

# Özelleştirilmiş Fonksiyonlar

**Özelleştirilmiş Fonksiyon** (*Custom Function*) yazabilmek için izlenmesi gereken adımlar, hazırlanmış bir fonksiyonu içeri alma işlemleri ve konu ile ilgili hazır paket örnekler aşağıdaki gibidir:

- [Kurulum](#)
- [Fonksiyon Oluşturulması](#)
- [Versiyon](#)
- [İçeri Alma İşlemi ve Tasarım Aracı Üzerinde Özelleştirilmiş Fonksiyon Kullanımı](#)
- [Örnek İndir](#)

## Kurulum

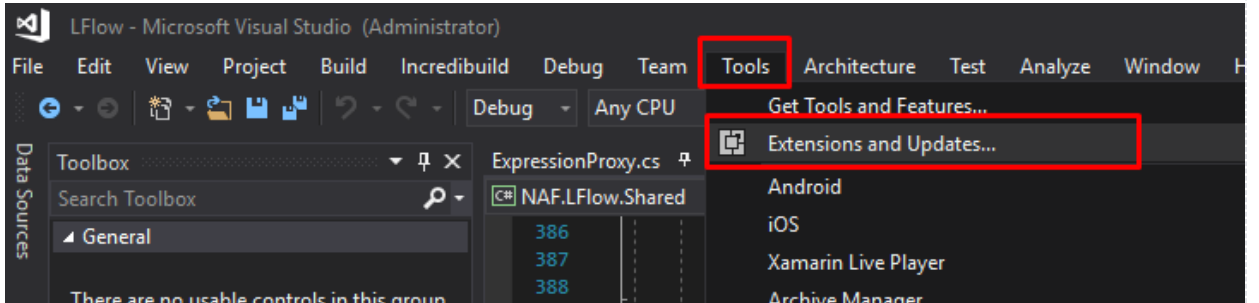
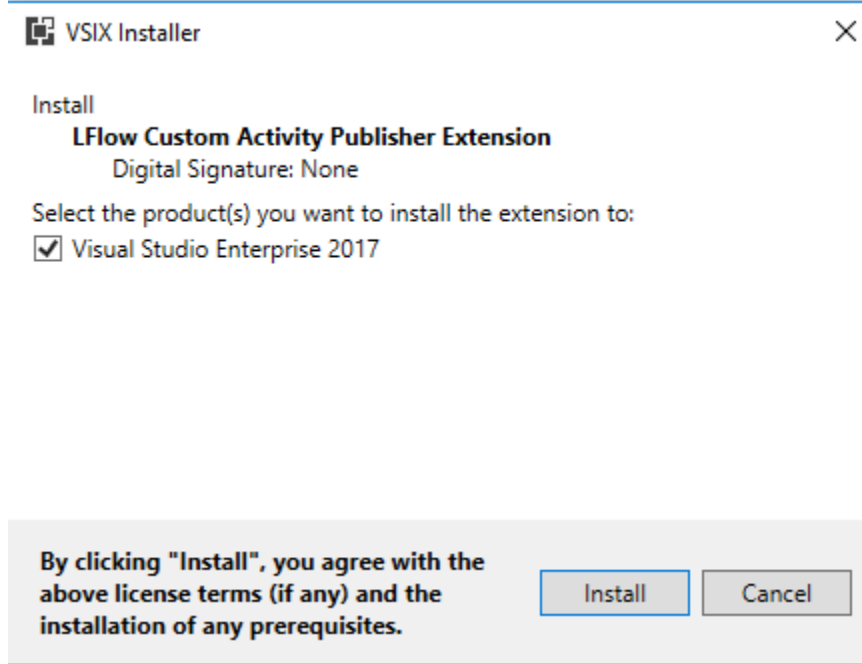
1. Custom Function için gerekli template ve yardımcı uygulamalar için visual studio'ya ilgili Framework versiyonuna göre paylaşılan eklentinin çalıştırılması gerekmektedir.

