Lesson 1-9: Use the Error List window

If you've been working with Visual Studio for some time, you've almost certainly encountered the *Error List* window at least once.

In this lesson you'll use the *Error List* window to quickly find and fix errors.

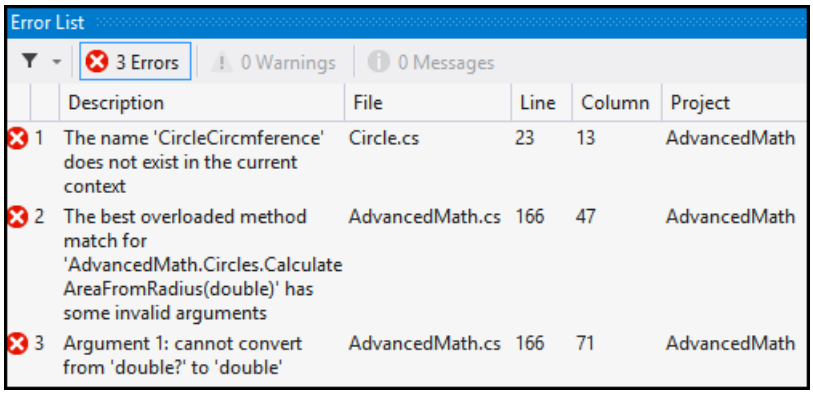
1 Open *AdvancedMath.sln* from your sample files folder.

This is a *Class Library* project, similar to the project that created the *MailModule.dll* file that you added to a project in: *Lesson 1-2: Add references to a project.*

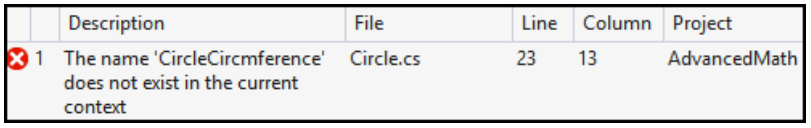
2 Build the project.

Right-click *AdvancedMath* in the *Solution Explorer* and click *Build* from the shortcut menu.

Three errors appear in the *Error List* window at the bottom of the screen. Building a project causes the *Error List* window to display any errors that prevent the project from compiling.



3 Examine the first error in the *Error List* window.



The *Description* field tells you the description of the error. The error mentions a *CircleCircmference* name, which is an obvious misspelling.

The *File* field in the *Error List* window tells you that the error is contained in the *Circle.cs* file.

The *Line* and *Column* fields tell you where the error is located within the file. In this case the error is on line 22 at column 13.

The *Project* field tells you which project contains the error. The *AdvancedMath* project is the only project in this solution.

4 Quickly view the first error.

It wouldn't be difficult to manually find the error, but the *Error List* window is even more helpful.

Simply double-click the first error in the *Error List* window.

Circle.cs is automatically opened with the error highlighted.

```
public Circle(double? Radius, double? Diameter, double? Circumference,
{
    CircleRadius = Radius;
    CircleDiameter = Diameter;
    CircleCircmference = Circumference;
    CircleArea = Area;
}
```

5 Fix the first error.

Change *CircleCircmference* to **CircleCircumference** to fix the problem.

```
public Circle(double? Radius, double? Diameter, double? Circumference,
{
    CircleRadius = Radius;
    CircleDiameter = Diameter;
    CircleCircumference = Circumference;
    CircleArea = Area;
}
```

The error disappears from the *Error List* window.

6 View help for an error.

The *Error List* window allows you to quickly search Microsoft's help files for information about an error.

Right-click the first error in the *Error List* window and click *Show Error Help* from the shortcut menu (see sidebar if the *Error List* window isn't visible).

A web browser window is opened, showing information about the error.

7 Fix the remaining errors.

1. Close your web browser and return to Visual Studio.
2. Double-click the second error in the *Error List* window (*cannot convert from 'double?' to 'double'*).

   This error is the root cause of the other error, so fixing it will fix both problems.
3. Change the underlined code:

   CalculateAreaFromRadius(CircleToPopulate.CircleRadius);

   …to:

   CalculateAreaFromRadius(**(double)***CircleToPopulate.CircleRadius);*

```
CircleToPopulate.CircleArea =
CalculateAreaFromRadius((double)CircleToPopulate.CircleRadius);
```

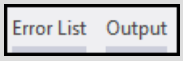
8 Build the project.

This time the project builds successfully and all errors disappear from the *Error List* window.

note

If the Error List window doesn't appear

If the Error List window disappears from view, you can easily bring it back by clicking the *Error List* icon in the bottom-left corner of the screen.



note

The cast conversion method

You fix an error in this lesson by using the *cast* conversion method to make C# recognize a value as a *double* type instead of a nullable *double?* type.

The *cast* conversion method is covered in Session 5 of the Essential Skills course in this series.