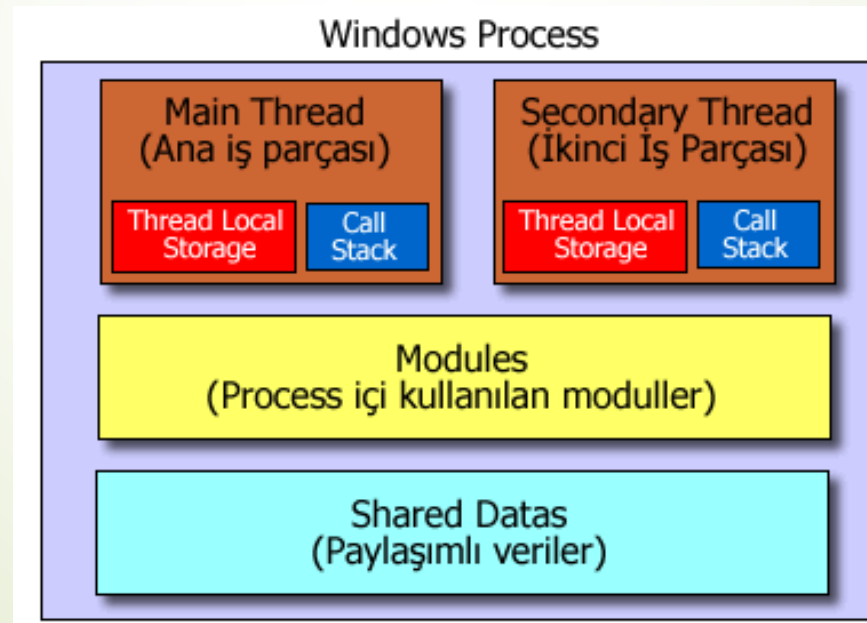


# C# ve .Net Process

- Process.Start metodu uygulamamızda istenildiğinde başka uygulamaları (kullanılan bilgisayarda olan uygulamaları) açmamıza olanak sağlayan bir metottur.
- Bu metod ile pratik bir şekilde Excel, Word ve Internet Explorer gibi uygulamaları açmamız mümkün olmaktadır.
- Bir Windows uygulamasını çalıştırdığımızda işletim sistemi bellek üzerinde söz konusu programın çalışabilmesi için bir **Process** açar. Bu Process içerisinde, uygulamanın çalışması için gerekli bellek ayırma işlemleri, harici olarak kullanılan Module' ler (örneğin başka Com nesneleri veya .net assembly' ları gibi) ve process içi Thread' ler bulunur.

# C# ve .Net Process

- .Net Framework aşağıdaki şekilde görülen yapıdaki bilgileri çalışma zamanında elde edebilmemizi sağlayan bazı tipler (types) içerir. **System.Diagnostics** isim alanında bulunan bu tipler yardımıyla çalışma zamanda bir Process' e ait çeşitli bilgilerini elde edebiliriz. Diğer taraftan dilersek çalışma zamanında başka Process' leri çalıştırabilir, var olanları yok edebilir, bazı Process' lere parametrik bilgi aktarabilir.



## Process sınıfını kullanarak neler yapabiliriz;

Gerçekleştireceğimiz örnekler basitten daha işlevsel ve kapsamlı örneklerle doğru gidecek şekilde hazırlanacak.

**Process** sınıfı ile gerçekleştireceğimiz ilk işlem **static** tek parametrelili **overload** edilmiş **Start** metodunu kullanarak klasör, dosya açma ve web adreslerini tarayıcılarda bu metod ile nasıl açacağımızı göreceğiz.

