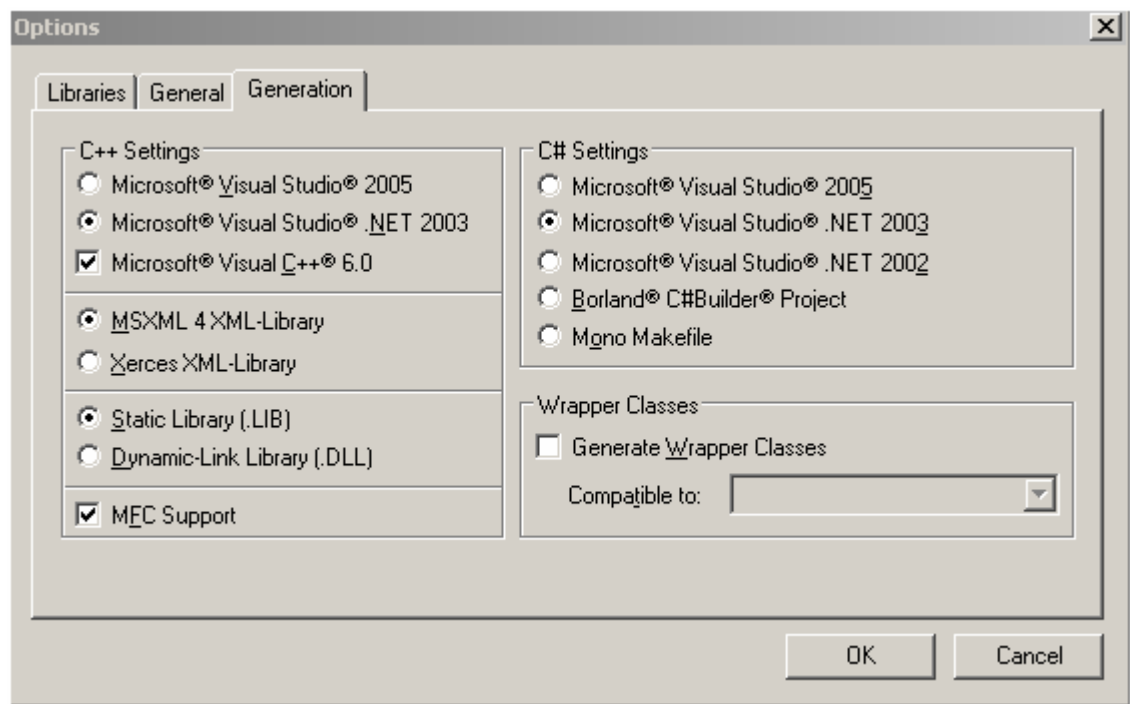


### Section 3: Exporting, Compiling, and Executing Mapforce generated code

Assuming you are using the sample.EIF file and have set up your interchange numbers. (I will be using 01 for FI05 001111111 for F142, T for FI14 and F455, and 08 for FI05\_2 and 002222222 for FI07 for this example) and that you have set up the control number as described in section 2. If you click on the output tab you should get something like this. (Probably with different element, composite, repetition, and segment terminators)

