**NOTA TECNICA: Utilizzo di librerie di gestione Polhemus**

**Destinatari: sviluppatori e manutentori software BeautySim**

**06.07.2023**

Il progetto Device.Polhemus (utilizzato da progetto BeautySim2023) carica dinamicamente con DLLImport le dll:

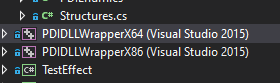
* PDIDLLWrapperX64 e
* PDIDLLWrapperX86

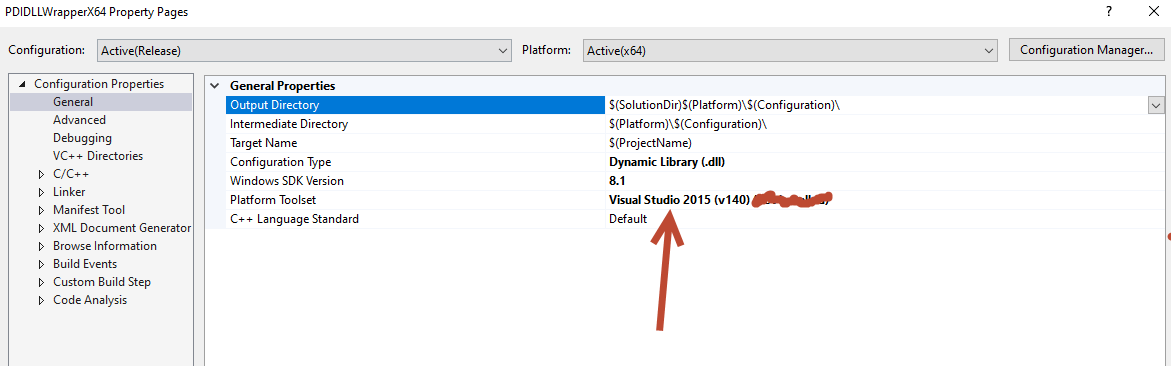
che costituiscono i wrapper .NET alle librerie native c++ di comunicazione con Patriot Polhemus.

Queste dll vengono generate in sottocartelle (un po’ “nascoste”) tramite compilazione dei progetti PDIDLLWrapperX64 e PDIDLLWrapperX86:

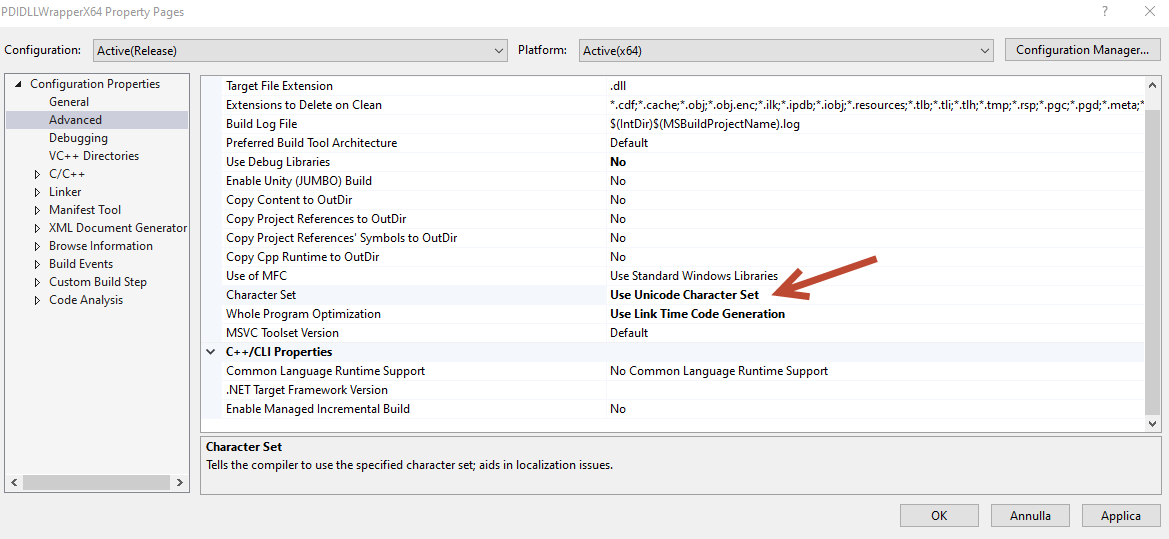
* BlockSim\VStudio\Debug
* BlockSim\VStudio\Release
* BlockSim\VStudio\X64\Debug
* BlockSim\VStudio\X64\Release

Al fine di essere pienamente compatibili con tutte le versioni di windows 10, tali progetti vanno compilati solo ed esclusivamente con profilo Visual Studio 2015, SDK Windows 8.1.





Per compatibilità con lingue non europee, va selezionato anche il character Set Unicode.



**ATTENZIONE! in generale, a runtime, la ricerca dinamica delle librerie runtime (con DLL import) avviene partendo dalla cartella dove risiede l’eseguibile (cartella di deploy applicazione) e poi andando in ricerca sulle altre cartelle dei path di ricerca.**

Per evitare confusioni e facilitare il deploy di BlockSim, le due dll compilate come specificato sopra, sono state spostate nelle stesse cartelle dove, a runtime su PC di destinazione, vengono fatte risiedere anche le librerie native c++, cioè sui path:

* C:\ACCURATE\_DEVICES\G4DLLs\_x64 e
* C:\ACCURATE\_DEVICES\G4DLLs\_x86

Bisogna fare attenzione affinché sulla cartella di deploy applicazione non finisca (per sbaglio) nessuna versione delle due librerie PDIDLLWrapperX64 e PDIDLLWrapperX86 per un qualsivoglia motivo; infatti le due dll devono stare SOLO ED ESCLUSIVAMENTE sui path

* C:\ACCURATE\_DEVICES\G4DLLs\_x64 e
* C:\ACCURATE\_DEVICES\G4DLLs\_x86

che vengono importati nei path di ricerca con la funzione AddDllDevices sul progetto Device.Polhemus.