

Yapımcı

Yunus Emre AK ~ YEmreAk.com

Son Güncelleme

10.09.2020




İçerikler

- İlk Projeyi Oluşturma
 - Ön Gereksinimler
 - Çalışma Yolu
 - VSIX Yapımına Hazırlanma
 - Proje İsmi Güncelleme
 - Manifest Ayarları
 - Hedef Uygulamaları ve Sürümleri Değiştirme
 - VSIX için ikon ekleme
 - İlk Komut Butonunu Ekleme
- Eklenti Yönetimi
 - .vsct Dosyası Nedir
 - GUID Değerlerini Tanımlama
 - Komut Ekleme
 - Buton Ekleme
 - Kaynak Koda Bağlama
- IDE Objelerine Erişme
 - Visual Studio Ortam Objelerine Erişme
 - ProjectItem
 - FileCodeModel
 - CodeElement
- Kod Örnekleri
 - Aktif Dökümanın İçeriğini Alma
 - Editör Üzerindeki Seçili Metni Sıralama
 - Editör Üzerindeki Seçili Metnin içerisindeki Method İçeriğini Sıralama
 - Aktif Dokümandaki Üretilen Kodları Sıralama
 - İki Dosya Arasındaki Farklılıkları Gösterme
 - Git Komutu Çalıştırma
 - Proje dizinini ve dosya yolunu alma
 - Git process oluşturma ve okuma
 - Dosyadan FileCodeModel Oluşturma
 - Designer Dosyasını Önceki Sürümü ile Sıralanarak Karşılaştırılması
- VSIX için Dialog Box Oluşturma
 - Bağımlılıkları Tanımlama
 - Dialog penceresini oluşturma
 - OnClick ve OnCompare eylemleri tanımlama
 - OnCompare Butonu Algoritmasını Oluşturma

İlk Projeyi Oluřturma

Ön Gereksinimler

Eklenti Visual Studio için C# ile programlamayacağımızdan dolayı:

-  **Visual Studio** ve **Visual Studio extension development** iş yükü indirilmelidir
-  Eklenti için derinden bir C# bilgisi yerine hızlı bir öğrenmeye odaklanılması kafidir
-  Hızlıca C# öğrenmek için [CSharp Quick Guide](#) sayfasına bakmalısın
-  Yazım standartları için [CSharp Coding Standarts](#) alanına da bakabilirsin







 C# Hakkında bilgi için [C# Quick Start](#) pdf notlarımı da inceleyebilirsin



Visual Studio extension development

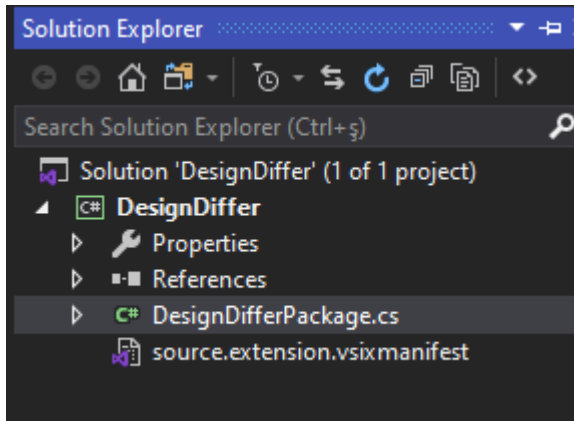
Create add-ons and extensions for Visual Studio, including new commands, code analyzers and tool windows.

Çalışma Yolu

-  Eklenti programlanır, derlenir ve çalıştırılır
-  GitHub üzerinden versiyon kontrol sistemi ile ilerleme kontrol edilir
-  Derlenme ve test işlemleri Continuous Integration (CI) ile otomatikleştirilir
-  Herkese açık stabil bir sürümü üzerinden eklenti yayınlanır
 -  Tabi bu sadece açık kaynak olmasını isterseniz yapılır
-  VS Gallery üzerine eklenti aktarılır