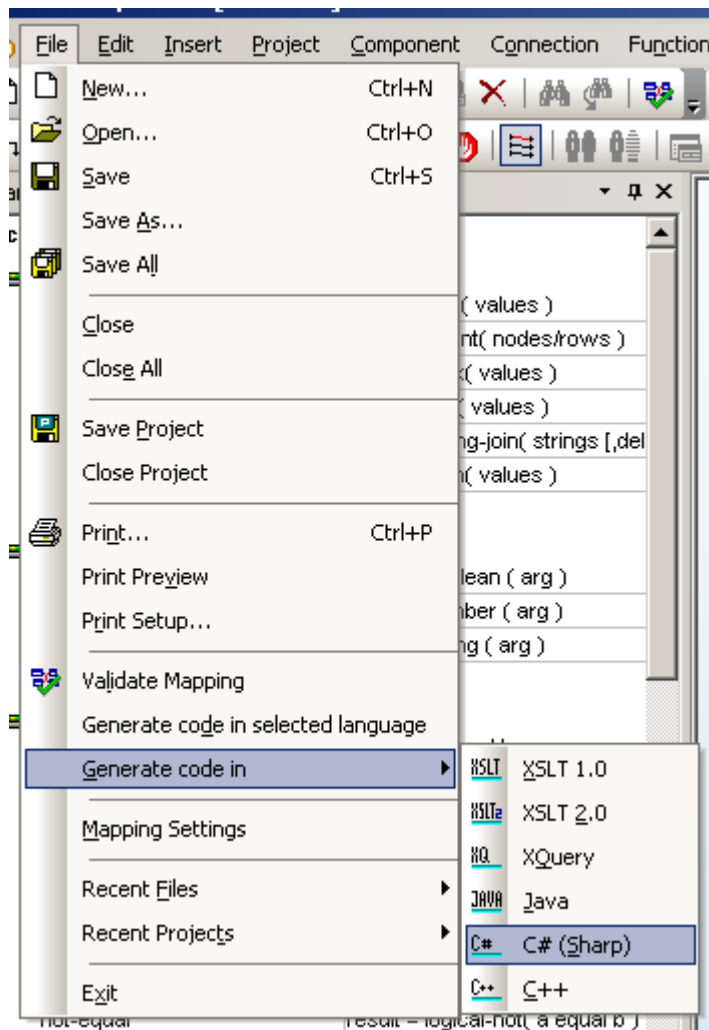
```
ISA*00*      *00*      *01*001111111  *08*002222222  *071126*0947*!*00501*000000053*1*T*>~
GS*****20071126*094703*53*T*004010UCS~
ST*940*  ~
W05~
SE*3*  ~
GE*1*53~
IEA*1*000000053~
```

Ok, lets generate some code. Generating code is very easy. First lets check our code output settings. Go to Tools, Options, and then the Generation Tab.

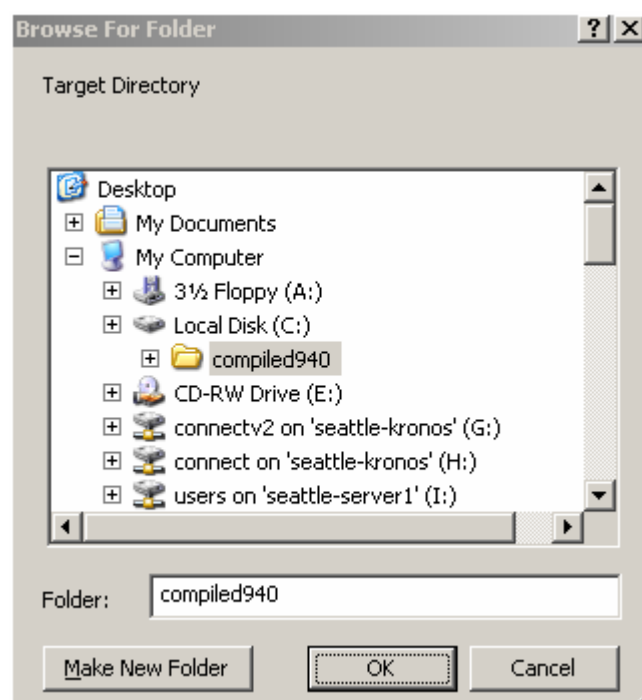


For this guide I'm going to assume the use of Visual Studio .NET 2003, but this is the screen that you would change it to output code for 2005.

