VS Project Conditional Constants

When developing cross-platform libraries, you will have the need to enable and disable different parts of your source code based on what statements or functions work on each platform.

For example: Processes are queried differently in Windows than linux.
And, command line interaction is different, as well.

But, we still want to have the same method signatures and calls, for each platform, so that our libraries can be used in both operating systems, with minimal friction.

The easiest way to do this, is to add preprocessor directives (#if).

Here's an example of a declaration that is OS-dependent:

```
#if Windows
        static private System.Threading.Mutex _processmutex;
#elif Linux
        FileStream? _processmutex;
#endif
```

When compiled for Windows, the above declares a Mutex as the process mutex.
But, when compiled for Linux, the above declares a FileStream instance as the process mutex.

The compiler achieves the conditional compilation based on recognizing the constants 'Windows' or 'Linux'.

These constants are set, based on, if the project is set to compile as: DebugWin or DebugLinux.

We use a special PropertyGroups, each with a condition that recognizes the compiler output, and defines the required constants.

The property groups look like this:

```
  <PropertyGroup Condition="$([System.String]::Copy($(Configuration)).EndsWith('Win'))">
    <DefineConstants>$(DefineConstants);Windows;NET6</DefineConstants>
  </PropertyGroup>
```

```
<PropertyGroup Condition="$([System.String]::Copy($(Configuration)).EndsWith('Linux'))">
  <DefineConstants>$(DefineConstants);Linux;NET6</DefineConstants>
</PropertyGroup>

<PropertyGroup Condition="$([System.String]::Copy($(Configuration)).EndsWith('OSX'))">
  <DefineConstants>$(DefineConstants);OSX;NET6</DefineConstants>
</PropertyGroup>
```

To correctly enable one of these, you need to update your project configurations from the default: Debug;Release.

To the new:

- DebugWin
- DebugLinux
- ReleaseWin
- ReleaseLinux

This can be done by setting the Configurations property in the main PropertyGroup of your project, like this:

```
  <Configurations>DebugWin;ReleaseWin;DebugLinux;ReleaseLinux</Configurations>
```

Once done, selecting a configuration that ends in 'Win', will compile the Windows sections of source.
And, selecting a configuration that ends in 'Linux', will compile the Linux sections of source.

Revision #1
Created 11 August 2025 00:16:02 by glwhite
Updated 11 August 2025 00:33:22 by glwhite