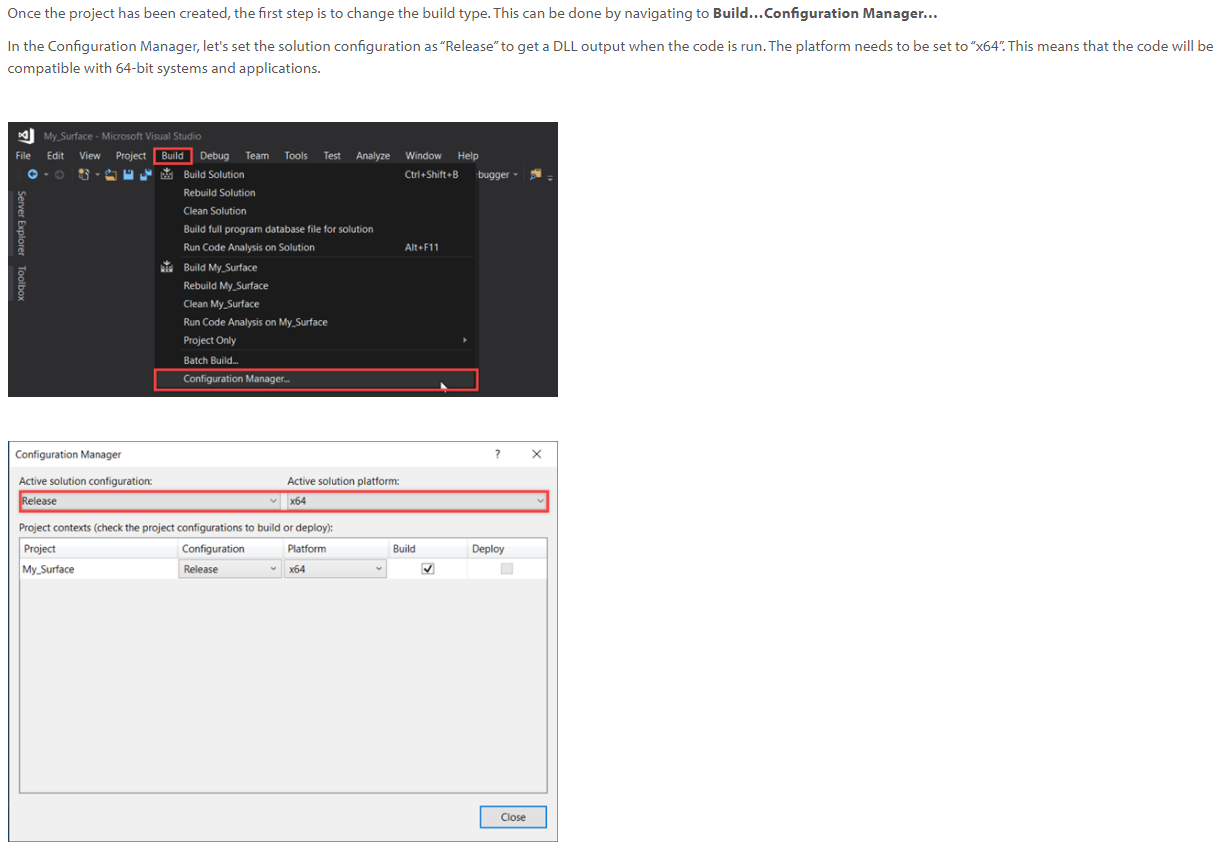
1. צריך לקנפג את הקימפול בהתאם לגרסת הזימקס שרצה על המחשב ל- 32 או 64 ביט  
   זה נמצא בלשונית properties מצד ימין --> סימון של מפתח שוודי