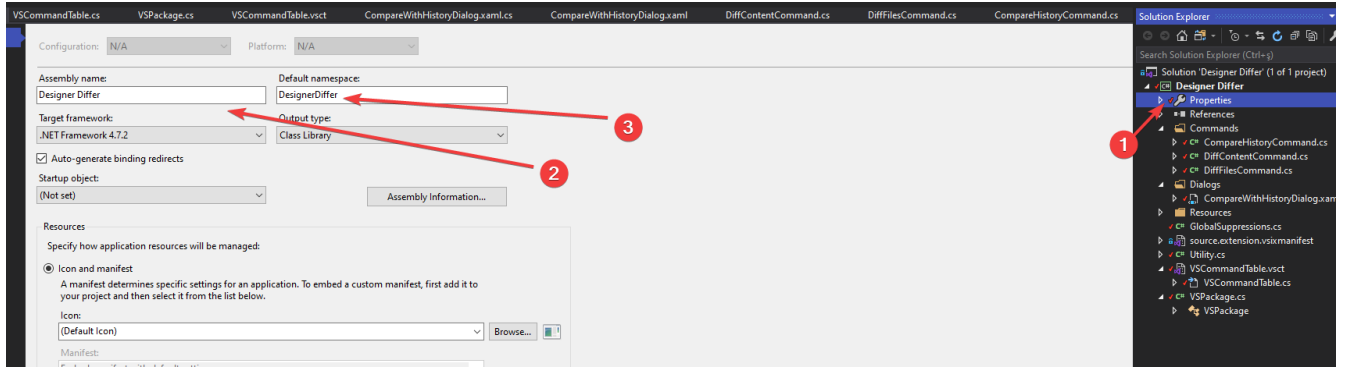
VSIX Yapımına Hazırlanma

-  **Create New Project - VSIX Project** alanından proje oluşturulur



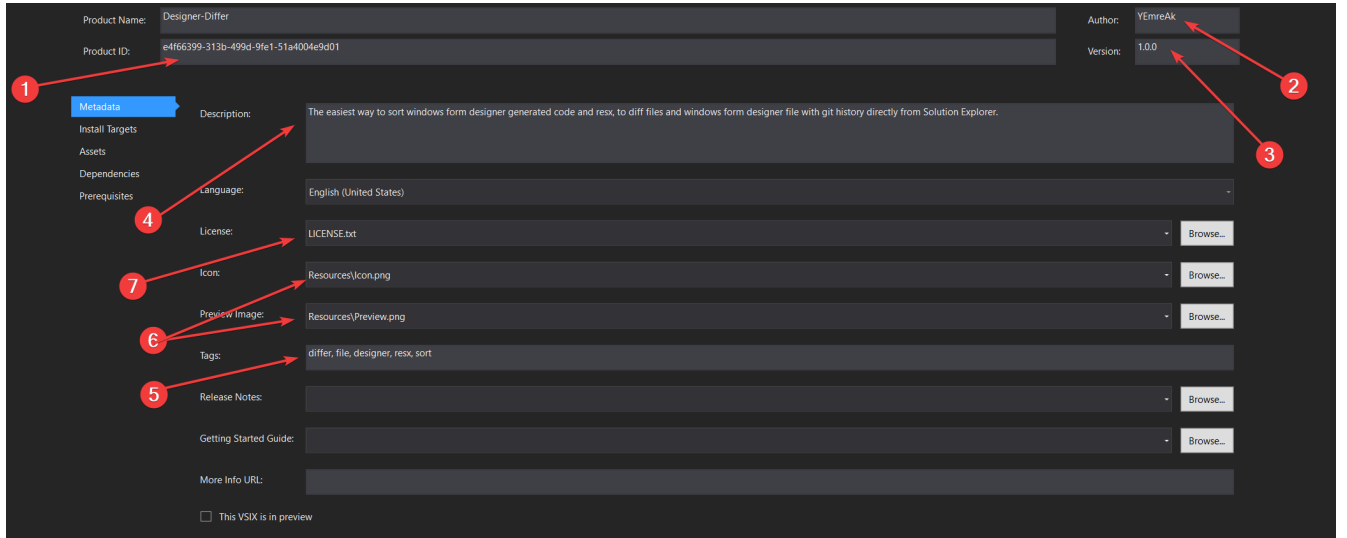
Proje İsmi Güncelleme

- Proje isminden vazgeçmeniz veya ismini yanlış belirtmeniz durumunda projenin adını güncellemeniz gerekebilir
- Solution Explorer** üzerinden **Properties** alanından güncellenir



Manifest Ayarları

- *.vsix.manifest** dosyası manifest dosyasıdır ve projenin ayarlarını içerir
- Product ID** kısmında **<Projeİsmi>.<GUID>** şeklinde olan değerden **Projeİsmi** silinir
- Author** alanında eklentiye geliştiren ekibin ve kişinin adı bulunur
- Version** kısmı **Semantic Versioning** tarzı ile uygulama sürüm bilgisini tutar
- Description** alanına eklentinin ne işe yaradığına dair bilgi verilir
- Tags** kısmında eklentinin hitap ettiği konulara **<konu>, <konu2>, ...** şeklinde yer verilir, aramalarda bu şekilde çıkar
- Preview Image** 175x175 ve **Icon** ise 90x90 boyutunda eklentinin ikon resmini barındırır
- License** alanına kendi lisans dosyanızın yolunu eklersiniz (örn. [Apache License 2](#))



Hedef Uygulamaları ve Sürümleri Deęiřtirme

- **Install Targets** alanından hangi sürümler üzerinde kullanabileceęinin tanımlaması yapılır
- **Identifier** Visual Studio IDE türünün seçildięi alandır
- **Version Range** ise seçilen türde hangi sürümlere kadar desteklendięini belirtir
- **Dependencies** kısmı eklentinin kullandığı veya baęlı olduęu framework veya kütüphanelerin belirtildięi kısımdır
- **Prerequisites** ile gereksinimleri ve sürümlerini tanımlarız

Metadata

Install Targets

Assets

Dependencies

Prerequisites

Type of install: Visual Studio Extension

Identifier	Version Range
Microsoft.VisualStudio.Community	[12.0, 17.0)
Microsoft.VisualStudio.Enterprise	[12.0,17.0)
Microsoft.VisualStudio.Pro	[12.0,17.0)

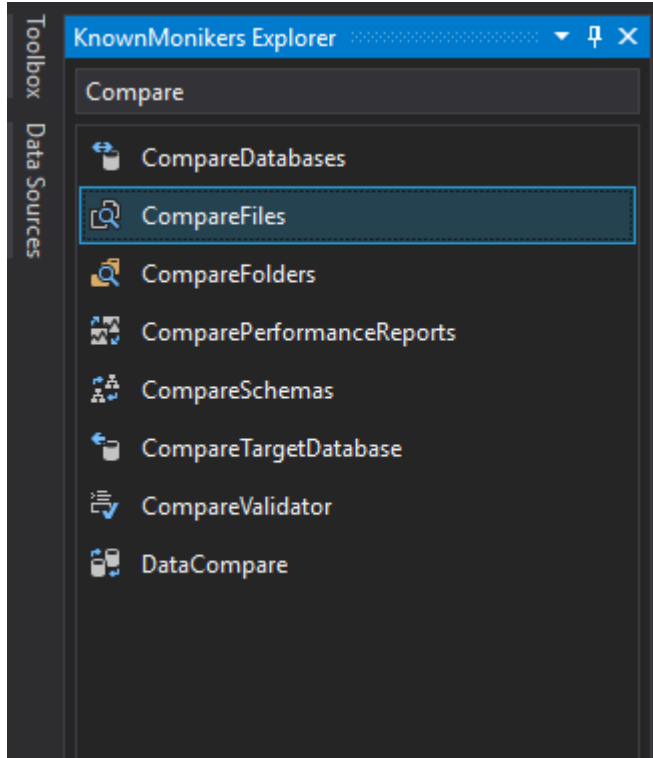
☐ This VSIX is installed for all users (requires elevation on install)

☐ This VSIX is installed by Windows Installer

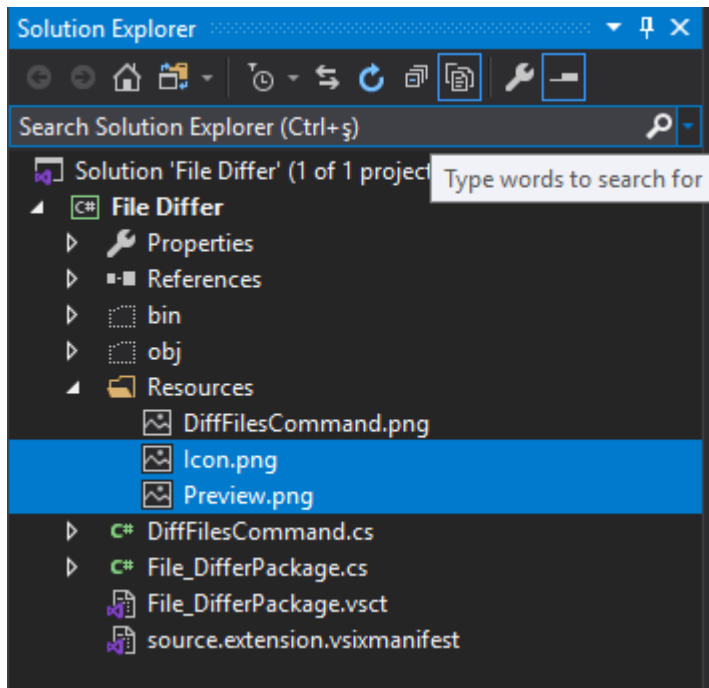
Detaylı bilgi için [Visual Studio extensions and version ranges demystified](#) alanına bakabilirsiniz.