Process sınıfını kullanabilmek için **System.Diagnostics** namespace'ini uygulamanıza dahil etmelisiniz

## Dizini Açma

Şimdi üstte bahsettiğimiz Start metodunu kullanarak C:\ dizinini Windows Gezgini ile açalım.

```
using System.Diagnostics;

class Program
{
    static void Main()
    {
        // Process.Start metodunu burada kullanıyoruz.
        Process.Start("C:\\");
    }
}
```

## Dosyayı Açma

ornek.txt yerine C:\ornek.txt gibi yolu kesin olarakta belirtebilirsiniz.

```
using System.Diagnostics;

class Program
{
    static void Main()
    {
        // "ornek.txt" adındaki dosya açılacak.
        // ... Konsol uygulamasının exe uzantısının olduğu yerde
        arayacaktır dosyayı.
        Process.Start("ornek.txt");
    }
}
```

## Web Sitesini Açma

Üstteki işlemlerden sonra aynı metodu kullanarak bir web sitesini açma işlemini gerçekleştireceğiz dosyayı açma kısmında düştüğümüz not'da olduğu gibi açılacak olan site sistemde varsayılan olan tarayıcı ile açılacaktır.

```
using System.Diagnostics;

class Program
{
    static void Main()
    {
        // Google'da aranacak terimi metoda geçiyoruz.
        GoogleArama("Google.com");
    }

    static void GoogleArama(string t)
    {
        Process.Start("http://google.com/search?q=" + t);
    }
}
```

## Word dosyası açma

Kullanıcının seçeceği Word dosyasını winword.exe kullanılarak açılacaktır. Açılacak dosya parametre olarak processe gidecektir.

Process'te parametre kullanımı için **ProcessStartInfo** sınıfının **FileName,Arguments** parametrelerini kullanarak açılacak dosya ismi process'e gönderilebilir. OpenFileDialog ile dosya okunup fonksiyona gönderilip, fonksiyonda process ayarları yapıp dosya açılır.

```
public void WordDosyasiAc(string dosyaadi)
{
    ProcessStartInfo pStart = new ProcessStartInfo();
    pStart.FileName = "WINWORD.EXE";
    pStart.Arguments = dosyaadi;
    Process.Start(pStart);
}
```



# C# ve .Net Process

## Kill Metodu ile Process Durdurma

Çalışma zamanında Process' leri başlatabileceğimiz gibi onları yok etmek isteyebiliriz de. Bunun için tek yapmamız gereken Process sınıfına ait **Kill** metodunu kullanmak olacaktır. Process' lerin listesi **GetProcessesByName** metodu ile çekilmiş ve elde edilen dizideki her bir Process tipi nesne örneği için Kill metodu kapatmak

```
Process[] prcs = Process.GetProcessesByName("winword");  
    MessageBox.Show(prcs[0].ToString());  
    for (int i = 0; i < prcs.Length; i++)  
    {  
  
        prcs[i].Kill();  
  
    }
```



## Mevcut uygulama kontrolü ve SendKeys

Çalışan process'i kontrol edip zaten açık ise tekrar açmayacak ve klavye tuşu gönderecek..

KAYDIRMA	+
CTRL	^
ALT	%

```
[DllImport("User32.dll")]
```

```
static extern int SetForegroundWindow(IntPtr point);
```

```
private void button6_Click(object sender, EventArgs e)
```

```
{
```

```
Process p = Process.GetProcessesByName("notepad").FirstOrDefault();
```

```
if(p != null)
```

```
{
```

```
    IntPtr h = p.MainWindowHandle;
```

```
    SetForegroundWindow(h);
```

```
    SendKeys.SendWait("k");
```

```
}
```

```
}
```