Framework 4.6 versiyonu için [eklenti dosyası](#)

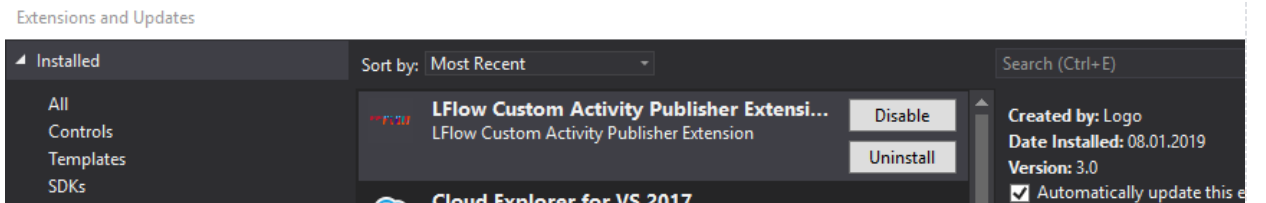
Framework 4.8 versiyonu için [eklenti dosyası](#)

 LFlow.CustomActivity.Publisher.Extension.vsix	08.01.2019 09:54	Microsoft Visual S...	17.667 KB
---	------------------	-----------------------	-----------

2. Kurulum tamamlandıktan sonra Visual Studio uygulamasını açıp, aşağıdaki menü üzerinden "Extensions and Updates" ekranını açılır.



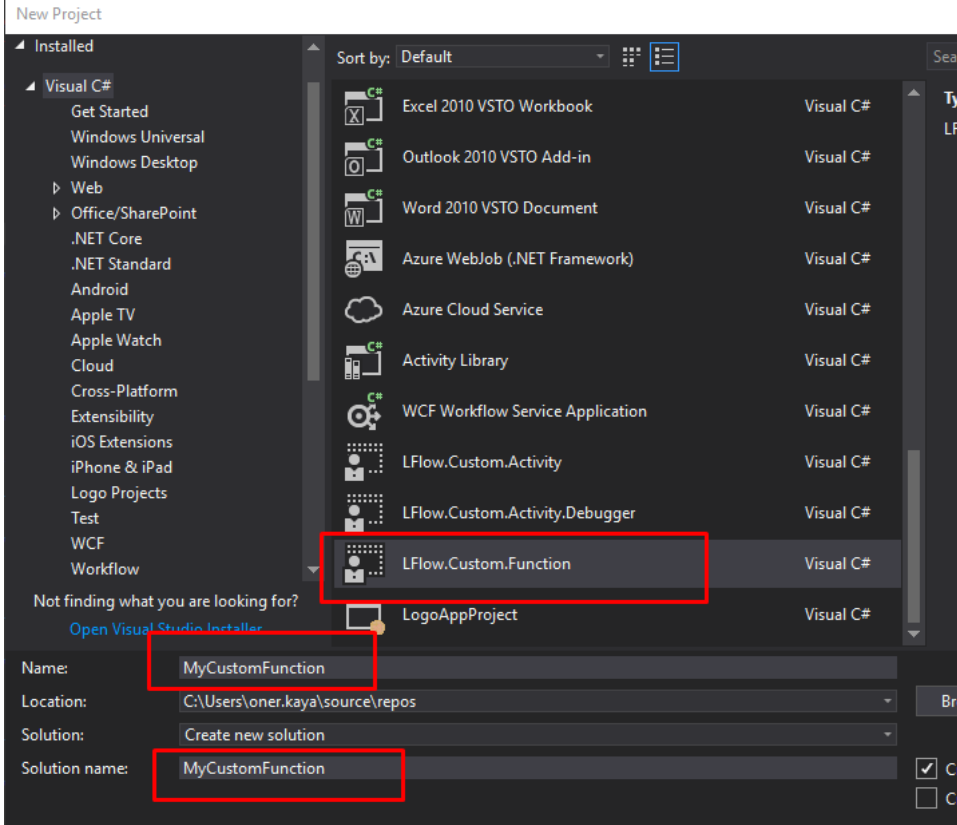
Aşağıdaki gibi eklentinin yüklenme durumu gözlemlenir.