VSIX için ikon ekleme

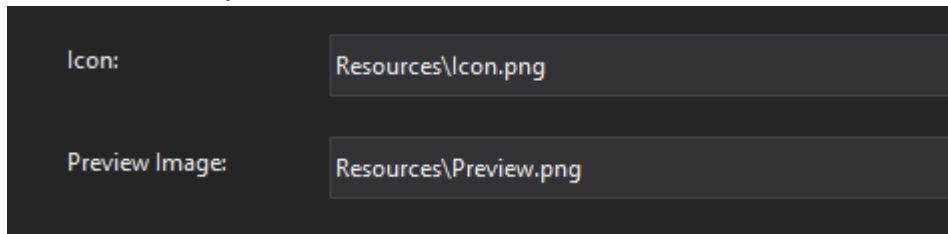
- ✨ PNG dışındaki formatları da destekler ama PNG kullan
- 📁 VSIX'de 3000 icon vardır bunları kullanabilmek için [Extensibility Essentials 2019](#) eklentisini indir
- ⚙ View -> Other Windows -> KnownMonikers



- 📄 Çıkan panelde seçilen iconu Resource içerisine alttak özelliklerle eklemeliyiz:
 - 16 width ile *Command.png icon dosyasını overwrite ederek
 - 175 width ile Preview isimle
 - 90 width ile Icon isimle
- 🗑 *.vsct dosyası içerisinde **silmen gereken** kısımlar
 - Bitmap alanında usedList kısmındaki değerlerden ilki hariç diğerlerini
 - GuidSymbol alanındaki IDSymbol satırlarından ilki hariç diğerlerini
- + Son eklenen resimleri projeye dahil etmek için **Solution Explorer** alanında sağdan 3. ikon **Show all files** ile resimleri bulup, onları seçip, sağ tıklayıp **Include From Project** demeliyiz

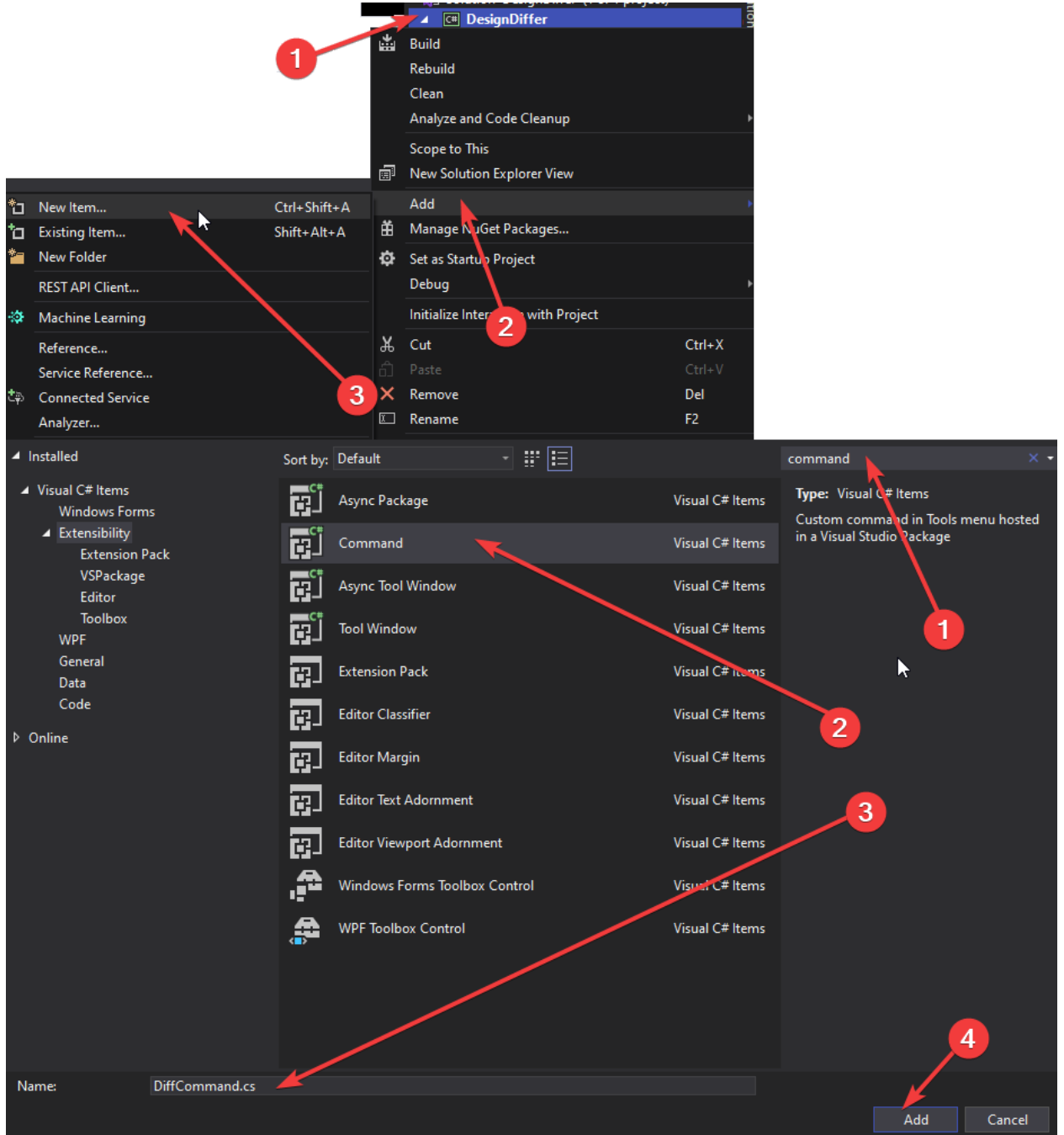


-  *.vsixmanifest dosyasına ikon ve ön izleme resmi eklenmeli

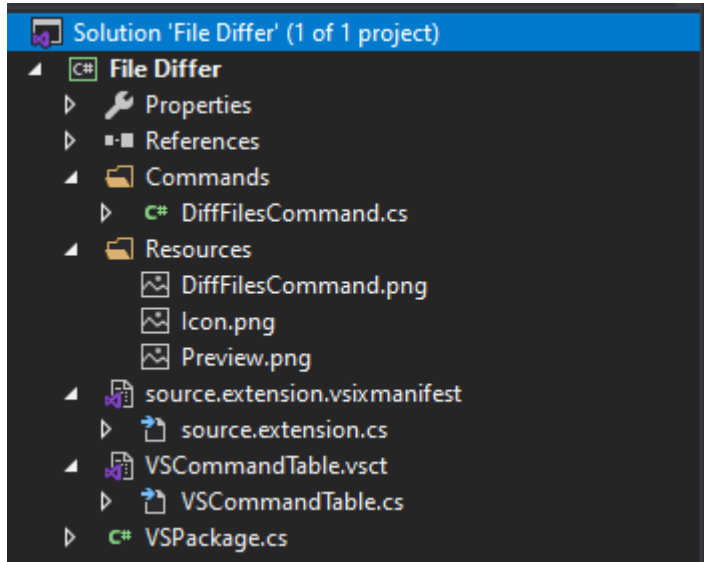


İlk Komut Butonunu Ekleme

- + **Solution Explorer** üzerinden projeye sağ tıklayıp **Add -> New Item-> Command** ile ilk komutumuzu ekliyoruz



- Yukarıdaki işlem ile projeye eklenen dosyalar
 - <dosya ismi>.cs ve
 - <proje ismi>.vsct ile butonları ve diğer fonksiyonelliklerin tanımlandığı dosya
 - **Resources** dizini içerisine eklenti ile ilgili görsel objeler
- İlerleyen konularda da yöneleceğimiz ve önerilen proje yapısı aşağıdaki gibidir



Eklenti Yönetimi

.vsct Dosyası Nedir

- Proje içerisindeki komutlar, resimler ve benzeri her bileşen bilgisi burada saklanır
- Tüm bileşenler birbirlerine ve kaynak kodlara guid değerleri ile bağlanır
- Proje içerisinde tek bir vsct dosyası bulunur