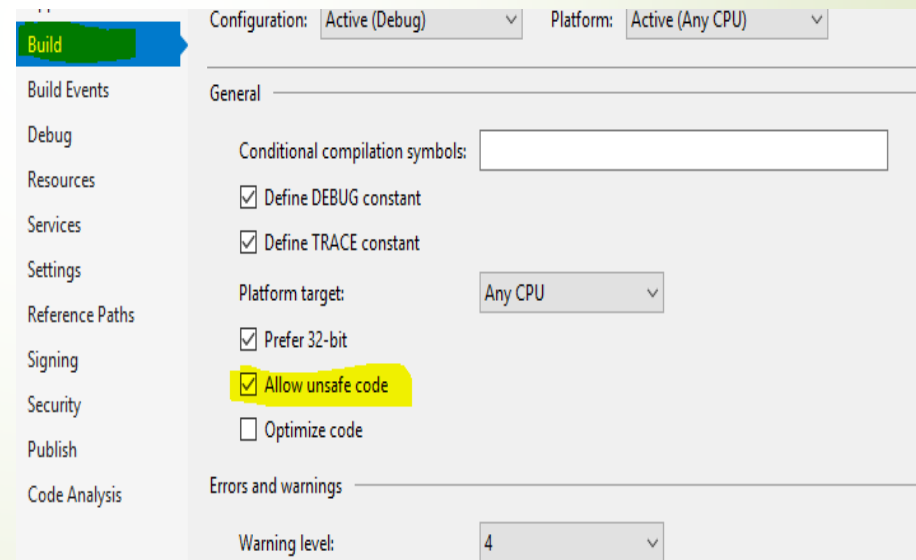
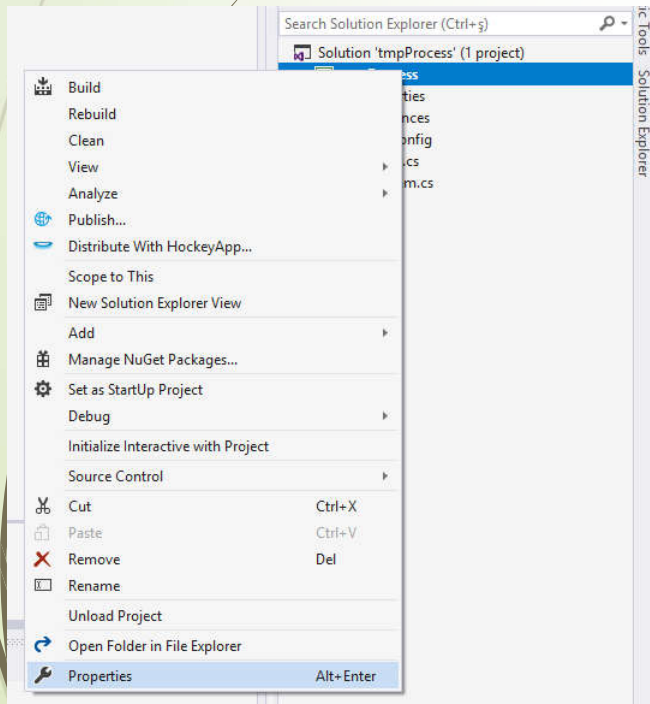
## Waitforexit kullanımı;

Notepad uygulaması açılıyor ardından açılan bu process kapanan kadar kendi uygulamamıza geçmeye müsaade etmez. Waitforexit() metodu çağırılan processin kapatılana kadar bekletmeyi sağlar.

```
private void button5_Click(object sender, EventArgs e)
{
    ProcessStartInfo ps = new ProcessStartInfo();
    ps.FileName = "NOTEPAD.EXE";
    label3.Text = "Çalışmaya başladı";
    Process pro= Process.Start(ps);
    pro.WaitForExit();
    label3.Text = "Kapatıldı!!";
}
```

# C# ve Pointer Kullanımı –unsafe compile

C#'ta gösterici kullanmanın anahtarı olan ***unsafe(emniyetsiz)*** keyword'üdür. C#'ta gösterici kullanacaksak eğer ilgili kod bloğunu unsafe ile işaretlenmiş bir yapı içerisine almamız gerekmektedir. Bu yapı bir sınıf, metot yahut direkt olarak unsafe yapısı olabilir. Eğer ki unsafe keywordü kullanmaksızın gösterici kullanmaya kalkarsak programımız derlenmeyecektir. Unsafe derleme için Proje / Özellikleri / Build başlığında "allow unsafe code" seçilir.



# C# ve Pointer Kullanımı –unsafe compile

Bir değişkenin pointer olarak atanıp, işaretçi adresini , işaretçinin gösterdiği yerdeki değeri yazdıran fonksiyon örneği aşağıdadır.

```
public unsafe void PointerTest()
{
    int x = 100; // pointer değişken oluştur

    int *isaretci = &x; // x adresi pointera atanıyor

    textBox2.Text = ((int)isaretci).ToString(); // isaretcinin
    kendi bellek adresi

    textBox3.Text = (*isaretci).ToString(); // işaretçinin
    gösterdiği yerdeki değeri
}
```