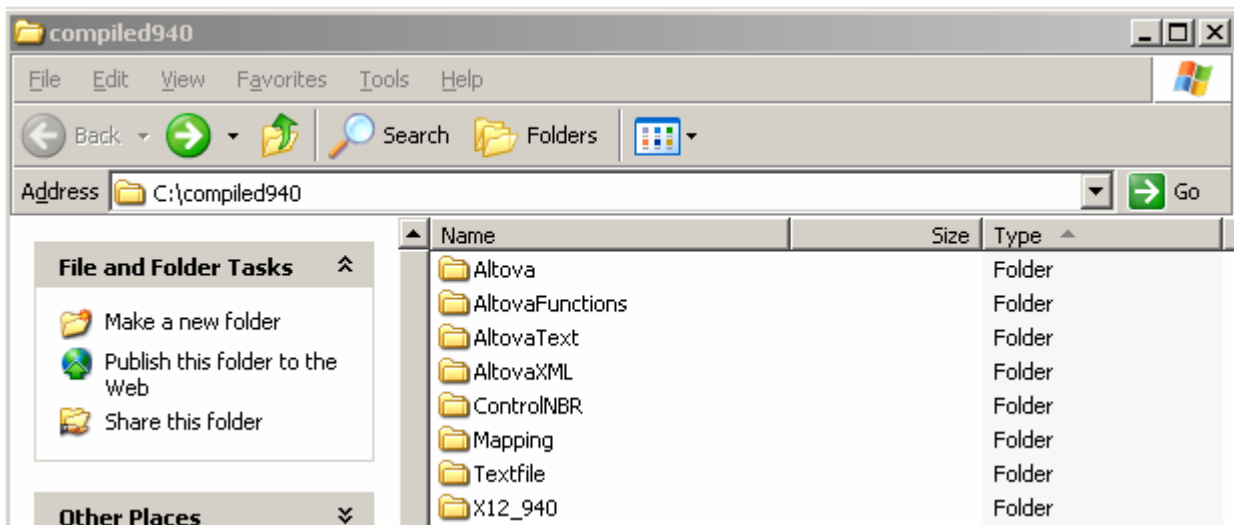
To generate the code using these settings go to File, Generate code in, and then select your language.



For this guide I will use C# code generation. So select C#. Now it wants you to choose your target directory. I created a folder named compiled940. So I will choose that.

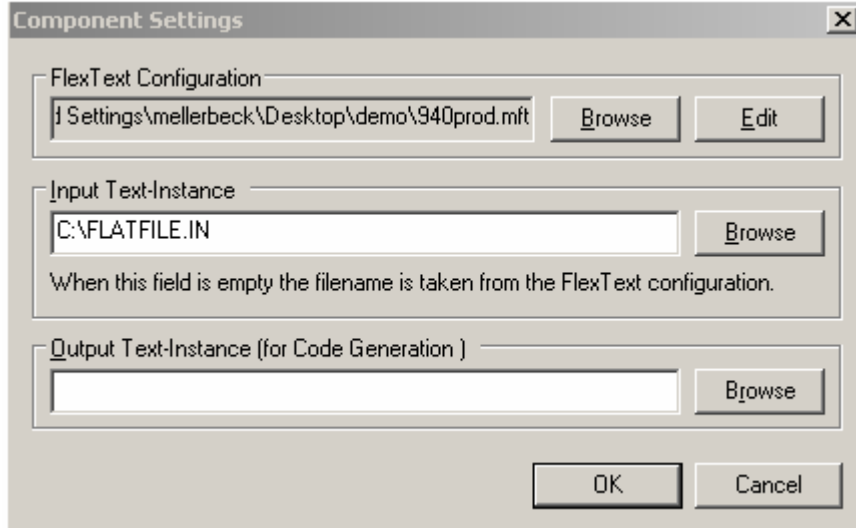


Next open up your file browser and browse to the folder you choose to export the C# code to. You should see something like this.



Open the Mapping folder. And then double click the Mapping.sln which opens the solution in visual studio. Change your build option to release and then fire off a build solution. If all goes well you should have a Build succeeded.

You will need your controlNBR.txt file copied over to the bin\release folder so that it can find it. It is also hard coded to look for the sample.EIF file wherever you have downloaded and loaded it from inside of Mapforce. (The input file can be changed by right clicking on a FlexText, going to properties, and the changing the Input Text-Instance. In my case I renamed it to FLATFILE.IN



And then on the Output Text-Instance (I right clicked the EDI and went to properties) and changed it to C:\OUT.EDI

Ok, lets test the code. I like to fire up a command window and then run the Mapping.exe

If all goes well it should say something like.

\*\*\*\*\*

Saving c:/ControlNBR.txt...

Parsing Text File ...

Saving c:/OUT.EDI

Finished

\*\*\*\*\*

Now if we open up the c:\OUT.EDI it should contain the same data that we saw in our output preview pane.

This would be a good spot to document how to change the code to accept a variable instead of a hard coded file path.