GUID Değerlerini Tanımlama

- Bileşen arasındaki bağlantılar için değişkenleri tanımlayan guid değerleri kullanılır
- Her guid değeri eşsiz olmak zorundadır
- **Symbols** alanı içerisinde ilk başta paket guid değeri tanımlanır
- **GuidSymbol** ile paket içerisindeki belirli guidler için alt değerler **IDSymbol** ile tanımlanır
 - `usedList="1"` için **IDSymbol** değeri **1** olan `usedList="bmpPic1"` aynıdır
 - ***CmdSet** isimli **GuidSymbol** değerleri, komut gruplarını işaretlemek için kullanılır
 - ***Images** olarak tanımlananlar ise resimler işaretlerler
- **IDSymbol** değeri **CommandId** ile bitenler komutların guid değerlerini temsil eder


```
<Symbols>
<!-- Paket bilgisi tutan değişken. -->
<GuidSymbol name="guidDesignDifferPackage" value="{97999930-ccf3-4150-8507-52957afe824c}" />

<!-- Menü komutlarını bir arada tutmak için kullanılan değişkenler. -->
<GuidSymbol name="guidFile_VSPackageCmdSet" value="{020df2a1-db50-4da9-b02d-429321000270}">
    <IDSymbol name="FileContextMenuGroup" value="0x110" />
    <IDSymbol name="EditorContextMenuGroup" value="0x120" />
    <IDSymbol name="DiffFilesCommandId" value="0x0100" />
    <IDSymbol name="CompareHistoryCommandId" value="0x0101" />
    <IDSymbol name="DiffContentCommandId" value="0x0102" />
</GuidSymbol>

<!-- Resimleri bağlamak için kullanılan değişkenler. -->
<GuidSymbol name="guidImages" value="{2a122da1-1d9a-48b8-89ff-ee6527567153}">
    <!-- Birden fazla resim içeren bir dosyadan istenen sıradaki resmi almak için kullanılır -->
    <IDSymbol name="firstImage" value="1" />
    <IDSymbol name="bmpDiff" value="2" />
</GuidSymbol>

<!-- İsteğe bağlı sabit değerler de kullanılabildiğinden değişkensiz resim işaretçisi tanımlanabilir -->
<GuidSymbol name="guidImages1" value="{ed02defe-f3bf-4cf7-913b-9772fffe8e26}">
</GuidSymbol>
<GuidSymbol name="guidImages2" value="{1dd321f2-63ba-4bff-ae8-6f6d3995c650}">
</GuidSymbol>
</Symbols>
```



Komut Ekleme

- Komutlar **Commands** içerisinde **Groups** objeleri içindeki **Group** değerleri ile derlenir
- **Group** içerisindeki **Parent** objelerindeki **id** değeri ile komutun nasıl çalışacağını ifade ederiz
 - **IDM_VS_CTXT_ITEMNODE** ile **Solution Explorer** üzerine sağ tıklandığında çıkan menüye eklenen komutları
 - **IDM_VS_CTXT_CODEWIN** ile editöre sağ tıklandığında çıkan menüdeki komutları
 - [GUIDs and IDs of Visual Studio menus](#)
 - [IDE-Defined Commands for Extending Project Systems](#)
- **priority** değeri ile grubun bulunacağı konum belirlenir, düşük değerler daha yukarıya alır

```
<Groups>
  <Group guid="guidFile_VSPackageCmdSet" id="FileContextMenuGroup"
priority="0x0400">
    <Parent guid="guidSHLMainMenu" id="IDM_VS_CTXT_ITEMNODE" />
  </Group>
  <Group guid="guidFile_VSPackageCmdSet" id="EditorContextMenuGroup"
priority="0x0300">
    <Parent guid="guidSHLMainMenu" id="IDM_VS_CTXT_CODEWIN" />
  </Group>
</Groups>
```

Buton Ekleme

- Butonlar **Buttons** alanına **GUID** değeri ile eklenir
- Buton guid değeri paket guid değeri ile aynı olur
- **priority** değeri ile butonun bulunacağı konum belirlenir, düşük değerler daha yukarıya alır
- Parent alanı ile bağlı olduğu komut butonu guid değeri verilir
 - Bu sayede grup üzerindeki komutların davranışlarına uygun konumlanır
 - **IDM_VS_CTXT_ITEMNODE** id değerine sahip komut grubu için **Solution Explorer** üzerine sağ tıklandığında çıkan menüye eklenir
- Icon alanı ile butonun sağında olan ikon guid ile belirlenir
 - **guid** ile ikon dosyasına **id** ile kaçınıcı resmi almak istediğimizi belirtiriz
 - Resim indeksleri 0'dan değil 1'den başlar
 - İkonları tutmak için **Bitmaps** alanında guid değerli **Bitmap** objeleri kullanılır
 - **href** alanı temsil ettiği resim objenin yolunu tutar
 - **usedList** ile birden fazla resim içeren objelerden, hangi resimlerin alınacağını söyleriz
 - **usedList="1"** ile ilk resmi, **usedList="1, 2, 3"** ile ilk 3 resmi projeye dahil ederiz
- **Strings** alanında **ButtonText** ise butondaki metni temsil eder

```

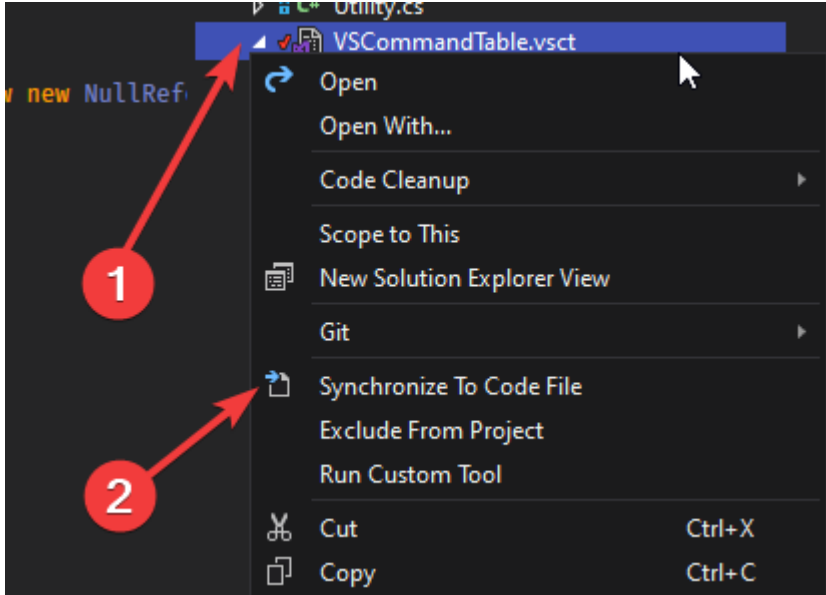
<Buttons>
  <Button guid="guidFile_VSPackageCmdSet" id="DiffFilesCommandId" priority="0x0100"
type="Button">
    <Parent guid="guidFile_VSPackageCmdSet" id="FileContextMenuGroup" />
    <Icon guid="guidImages" id="1" />
    <Strings>
      <ButtonText>Compare with file...</ButtonText>
    </Strings>
  </Button>
  <Button guid="guidFile_VSPackageCmdSet" id="CompareHistoryCommandId"
priority="0x0100" type="Button">
    <Parent guid="guidFile_VSPackageCmdSet" id="FileContextMenuGroup" />
    <Icon guid="guidImages1" id="1" />
    <Strings>
      <ButtonText>Compare designer file with history...</ButtonText>
    </Strings>
  </Button>
  <Button guid="guidFile_VSPackageCmdSet" id="DiffContentCommandId"
priority="0x0100" type="Button">
    <Parent guid="guidFile_VSPackageCmdSet" id="EditorContextMenuGroup" />
    <Icon guid="guidImages2" id="1" />
    <Strings>
      <ButtonText>Sort Windows Form Designer Generated Code</ButtonText>
    </Strings>
  </Button>
</Buttons>

<!-- guidImages değişkeninin GuidSymbol olarak tanımlanmış olmaları gerekmektedir -->
<Bitmaps>
  <Bitmap guid="guidImages" href="Resources\DiffFilesCommand.png" usedList="1" />
  <Bitmap guid="guidImages1" href="Resources\CompareHistoryCommand.png"
usedList="1" />
  <Bitmap guid="guidImages2" href="Resources\DiffContentCommand.png" usedList="1"
/>
</Bitmaps>

