Chapter 28: Option Strict

Section 28.1: Why Use It?

`Option Strict On` prevents three things from happening:

1. Implicit Narrowing Conversion Errors

It prevents you from assigning to a variable that has *less precision or smaller capacity* (a narrowing conversion) without an explicit cast. Doing so would result in data loss.

```
Dim d As Double = 123.4
Dim s As Single = d 'This line does not compile with Option Strict On
```

2. Late Binding Calls

Late binding is not allowed. This is to prevent typos that would compile, but fail at runtime

```
Dim obj As New Object
obj.Foo 'This line does not compile with Option Strict On
```

3. Implicit Object Type Errors

This prevents variable being inferred as an Object when in fact they should have been declared as a type

```
Dim something = Nothing. 'This line does not compile with Option Strict On
```

Conclusion

Unless you need to do late binding, you should always have `Option Strict On` as it will cause the mentioned errors to generate compile time errors instead of runtime exceptions.

If you *have* to do late binding, you can *either*

- Wrap all your late binding calls into one class/module and use `Option Strict Off` at the top of the code file (this is the preferred method as it reduces the likelihood of a typos in other files), *or*
- Specify that Late Binding does not cause a compilation failure (`Project Properties > Compile Tab > Warning Configuration`)

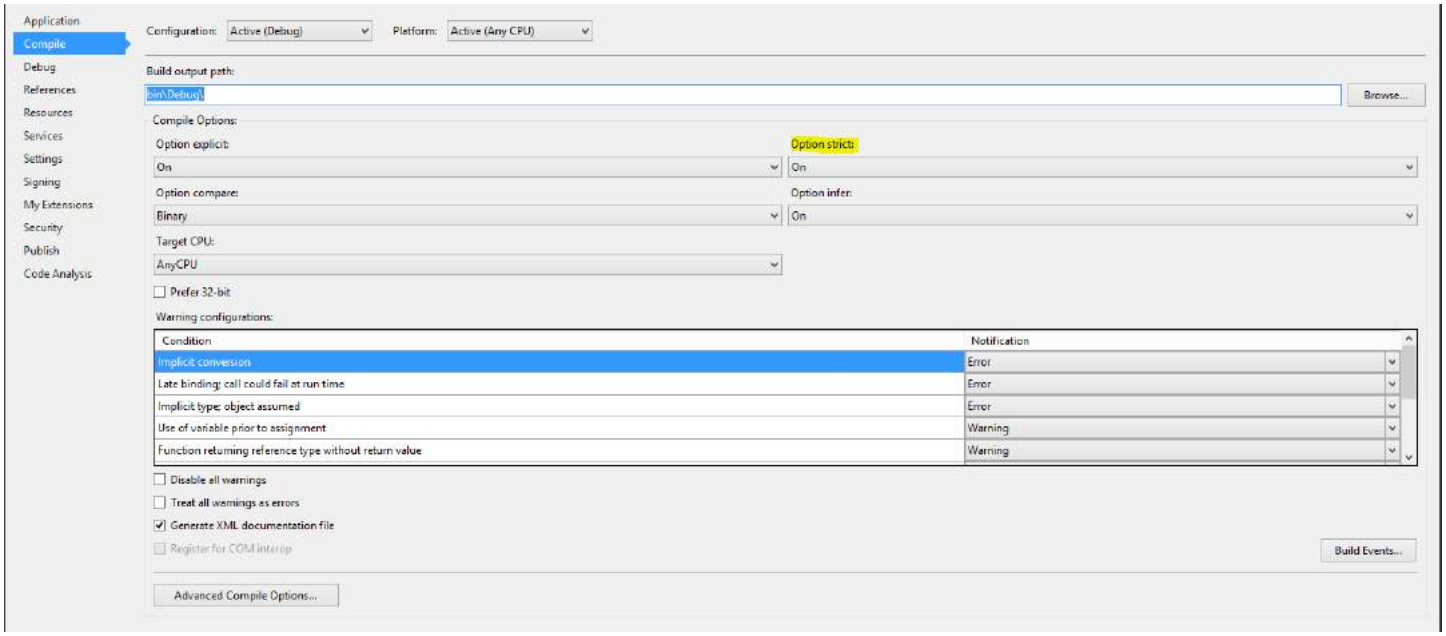
Section 28.2: How to Switch It On

- You can switch it On at the Module/Class Level by placing the directive at the top of the code file.

  ```
  Option Strict On
  ```

- You can switch it on at the project level via the menu in Visual Studio

> Project > [Project] Properties > Compile Tab > Option Strict > On



- You can switch it On by default for all new Projects by selecting:

> Tools > Options > Projects and Solutions > VB defaults > Option Strict
> Set it to On.