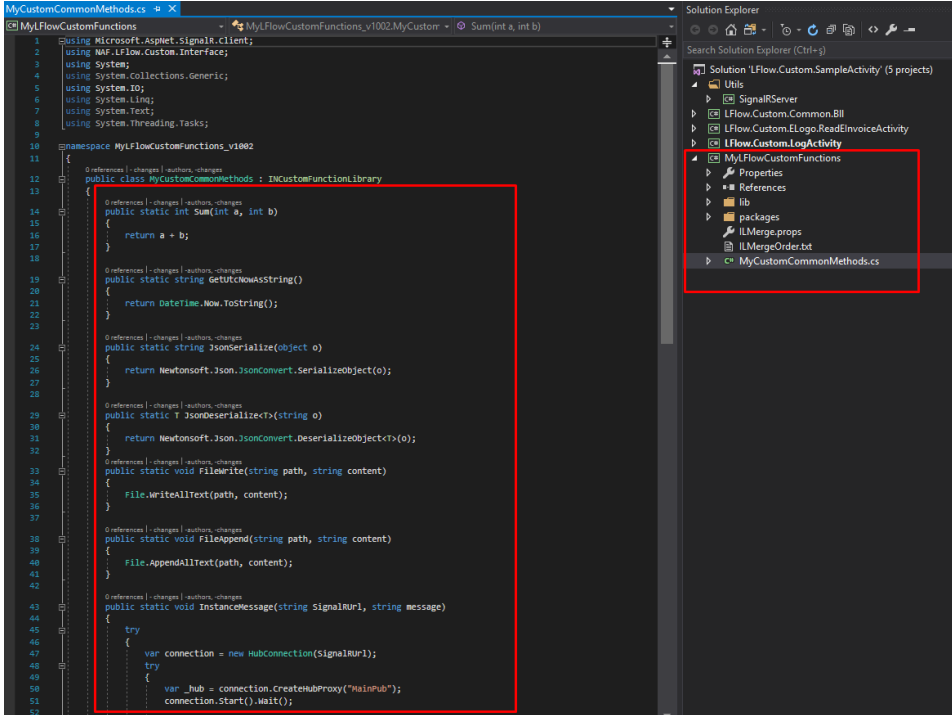
## Fonksiyon Oluşturulması

1. Visual Studio'da New Project üzerinden "LFlow.Custom.Function" template seçilir.

**Not:** Name ve Solution Name alanlarında '.' içermeyecek şekilde namespace belirlenmelidir. Örnek: "MyCustomFunction" vs.