```

Kaynak Koda Bağlama

- Komutların tetiklenmesi durumunda yapılacak eylemler kaynak kod tarafında **Execute** metodu içerisinde belirlenir
- Kaynak koda komutu bağlamak için **CommandSet** ve o küme içerisindeki **CommandId** değeri gerekir
 - **CommandSet**, **vsct** dosyası içerisinde ***CmdSet** olarak adlandırılan guid değerini alır
 - **CommandId** yukarıdaki guid içerisindeki hedeflenen komutun **IDSymbol** değerlerini alır
- Bu işlemi senkronize etmek ve her guid değişikliğinde güncellemekten kurtulmak için
 - **vsct** dosyasına sağ tıklayın ve **Synchronize code file** butonuna tıklayın
 - **PackageGuids** ve **PackageIds** içeren C# **class** objeleri otomatik olarak tanımlacaktır



```
namespace DesignerDiffer
{
    /// <summary>
    /// Command handler
    /// </summary>
    internal sealed class DiffContentCommand
    {
        /// <summary>
        /// Command ID.
        /// </summary>
        public const int CommandId = PackageIds.DiffContentCommandId;

        /// <summary>
        /// Command menu group (command set GUID).
        /// </summary>
        public static readonly Guid CommandSet =
        PackageGuids.guidFile_VSPackageCmdSet;




        /// <summary>
        /// VS Package that provides this command, not null.
        /// </summary>
        private readonly AsyncPackage package;

        // ...
    }
}
```

```
}  
}
```

IDE Objelerine Erişme

Visual Studio Ortam Objelerine Erişme

-  IDE objelerine erişmek için `EnvDT80.DTE2` objesi kullanılır
-  `await <asyncServiceProvider>.GetServiceAsync(typeof(DTE)).ConfigureAwait(false) as DTE2` kodu ile DTE objesi alınır
-  `dte2.ItemOperations` kodu ile dosya açma, ekleme ve benzeri işlemler IDE ile otomatikleştirilebilir

Kod	Açıklama
<code>dte2.ActiveDocument</code>	IDE üzerinde aktif olan doküman
<code>dte2.ActiveDocument.ProjectItem</code>	Aktif dokümanın proje objesi (kaynak kodlara vb erişim)
<code>dte2.ToolWindows.SolutionExplorer.SelectedItems</code>	Solution Explorer üzerindeki seçilen dosyalara erişiriz
<code>dte2.ExecuteCommand("<komut>", "<argümanlar>")</code>	Command Window üzerinde komut çalıştırır
<code>dte2.ItemOperations.AddExistingItem(<filepath>)</code>	Projeye var olan dosyayı ekler ve yolun -proje dizininde olduğundan- günceller
<code>dte2.ItemOperations.OpenFile(<filepath>)</code>	IDE ile dosyayı açar, projeye dahil etmem, kaynak kod derlenmez (FileCodeModel olmaz)

ProjectItem

- 💡 Solution içerisinde yer alan ve derlenen proje dosyasını tutan objedir
- 🛠️ Dosya üzerindeki otomasyon işlemleri bu obje ile yapılır
- 📁 Dosya işlemleri `<projectItem>.Delete()`, `<projectItem>.Save()`, `<projectItem>.Remove()` gibi işlemler buradan yapılır
- 📄 Dosya içerisindeki kaynak kod modeline `<projectItem>.FileCodeModel` şeklinde erişebiliriz

```
ProjectItem selectedProjectItem = dte2.ItemOperations.AddExistingItem(filePath);
FileCodeModel selectedFileCodeModel = selectedProjectItem.FileCodeModel;
```

FileCodeModel

- 💡 IDE üzerinde derlenen (build) proje dosyaları (ProjectItem) kaynak kodlarını tutan modeldir
- 🍷 `CodeElements` olan kod elemanlarını tutan objelerden oluşur
- 📁 `CodeNamespace`, `CodeElement`, `CodeClass`, `CodeFunction` gibi kaynak kod özelliğine göre obje içerir
- 📄 `<codeNamespace | codeClass >.Children` komutu ile namespace veya class içerisindeki kaynak kod objelerine erişilir

🔗 Derlenmemiş dosyalarda - yani projeye dahil olmayan harici dosyalar olan `Miscellaneous` dosyalarında - `FileCodeModel` olmaz





```

public static bool IsFuncExistInCodeElements(CodeElements codeElements, string name,
out CodeFunction cf)
{
    ThreadHelper.ThrowIfNotOnUIThread();
    foreach (CodeElement element in codeElements)
    {
        if (element is CodeNamespace)
        {
            CodeNamespace nsp = element as CodeNamespace;

            foreach (CodeElement subElement in nsp.Children)
            {
                if (subElement is CodeClass)
                {
                    CodeClass c2 = subElement as CodeClass;
                    foreach (CodeElement item in c2.Children)
                    {
                        if (item is CodeFunction)
                        {
                            CodeFunction _cf = item as CodeFunction;
                            if (_cf.Name == name)
                            {
                                cf = _cf;
                                return true;
                            }
                        }
                    }
                }
            }
        }
    }
    cf = null;
    return false;
}

```


CodeElement

-  CodeElement objelerinin metinlerine
`<codeElement>.GetStartPoint(vsCmpPart.vsCmpPartBody).CreateEditPoint()` şeklinde erişilir
-  `GetStartPoint(<vsCmpPart>)` ile enum değerleri olarak tanımlanan alanların başlangıç konumu alınır
-  `CreateEditPoint` ile konum bilgisinden içerik metnine erişilir
- İçerik metni üzerinden `GetText(<point>)`, `ReplaceText(<point>)` gibi komutlar metni değiştirebiliriz
-  Obje sonuna kadar almak veya değiştirmek için `<codeElement>.EndPoint` değeri kullanılır

```
public static bool IsFuncExistInCodeElements(CodeElements codeElements, string name,
out CodeFunction cf)
{
    string functionBodyText =
cf.GetStartPoint(vsCmpPart.vsCmpPartBody).CreateEditPoint().GetText(cf.EndPoint);
    funcitonBodyText = "Test";

cf.GetStartPoint(vsCmpPart.vsCmpPartBody).CreateEditPoint().ReplaceText(cf.EndPoint,
funcitonBodyText, (int)vsEPReplaceTextOptions.vsEPReplaceTextAutoformat);
}
```

Kod Örnekleri

Aktif Dökümanın İçeriğini Alma

```
protected DTE2 dte;
dte2 = (EnvDTE80.DTE2)GetService(typeof(EnvDTE.DTE));

public string GetCurrentTextFile(){

    TextDocument doc = (TextDocument)(dte.ActiveDocument.Object("TextDocument"));
    var p = doc.StartPoint.CreateEditPoint();
    string s = p.GetText(doc.EndPoint);

    return s;
}
```

 In VisualStudio DTE, how to get the contents of the ActiveDocument?

