

C# .NET has the following features

- Interoperability - The term interoperability is used to describe the capability of different programs to exchange data via a common set of exchange formats, to read and write the same file formats, and to use the same protocols.
- Multi-language support
- A common run-time engine that is shared by all .NET languages
- Language integration – supports cross language inheritance, exception handling, and debugging.
- A simple deployment model.

Terminology

CIL – Common Intermediate Language – This is a language, much like assembler, which all .NET programs compile to. The CIL is platform agnostic. The C# compiler does not produce machine language for a specific machine. Rather, it produces a binary (*.dll or *.exe) in the *Common Intermediate Language* or CIL.

Assembly – This is a single file that has the binary data for an application. It is written in CIL.

Module – This is a group of assemblies or a multi-file assembly.

Jitter – This is the *Just in time* compiler that changes the CIL into machine specific instructions. The Jitter not only compiles the CIL language, it also saves it in the cache so that if the assembly is asked for again it does not have to be recompiled.

CLR – Common Language Runtime – This is a software package which loads and manages .NET types. It is comparable to the Java Virtual Machine. The CLR contains the Jitter. It locates, loads, and compiles the CIL to machine specific code for execution.

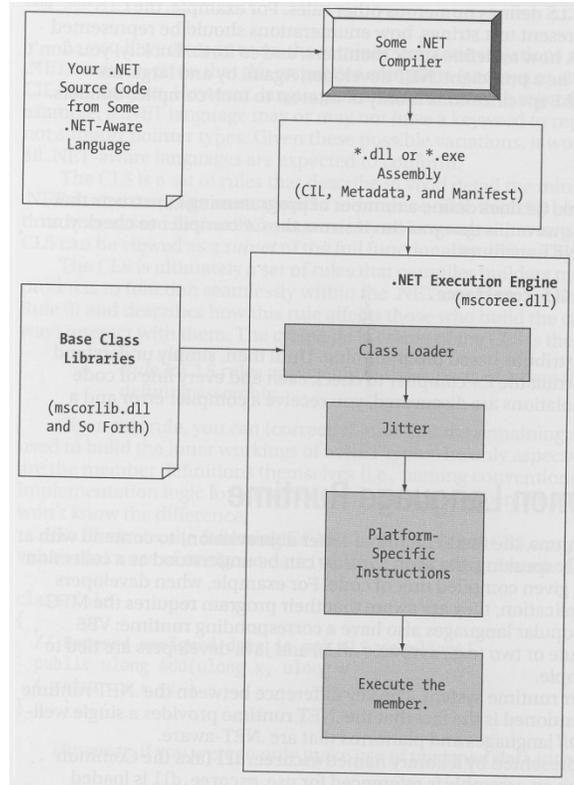


Figure 1
Using the .NET compiler

Class Exercise

1. Open Visual Studio 10. Create a new console project called "HelloMomConsole".

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace HelloMomConsole
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello Mom!");
        }
    }
}
  
```

2. Compile your code and verify that it runs.

3. Close VS 10 and locate the Intermediate Language Disassembler utility called ildasm.exe. On my machine this is located in

C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin

4. Double click on ildasm.exe to run the program.