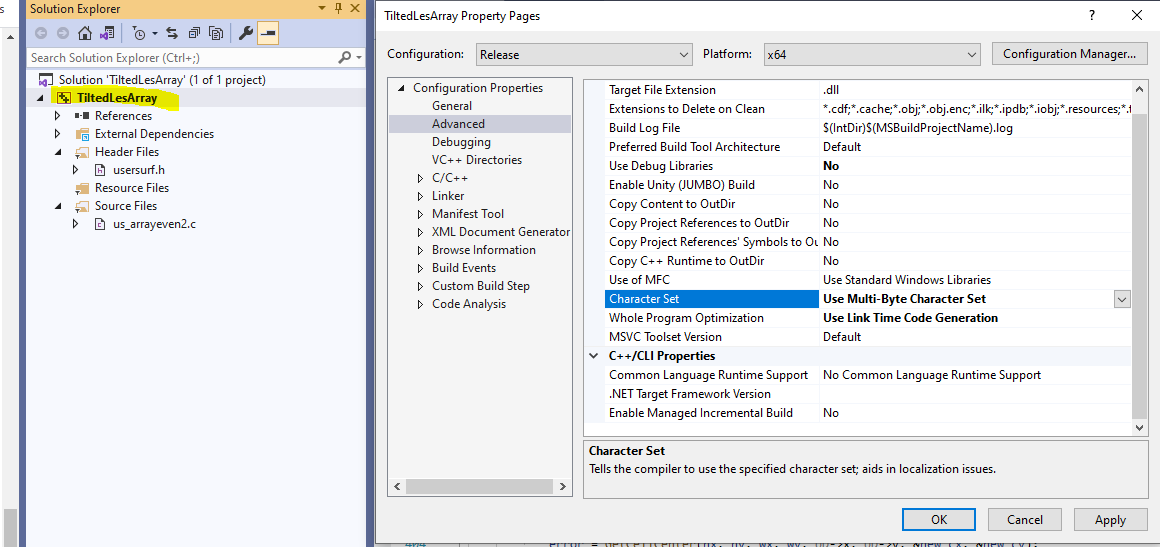
בפעם הקודמת (במפרולייט) זה נראה ככה:

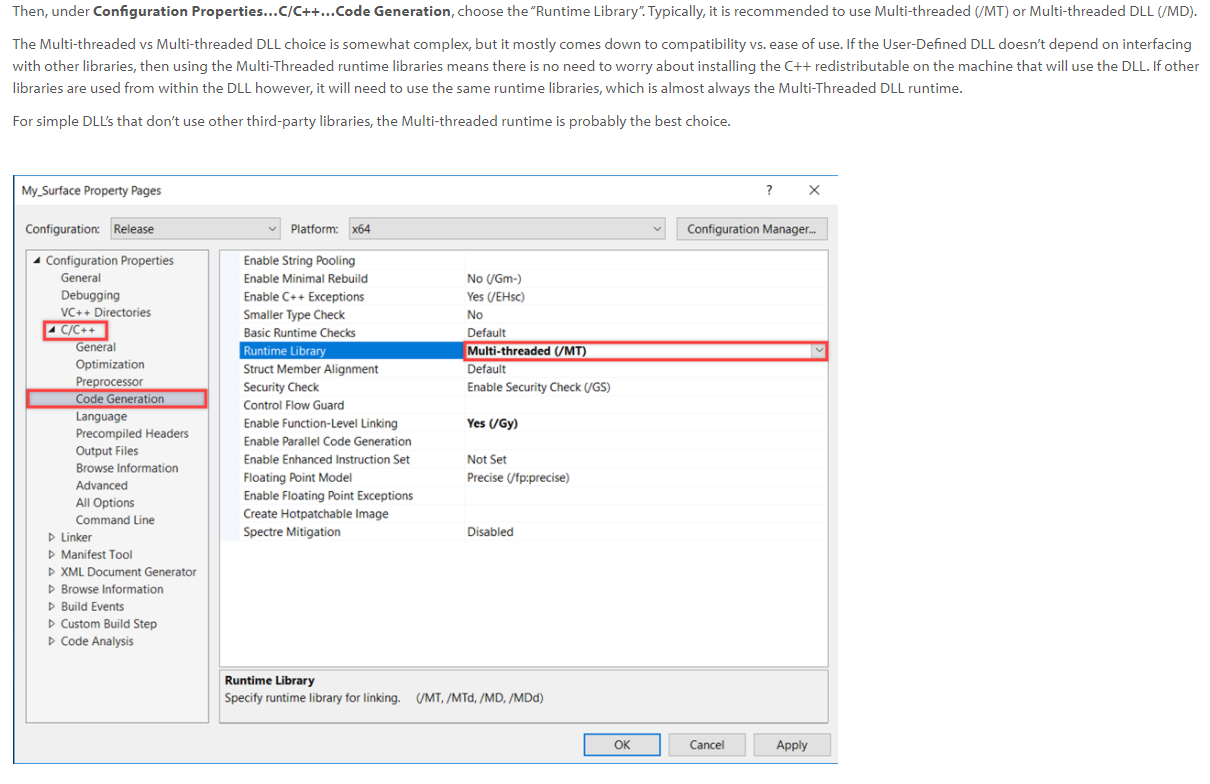


משום מה, נראה שצריך לחזור על הפעולה הזאת עוד פעם, כלהלן:



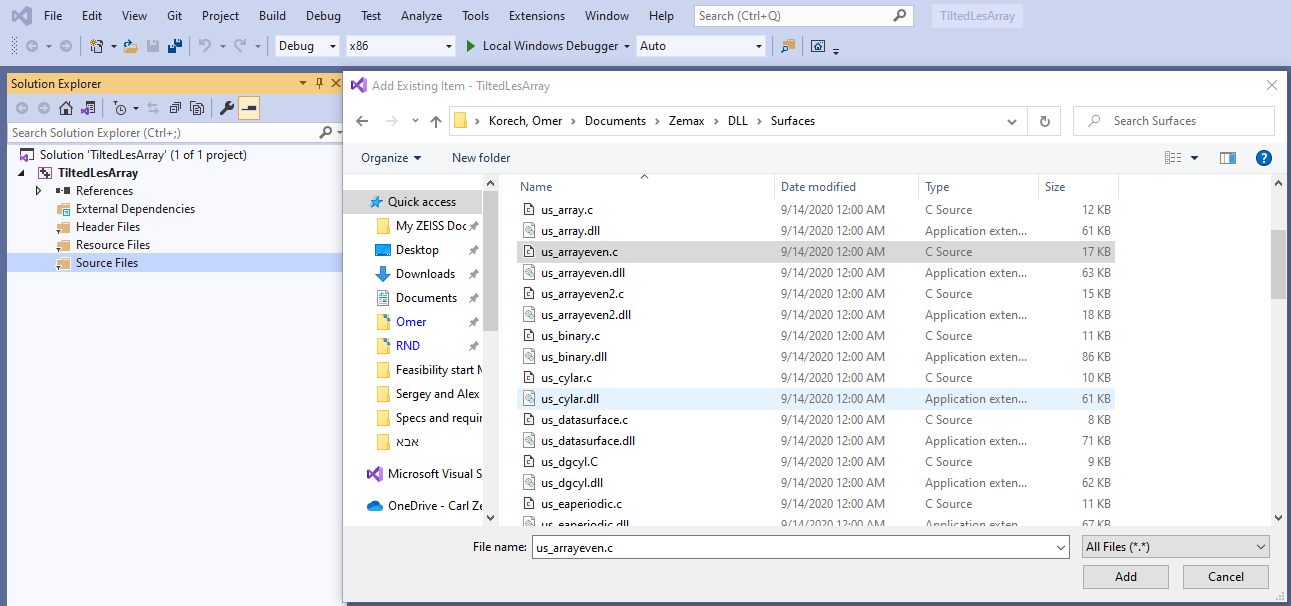
כפתור ימני על שם הפרוייקט (מודגש בצהוב להלן) וקינפוג כדלהלן:





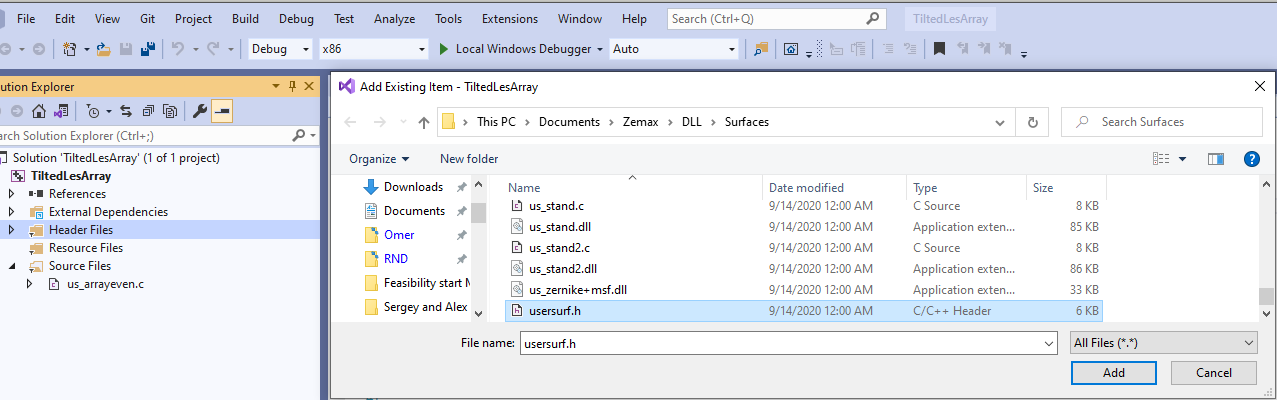
1. טעינת הקוד של המשטח המקורי של זימקס

כפתור ימני על source files  
Add 🡪 Existing file

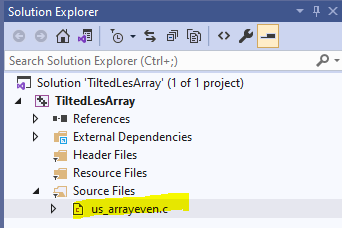


באופן דומה:  
כפתור ימני על Header files  
Add 🡪 Existing item

Usersurf.h



Double click על הקובץ שהתווסף:



בניגוד להנחיות באתר של זימקס, זאת טעות (שלא מאפשרת קמפול) להוסיף בתחילת הקוד c את השורות הבאות:

#ifdef \_\_cplusplus

extern "C"

{

#endif

int \_\_declspec(dllexport) APIENTRY UserDefinedSurface(USER\_DATA\* UD, FIXED\_DATA\* FD);

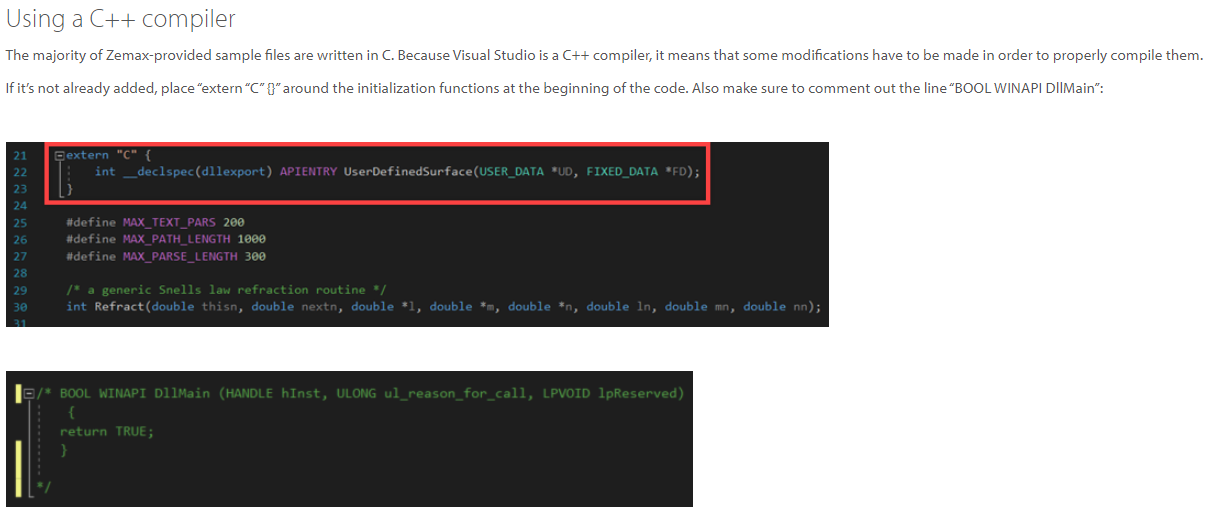
#ifdef \_\_cplusplus

}

#endif

extern "C" { guards are only required (or understood) by C++. You can omit them from a C file, should include them in a C++ file and should guard them with a \_\_cplusplus ifdef in a header file.

אבל כן הורדתי את DllMain בהתאם להנחיות באתר של זימקס



1. שינוי קוד המקור

מסמך זה מיועד למתכנן אופטי ומקנה ידע לשינויים קלים של הקוד.  
שינוי הקוד עבור הפרוייקט tilted MLA הצריך ידע ב-c כולל הבנה של הקונספט של pointers והעברת משתנים לפונקציות by reference ולכן חורג מגבולות המסמך.  
אבל בהחלט ניתן לעשות שינויים נוספים בקוד באופן עצמאי, למשל, להגדיל דיוק או מהירות על ידי שינוי הפרמטרים בקוד שלהלן:

if (loop > 5000)

{

// Extending this number 5000 --> more than 5000, may increase precision

if( fabs(t) > 1e-5)//More than 10 nm between beam and surface

{

return(-1);

}

else

{

t = 0.0; // Round the less than 10 nm path difference to 0.

}

}

או לאפשר גם סיבוב סביב ערכי DZ שליליים על ידי שינוי קטע הקוד:

Zr = fabs(FD->param[13]);// Zr is the Z coordinate of the center of rotation. param[13] is Dz obtained from the Zemax LDE