Editör Üzerindeki Seçili Metni Sıralama

```
using EnvDTE80;

var dte = await ServiceProvider.GetServiceAsync(typeof(DTE)).ConfigureAwait(false) as
DTE2 ?? throw new NullReferenceException("DTE alınamadı");
EnvDTE.TextSelection ts = dte.ActiveWindow.Selection as EnvDTE.TextSelection;
if (ts == null)
    return;

string[] selectedLines = ts.Text.Split('\n');
selectedLines = selectedLines.OrderBy(p => p).ToArray();
ts.Text = string.Join("\n", selectedLines);
```

- [How to get selected text of visual studio 2015 editor windows?](#)
- [Read a text file and sort in C#](#)



Editör Üzerindeki Seçili Metnin İçerisindeki Method İçeriğini Sıralama

```
using EnvDTE80;

var dte = await ServiceProvider.GetServiceAsync(typeof(DTE)).ConfigureAwait(false) as
DTE2 ?? throw new NullReferenceException("DTE alınamadı");

EnvDTE.TextSelection ts = dte.ActiveWindow.Selection as EnvDTE.TextSelection;
if (ts == null)
    return;
EnvDTE.CodeFunction func =
ts.ActivePoint.CodeElement[vsCMElement.vsCMElementFunction] as EnvDTE.CodeFunction;
if (func == null)
    return;

// Func içeriğini al -> sırala -> güncelle
string selectedCodeText =
func.GetStartPoint(vsCMPart.vsCMPartBody).CreateEditPoint().GetText(func.EndPoint);
selectedCodeText = string.Join("\n", selectedCodeText.Split('\n').OrderBy(p => p));
func.GetStartPoint(vsCMPart.vsCMPartBody).CreateEditPoint().ReplaceText(func.EndPoint
, selectedCodeText, (int) vsEPReplaceTextOptions.vsEPReplaceTextAutoformat);
```

- [vs2010 automation](#) : Get the text value of a EnvDTE.CodeElement
- [Get function body programatically using Automation](#)

Aktif Dokümandaki Üretilen Kodları Sıralama

```
using EnvDTE80;

private async void Execute(object sender, EventArgs e) {
    DTE2 dte = await Utility.GetDTE2Async(ServiceProvider);
    ProjectItem tempProjectItem = dte2.ItemOperations.AddExistingItem(tempFilePath);
    if (Utility.SortFunctionBodyIfExist(tempProjectItem.FileCodeModel,
Utility.GeneratedFunctionName))
    {
        tempProjectItem.Save();
        string oldFilePath = filePath.Replace(selectedProjectItem.Name,
tempProjectItem.Name);
        Utility.DiffFiles(dte2, oldFilePath, filePath);
    }
}

public static async Task<DTE2> GetDTE2Async(IAsyncServiceProvider
asyncServiceProvider) => await
asyncServiceProvider.GetServiceAsync(typeof(DTE)).ConfigureAwait(false) as DTE2 ??
throw new NullReferenceException("DTE alınamadı");

public static bool SortFunctionBodyIfExist(FileCodeModel fcm, string funcName)
{
    ThreadHelper.ThrowIfNotOnUIThread();
    if (IsFuncExistInFileCodeModel(fcm, funcName, out CodeFunction cf))
    {
        SortFunctionBody(cf);
        return true;
    }
    return false;
}

public static bool IsFuncExistInFileCodeModel(FileCodeModel fcm, string name, out
CodeFunction cf)
{
    ThreadHelper.ThrowIfNotOnUIThread();
    return IsFuncExistInCodeElements(fcm.CodeElements, name, out cf);
}

public static void SortFunctionBody(CodeFunction cf)
{
    ThreadHelper.ThrowIfNotOnUIThread();
    string generatedCode = GetFunctionBodyText(cf);
    generatedCode = StripComments(generatedCode);
    generatedCode = SortContentBy(generatedCode, '\n');
    ReplaceFunctionBodyText(generatedCode, cf);
}

public static bool IsFuncExistInCodeElements(CodeElements codeElements, string name,
out CodeFunction cf)
```

```

{
    ThreadHelper.ThrowIfNotOnUIThread();
    foreach (CodeElement element in codeElements)
    {
        if (element is CodeNamespace)
        {
            CodeNamespace nsp = element as CodeNamespace;

            foreach (CodeElement subElement in nsp.Children)
            {
                if (subElement is CodeClass)
                {
                    CodeClass c2 = subElement as CodeClass;
                    foreach (CodeElement item in c2.Children)
                    {
                        if (item is CodeFunction)
                        {
                            CodeFunction _cf = item as CodeFunction;
                            if (_cf.Name == name)
                            {
                                cf = _cf;
                                return true;
                            }
                        }
                    }
                }
            }
        }
    }
    cf = null;
    return false;
}

```

[Get current source file methods in Visual Studio Text Editor Extension](#)

İki Dosya Arasındaki Farklılıkları Gösterme

- Dosyayı geçici dizine aynı uzantı ve ismle kayıt ediyoruz
- Ardından `Tool.DiffFiles` komutu ile geçici dizindeki ile orjinal dosyayı karşılaştırıyoruz

```

string[] splitFilepath = filepath.Split('\\');
string bareFilename = splitFilepath[splitFilepath.Length - 1];
string tempFilepath = System.IO.Path.GetTempPath() + bareFilename;
System.IO.File.WriteAllText(tempFilepath, fileContent, System.Text.Encoding.UTF8);
dte2.ExecuteCommand("Tools.DiffFiles", $"\"{tempFilepath}\" \"{filepath}\"");

```

✂ Git Komutu Çalıştırma

📁 Proje dizinini ve dosya yolunu alma

```
string filepath = "...";
string solutionDir = System.IO.Path.GetDirectoryName(dte2.Solution.FullName);
filepath = filepath.Replace($"{solutionDir}\\", "").Replace("\\", "/");
```

🧑🏻 How do you get the current solution directory from a VSPackage?

Git process oluşturma ve okuma






- `arguments` alanına

```
static System.Diagnostics.Process GitProcess(string arguments, string workdir)
{
    return new System.Diagnostics.Process
    {
        StartInfo = {
            FileName = "git.exe",
            WorkingDirectory = workdir,
            Arguments = $"--no-pager {arguments}",
            RedirectStandardOutput = true,
            UseShellExecute = false,
            CreateNoWindow = true
        },
        EnableRaisingEvents = true
    };
}
```

```
string fileContent = "";
gitProcess.Start();
while (!gitProcess.StandardOutput.EndOfStream)
{
    string line = gitProcess.StandardOutput.ReadLine();
    fileContent += line + "\n";
}
```

- `Process.start`: how to get the output?
- Is there any async equivalent of `Process.Start`?