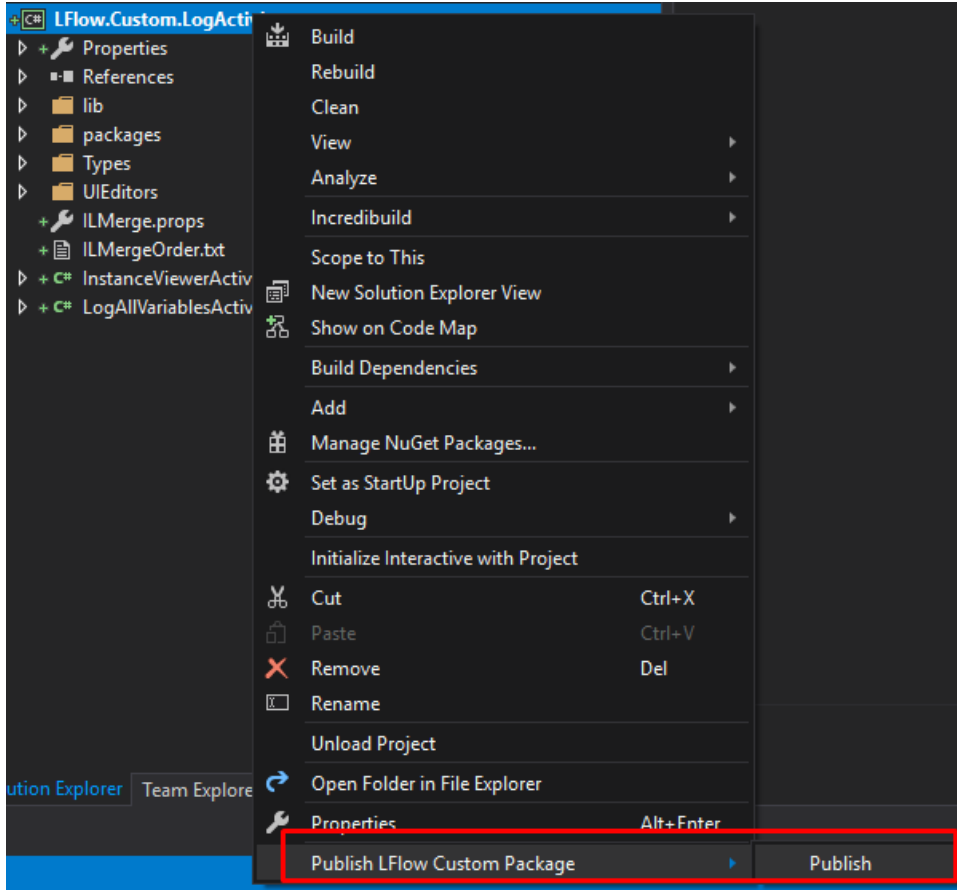


2. Oluşturulan proje detayı aşağıdaki gibi olacak.



3. Template açıldığında **FunctionLibrary.cs** dosyası örnek olması için eklenmiştir. Oluşturulacak fonksiyon sınıfı **ICustomFunctionLibrary** interface'inden implemente edilmelidir.

4. .Net Framework 4.6.1 ile derlenmiş dll leri referans edilebilmektedir. Publish işlemi sonrası oluşturulan paket içerisinde ilgili referans dll içerecektir.
5. Testleri ve geliştirimi tamamlanan özelleştirilmiş fonksiyon projesi sağ tıklanarak Publish edilebilir.



6. . Iflowfunccustom uzantılı bir dosya belirtilen dosya yoluna oluşturulur.

LFlow Custom Package Publisher

**LFlow custom package publish settings:**

Project File: C:\Deployment\Flow\CustomActivity\ActivitySample\MyLFlowCustomFunctions\My

Output Dir. Path: C:\Temp

Package Name: MyFlowCustomFunctions

Description: MyFlowCustomFunctions

Version: 1\_ - 0\_ - 0\_ - 1\_

Output:

Publish Close

MyLFlowCustomFunctions\_v1.0.0.2.iflowfunccustom 10.01.2019 15:45 LFLOWFUNCCUS... 668 KB

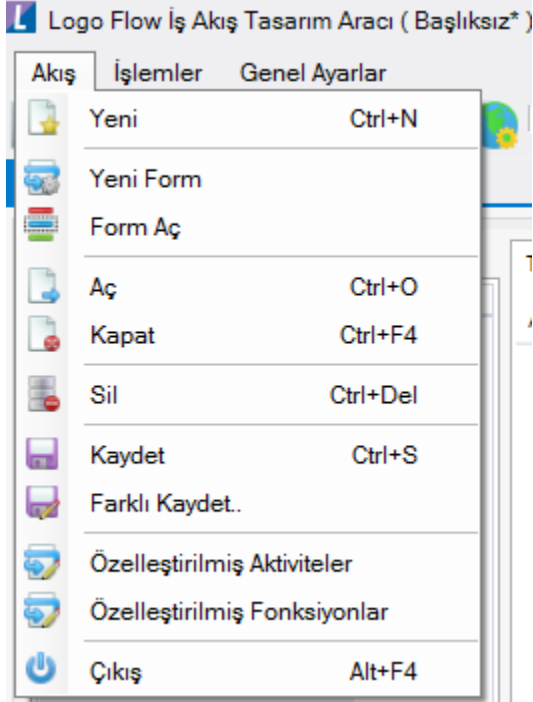
## Versiyon

Fonksiyon paketi oluşturma sırasında projenin namespace'i versiyon ile birlikte güncellenir. Bu şekilde fonksiyonun farklı versiyonları oluşturulmasına imkan sunularak iş akışlarında uygun versiyondaki fonksiyonların işletilmesi sağlanabilir.