5. On the file menu navigate to HelloMomConsole project and go to the bin\debug directory to find HelloMomConsole.exe. Open this file to get a screen like that in Figure 2.

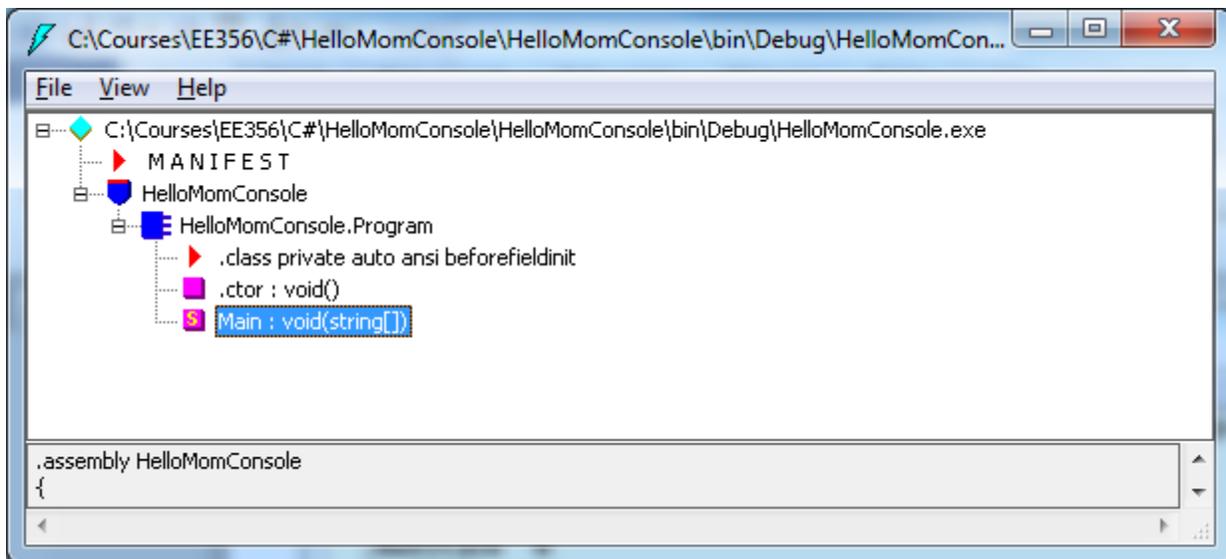


Figure 2

Opening HelloMomConsole.exe using ildasm.exe

Double click on the Main module to see the CIL code.

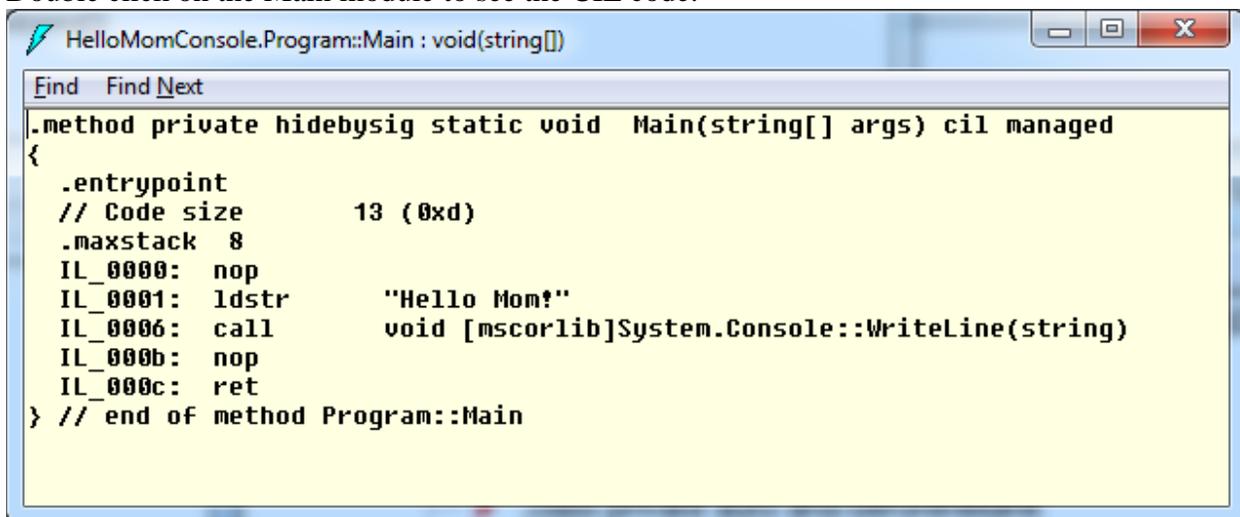


Figure 3

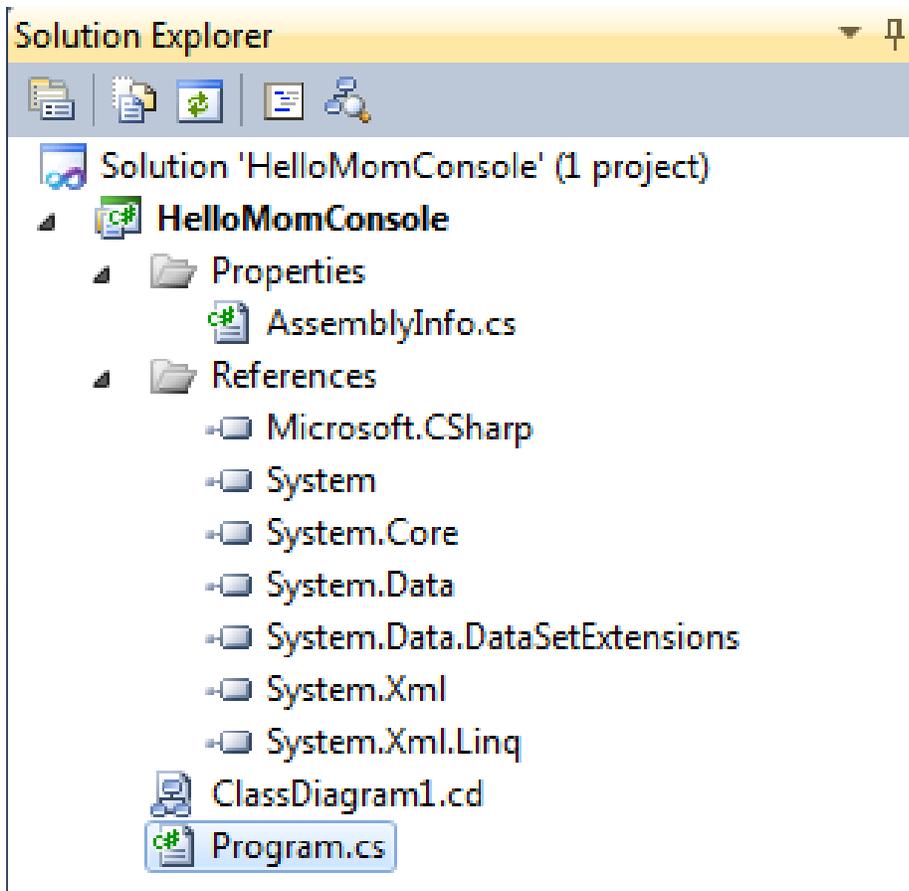
CIL code for the HelloMomConsole application.

There is another tool called reflector which does a better job of disassembly. It is available at www.redgate.com/products/reflector. One of the things reflector does is allow you to get source code back from an exe file. This costs money.

Note that to run any C# program on a given computer, that computer must have the .NET runtime already installed. If it does not, your program will fail. See pp. 36-37 for details of how to install .NET runtime with an application free.

Using Visual Studio 10

Open the console project that you created called "HelloMomConsole".



Notice the Solution is marked Solution 'HelloMomConsole' (1 project)

The project is marked HelloMomConsole

The program for this project is listed as Program.cs

The items under References are assemblies from the library that are included. You can add and delete these as needed.

- Use refactor to extract a method and change the name of a variable.
- Redo HelloMomConsole as a GUI.
- Look at Project 1

EE 356
Project 1 – Dictionary Words

August 24, 2011
Due: September 7, 2011

For this project you should write two programs. Both programs will do the same thing but the first will be a console program and the second will use a GUI with *forms*. The programs will use a word list given in "CrsWdLst.txt" on the web site. The user will provide two inputs. The first will always be a string of letters that we are interested in finding in the word list. The second will indicate the length of the words that we want to search. The format of the second entry can be:

- a single asterisk,
- a number from 1 to 25,
- two numbers separated by a hyphen where the second number greater than the first. The first number can range 1 to 24 and the second can range 2 to 25.

Some sample inputs for the console program might look like this:

```
abc, 7  
t, *  
in, 3-9
```

For example if I enter

```
abc, 7
```

your program should find the number of times the letter sequence "abc" appears in 7 letter words. If I enter

```
t, *
```

your program should find the number of times the letter t appears in all words. For the entry

```
in, 3-9
```

your program should find the number of times the letter sequence "in" appears in words of length 3, 4, 5, 6, 7, 8, and 9.

Your program must check the validity of the data and issue an appropriate error message if the data is in the wrong format.

Turn in the following electronically to your instructor

1. A Word document titled EE356P1XXX.docx (where XXX is your three initials) which contains:
 - Cover Sheet – The cover sheet should include your name, course number, project number, project title, and date handed in.
 - A short description of your project. Include an overview and any special features that you added that are not in the specifications.
2. A complete code file that can be executed. Your code file must have well documented source code.

Compress these two items into a zip file should have the title EE356P1XXX.zip where XXX are your three initials and email them to your instructor.

To do this project you need to know how to read a file. This console project reads a file and writes it to the screen.

- Create a new console project named FileIn
- In Solution Explorer right click on References and select Add Reference. Select System.Windows.Forms from the list and add this to the references.
- Copy the following code to your project
- Download CrsWdLst.txt from the web site and copy it into your project file under the bin\debug directory. Run the program.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Windows.Forms;

namespace FileIn
{
    class Program
    {
        static void Main(string[] args)
        {
            ReadFile();
        }

        static void ReadFile()
        {
            string s, fileName;
            FileStream fin;
            fileName = "CrsWdLst.txt";
            try
            {
                fin = new FileStream(fileName, FileMode.Open, FileAccess.Read);
            }
            catch(FileNotFoundException exc)
            {
                MessageBox.Show(exc.Message);
                return;
            }
            StreamReader fstrIn = new StreamReader(fin);
            while((s = fstrIn.ReadLine()) != null)
            {
                Console.WriteLine(s);
            }
            fstrIn.Close();
        }
    }
}
```