Dosyadan FileCodeModel Oluřturma

-  Öncelikle stack overflow üzerindeki cevabım [buradadır](#)
-  Visual Studio içerisinde açılan proje dosyalarının `FileCodeModel` objesine erişilebilir
-  `dte.ItemOperations.OpenFile` ile açılan dosyalar, `Miscellaneous` olarak gözükür, `ProjectItem` değildir
-  Dosyayı `ProjectItem` olarak açmak için `dte.ItemOperations.AddExistingItem(filePath);` kodu kullanılır
-  Eklenen dosyanın `FileCodeModel` içeriğine `projectItem.FileCodeModel` şeklinde erişiriz
- İsteğe baęlı olarak `ProjectItem.Delete()` ile eklenen dosya kaldırılabilir

```
using EnvDTE;

public static async Task<DTE2> GetDTE2Async(IAsyncServiceProvider
asyncServiceProvider) => await
asyncServiceProvider.GetServiceAsync(typeof(DTE)).ConfigureAwait(false) as DTE2 ??
throw new NullReferenceException("DTE alınamadı");

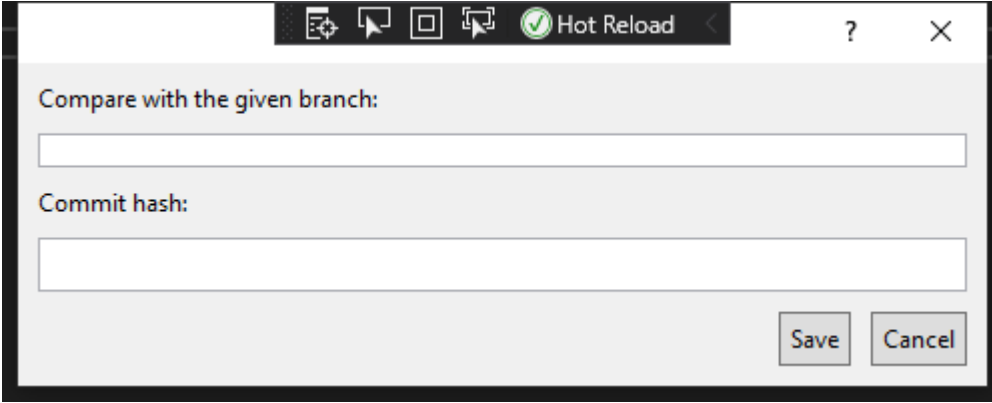
string filepath = "TODO";
DTE2 dte2 = await Utility.GetDTE2Async(asyncServiceProvider);
ProjectItem projectItem = dte2.ItemOperations.AddExistingItem(filepath);
FileCodeModel fcm = projectItem.FileCodeModel;
projectItem.Delete();
```

- [How do I programmatically add a file to a Solution?](#)
- [FileCodeModel null for file in "Misc Files" project. ~ Windows Tech](#)
- [FileCodeModel null for file in "Misc Files" project. ~ Microsoft](#)

Designer Dosyasını Önceki Sürümü ile Sıralanarak Karşılaştırılması

- `Form.Designer.cs` dosyasına sağ tıklayın `Compare with History...` butonunu seçin
- Çıkan form üzerinden istenen branch ve commit hash ile eski sürümü ile kıyaslamayı yapılandırın
- Dosyanın bir önceki sürümü `%TEMP%` dizininde `~<filename>` adıyla oluşturulur
 - `%TEMP%` dizini işletim sistemi otomatik olarak temizlenen bir dizindir
- Oluşturulan geçici dosya aktif projeye dahil edilir ve windows form tarafından üretilen kodlar sıralanır, kaydedilir
- Seçilen dosya içeriğindeki windows form tarafından üretilen kodlar sıralanır, kaydedilmez
- Dosyalar Visual Studio 2019 Tools aracı olan DiffFiles ile karşılaştırılır

VSIX için Dialog Box Oluşturma



Bağımlılıkları Tanımlama

- **Add references** alanından gerekli bağımlılıkları ekliyoruz
 - PresentationCore
 - PresentationFramework
 - WindowsBase
 - System.Xaml

- [Create a dialog box derived from DialogWindow](#)

Dialog penceresini oluşturma

- **Add new folder** ile **Dialogs** adı altında dizin açın
- Dizine sağ tıklayıp **Add - New Item - User Control (WPF)** ile **xaml** ve **xaml.cs** dosyaları oluşturuyoruz
- ***.xaml.cs** dosyasındaki **UserControl** -> **DialogWindow** olarak güncellenecek
- ***.xaml** dosyasına aşağıdaki değişiklikler yapılmıştır
 - `xmlns:platformUi="clr-namespace:Microsoft.VisualStudio.PlatformUI;assembly=Microsoft.VisualStudio.Shell.15.0"` (15 değeri güncel sürümlerde değişebilir)
 - `platformUi:DialogWindow`
 - `WindowStartupLocation="CenterScreen" Width="500" Height="200"`


```

<platformUi:DialogWindow x:Class="<NAMESPACE>.<CLASS_NAME>"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expressions/blend/2008"
    xmlns:platformUi="clr-
namespace:Microsoft.VisualStudio.PlatformUI;assembly=Microsoft.VisualStudio.Shell.15.
0"

    mc:Ignorable="d"
    WindowStartupLocation="CenterScreen"
    Width="500" Height="200">

    <!--Title="Örnek amaçlı GUI"-->
    <Border Margin="5" >
        <Grid>
            <Grid.RowDefinitions>
                <RowDefinition Height="
"/>
                <RowDefinition Height="
"/>
                <RowDefinition Height="
"/>
                <RowDefinition Height="*/>
                <RowDefinition Height="
"/>
            </Grid.RowDefinitions>
            <TextBlock Text="Compare with the given branch: " Margin="5"/>
            <TextBox Grid.Row="1" x:Name="SelectionTextBox" Margin="5"
MaxHeight="100"
                ScrollViewer.VerticalScrollBarVisibility="
"
                />
            <TextBlock Grid.Row="2" Margin="5">Commit hash:</TextBlock>
            <TextBox Grid.Row="3" x:Name="DocumentationTextBox"
                AcceptsReturn="True" TextWrapping="Wrap"
HorizontalAlignment="Stretch" Margin="5"
                ScrollViewer.VerticalScrollBarVisibility="
"/>
            <StackPanel Orientation="Horizontal" HorizontalAlignment="Right"
Grid.Row="4">
                <Button Margin="5" Padding="5" Click="OnCompare">Save</Button>
                <Button Margin="5" Padding="5" Click="OnCancel">Cancel</Button>
            </StackPanel>
        </Grid>
    </Border>
</platformUi:DialogWindow>

```

- Code tarafında aşağıdaki method ile diyalog penceresini gösteriyoruz

```
private void ShowCompareWithHistoryWindow()
{
    var compareFileWithHistoryDialog = new
CompareWithHistoryDialog("Microsoft.VisualStudio.PlatformUI.DialogWindow")
    {
        HasMinimizeButton = false,
        HasMaximizeButton = false
    };
    compareFileWithHistoryDialog.ShowDialog();
}
```