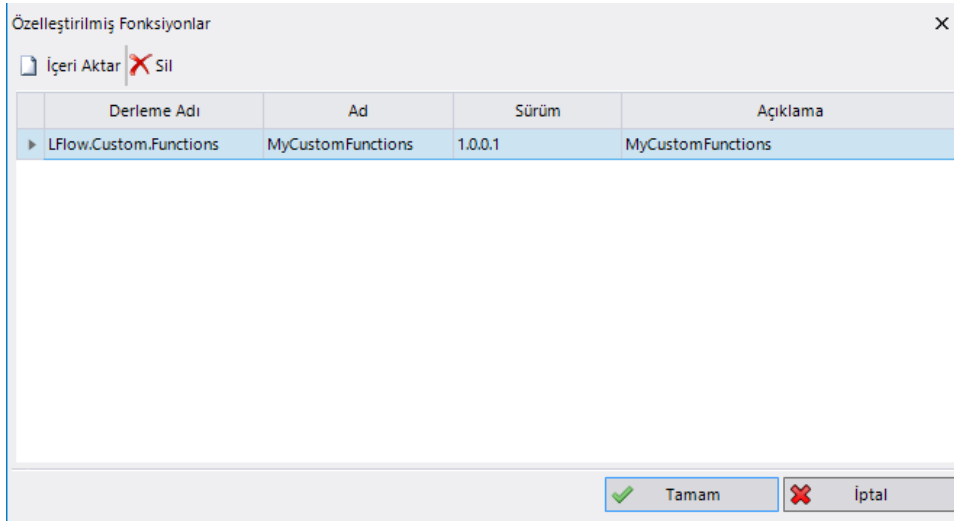
**namespace** MyCustomFunctions\_v1001

## İçeri Alma İşlemi ve Tasarım Aracı Üzerinde Özelleştirilmiş Fonksiyon Kullanımı

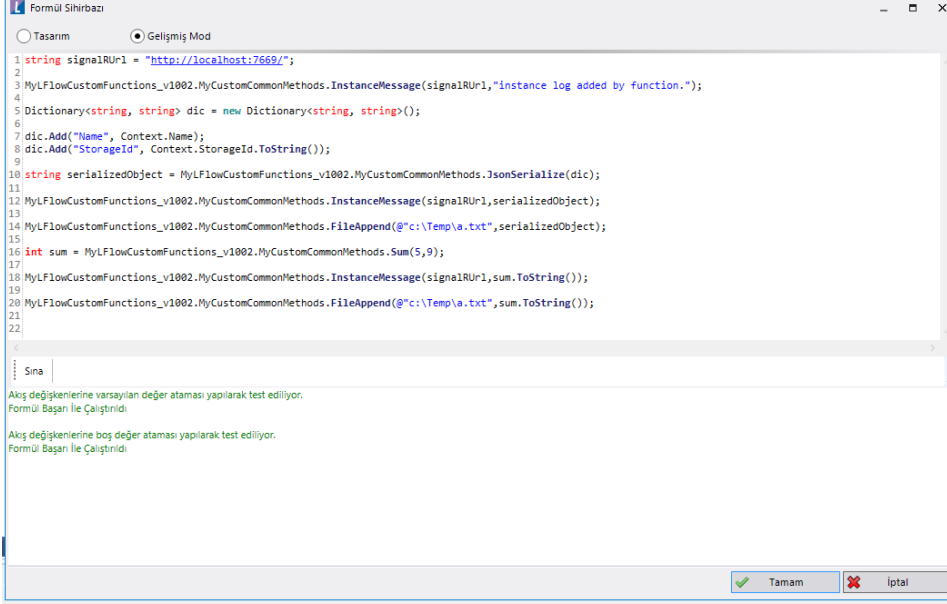
1. LogoFlow tasarım aracı "Akış" altında "Özelleştirilmiş Fonksiyonlar" açılır.



2. Önceden eklenmiş paketler burada görüntülenir. Yeni fonksiyon paketleri de "İçeri Aktar" ile aktarılır.