- [Options for displaying modal dialogs in Visual Studio extensions](#)
- [Options for displaying modal dialogs in Visual Studio extensions - GitHub Changes](#)
- [Part 4: Show a popup Window](#)
- [Part 4: Show a popup Window - Github Source Code](#)

OnClick ve OnCompare eylemleri tanımlama

- `xaml` dosyası içerisinde aşağıdaki gibi tanımlanan butonlar için kaynak kod örneği verilmiştir

```
<Button Margin="5" Padding="5" Click="OnCompare">Save</Button>
<Button Margin="5" Padding="5" Click="OnCancel">Cancel</Button>
```

```
public partial class <CLASS_NAME> : DialogWindow
{
    public <CLASS_NAME>(string helpTopic) : base(helpTopic)
    {
        InitializeComponent();
    }

    private void OnCancel(object sender, RoutedEventArgs e)
    {
        this.Close();
    }

    private void OnCompare(object sender, RoutedEventArgs e)
    {
        MessageBox.Show("Save"); // TODO
    }
}
```

- [Part 4: Show a popup Window](#)

OnCompare Butonu Algoritmasını Oluşturma

```
private async void OnCompare(object sender, RoutedEventArgs e)
{
    string branch = this.BranchTextBox.Text.Trim();
    string commitHash = this.CommitHashTextBox.Text.Trim();

    if (branch == "")
    {
        MessageBox.Show("Gerekli alanlar doldurulmadığı için HEAD ile kıyaslanacak");
        branch = "HEAD";
    }

    var dte2 = await GetDTE2Async(asyncServiceProvider);

    await ThreadHelper.JoinableTaskFactory.SwitchToMainThreadAsync();
    if (CanFileBeCompared(dte2, out string filePath))
    {
        string solutionDir = System.IO.Path.GetDirectoryName(dte2.Solution.FullName);
        string fileContent = GetFileHistoryContent(solutionDir, filePath, branch,
commitHash);
        string tempFilePath = CopyContentToTemp(filePath, fileContent);

        ProjectItem selectedProjectItem =
dte2.ItemOperations.AddExistingItem(filePath);
        FileCodeModel selectedFileCodeModel = selectedProjectItem.FileCodeModel;
        if (selectedFileCodeModel != null)
        {
            if (SortFunctionBodyIfExist(selectedFileCodeModel,
Utility.GeneratedFunctionName))
            {
                ProjectItem tempProjectItem =
dte2.ItemOperations.AddExistingItem(tempFilePath);
                if (SortFunctionBodyIfExist(tempProjectItem.FileCodeModel,
Utility.GeneratedFunctionName))
                {
                    tempProjectItem.Save();
                    string oldFilePath = filePath.Replace(selectedProjectItem.Name,
tempProjectItem.Name);
                    DiffFiles(dte2, oldFilePath, filePath);
                }
                else
                {
                    MessageBox.Show("Seçili dosyanın belirtilen commit hash için
kaydı git ile bulunamadı");
                }
                tempProjectItem.Delete();
            }
            else
            {

```

```

        MessageBox.Show("Seçili dosya designer dosyası değil");
    }
}
else
{
    MessageBox.Show("Dosya içeriği desteklenmiyor");
}
}
this.Close();
}

public static bool CanFileBeCompared(DTE2 dte, out string filepath)
{
    ThreadHelper.ThrowIfNotOnUIThread();

    filepath = GetSelectedFiles(dte).ElementAtOrDefault(0);
    return !string.IsNullOrEmpty(filepath);
}

/// <summary>
/// Dosya yollarının bilgilerini verir
/// </summary>
/// <param name="dte">VS için otomasyon objesi</param>
/// <returns></returns>
public static IEnumerable<string> GetSelectedFiles(DTE2 dte)
{
    ThreadHelper.ThrowIfNotOnUIThread();
    var items = (Array)dte.ToolWindows.SolutionExplorer.SelectedItems;
    return from item in items.Cast<UIHierarchyItem>()
           let pi = item.Object as ProjectItem
           select pi.FileNames[1];
}

public static string GetFileHistoryContent(string repo, string filepath, string
branch, string commitHash)
{
    string relatedFilePath = filepath.Replace($"{repo}\\", "").Replace("\\", "/");

    string fileContent = "";
    var gitProcess = GitProcess($"show {branch}{commitHash}:{relatedFilePath}",
repo);
    gitProcess.Start();
    while (!gitProcess.StandardOutput.EndOfStream)
    {
        string line = gitProcess.StandardOutput.ReadLine();
        fileContent += line + "\n";
    }

    return fileContent;
}

public static System.Diagnostics.Process GitProcess(string arguments, string workdir)
=> new System.Diagnostics.Process
{

```

```

        StartInfo = {
            FileName = "git.exe",
            WorkingDirectory = workdir,
            Arguments = $"--no-pager {arguments}",
            RedirectStandardOutput = true,
            UseShellExecute = false,
            CreateNoWindow = true
        },
        EnableRaisingEvents = true
    };

    public static string CopyContentToTemp(string filepath, string fileContent)
    {
        string tempFilepath = System.IO.Path.GetTempPath() + "~" +
        filepath.Split('\\').Last();
        System.IO.File.WriteAllText(tempFilepath, fileContent, Encoding.UTF8);
        return tempFilepath;
    }

    public static bool SortFunctionBodyIfExist(FileCodeModel fcm, string funcName)
    {
        ThreadHelper.ThrowIfNotOnUIThread();
        if (IsFuncExistInFileCodeModel(fcm, funcName, out CodeFunction cf))
        {
            SortFunctionBody(cf);
            return true;
        }
        return false;
    }

    public static bool IsFuncExistInFileCodeModel(FileCodeModel fcm, string name, out
    CodeFunction cf)
    {
        ThreadHelper.ThrowIfNotOnUIThread();
        return IsFuncExistInCodeElements(fcm.CodeElements, name, out cf);
    }

    public static bool IsFuncExistInCodeElements(CodeElements codeElements, string name,
    out CodeFunction cf)
    {
        ThreadHelper.ThrowIfNotOnUIThread();
        foreach (CodeElement element in codeElements)
        {
            if (element is CodeNamespace)
            {
                CodeNamespace nsp = element as CodeNamespace;

                foreach (CodeElement subElement in nsp.Children)
                {
                    if (subElement is CodeClass)
                    {
                        CodeClass c2 = subElement as CodeClass;
                        foreach (CodeElement item in c2.Children)
                        {

```

```

        if (item is CodeFunction)
        {
            CodeFunction _cf = item as CodeFunction;
            if (_cf.Name == name)
            {
                cf = _cf;
                return true;
            }
        }
    }
}

cf = null;
return false;
}

public static void SortFunctionBody(CodeFunction cf)
{
    ThreadHelper.ThrowIfNotOnUIThread();
    string generatedCode = GetFunctionBodyText(cf);
    generatedCode = StripComments(generatedCode);
    generatedCode = SortContentBy(generatedCode, '\n');
    ReplaceFunctionBodyText(generatedCode, cf);
}

public static void DiffFiles(DTE2 dte2, string filepath1, string filepath2)
{
    dte2.ExecuteCommand("Tools.DiffFiles", $"\"{filepath1}\" \"{filepath2}\"");
}

```