3. Özelleştirilmiş fonksiyon paketi eklendikten sonra formül editöründe LFlow.Custom üzerinden fonksiyon listesine ulaşılabilir ve işletilebilir.



```
1 string signalRUrl = "http://localhost:7669/";
2
3 MyLFlowCustomFunctions_v1002.MyCustomCommonMethods.InstanceMessage(signalRUrl,"instance log added by function.");
4
5 Dictionary<string, string> dic = new Dictionary<string, string>();
6
7 dic.Add("Name", Context.Name);
8 dic.Add("StorageId", Context.StorageId.ToString());
9
10 string serializedObject = MyLFlowCustomFunctions_v1002.MyCustomCommonMethods.JsonSerialize(dic);
11
12 MyLFlowCustomFunctions_v1002.MyCustomCommonMethods.InstanceMessage(signalRUrl,serializedObject);
13
14 MyLFlowCustomFunctions_v1002.MyCustomCommonMethods.FileAppend(@"c:\Temp\a.txt",serializedObject);
15
16 int sum = MyLFlowCustomFunctions_v1002.MyCustomCommonMethods.Sum(5,9);
17
18 MyLFlowCustomFunctions_v1002.MyCustomCommonMethods.InstanceMessage(signalRUrl,sum.ToString());
19
20 MyLFlowCustomFunctions_v1002.MyCustomCommonMethods.FileAppend(@"c:\Temp\a.txt",sum.ToString());
21
22
```

Sına

Akış değişkenlerine varsayılan değer ataması yapılarak test ediliyor.  
Formül Başarı ile Çalıştırdı.

Akış değişkenlerine boş değer ataması yapılarak test ediliyor.  
Formül Başarı ile Çalıştırdı.

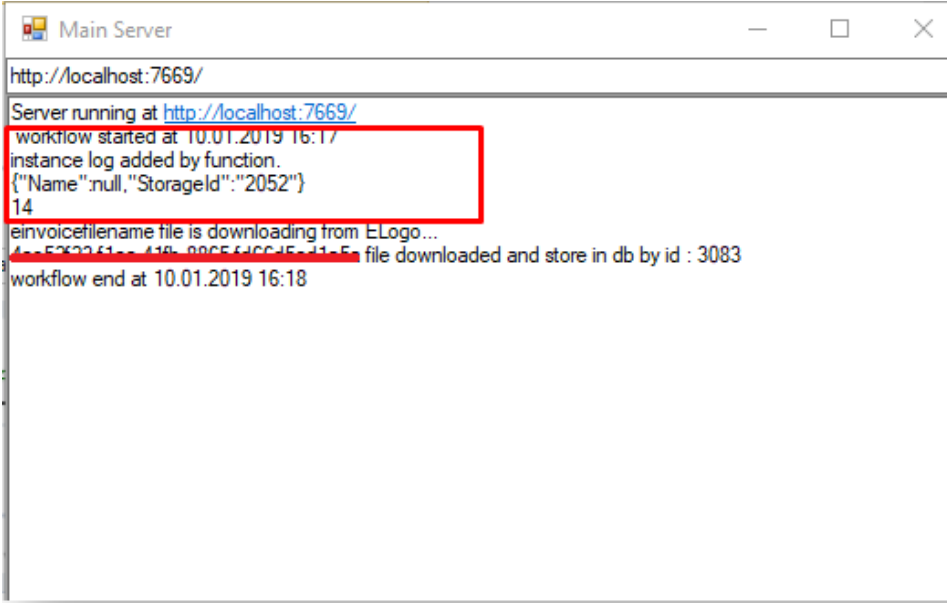
Tamam İptal

4. Yukarıdaki kod ile yapılan denemelerde dosya loglama, signalr ile anlık bilgi gönderilmesi denemeleri yapılmıştır. Çıktı görüntüleri ise aşağıdaki gibidir:

a.txt - Notepad

File Edit Format View Help

```
{"Name":null,"StorageId":"0"}14
```



Main Server

http://localhost:7669/

Server running at <http://localhost:7669/>

workflow started at 10.01.2019 16:17

instance log added by function.

{"Name":null,"StorageId":"2052"}

14

invoicefilename file is downloading from ELogo...

file downloaded and store in db by id : 3083

workflow end at 10.01.2019 16:18

## Örnek İndir

**Not :** Yukarıdaki örnekler için oluşturulmuş projeler için oluşturulan \*.nxm ve \*.lflowcustom dosyalarına aşağıdan ulaşabilirsiniz.

### Eklenti (Publisher Extension):



LFlow.Cus...nsion.rar

### Özelleştirilmiş Fonksiyon (\*.lflowfuncustom) dosyası:



MyLFlowC...nccustom

### Akış Paketi (\*.nxm) dosyası:



CA EInvoi...eader.nxm