

# Değişkenlerin Kodlama Bölümündeki Kullanım Örnekleri

- Değişkenlerin Değerini Alma
- SQL Sorgularında Değişkenlerin Kullanımı
- Değişkenlere Değer Atama
- Tablo Değişkeni Roslyn Editor Kullanımı
- Açılır Pencere (Pop-up) Değişkeni Roslyn Editor Kullanımı
- UBL Değişkeni Roslyn Editor Kullanımı
- Resim Değişkeni Roslyn Editör Kullanımı
- HTML Editör Değişkeni Roslyn Editör Kullanımı

Form tasarımında kullanılan tüm değişkenlerin özellikleri için [tıklayınız](#).

## Değişkenlerin Değerini Alma

Aşağıdaki tabloda değişken tiplerine göre değişkenlerin o anki değerlerinin nasıl alınması gerektiği gösterilmiştir.

Değişken Tipi	Değer Alma Formülü
Metin/Büyük Metin	metin.Value
Sayısal	sayi.Value.Value
Onay	onay.Value.Value
Açılır liste	Convert.ToString(liste.Value)
Tarih	tarih.Value.Value
Açılır Pencere	Pencere.Content.Items[0][Pencere_Columns.NAME].ToString()

## SQL Sorgularında Değişkenlerin Kullanımı

SQL İşleminde veya değişkenlerin kaynak seçimlerinde de formül editörü kullanılmaktadır. Aşağıda SQL sorgularında değişkenlerin değerlerinin nasıl alınacağını gösteren tablolar bulunmaktadır.

Değişken Tipi	SQL Sorgusunda Kullanımı
Metin/Büyük Metin	metin.Value. <b>toSql()</b>
Sayısal	sayi.Value.Value. <b>toSql()</b>
Onay	onay.Value.Value. <b>toSql()</b>
Açılır liste	liste.Value. <b>toSql()</b>
Tarih (SQL alanının tipi <b>DateTime</b> ise)	tarih.Value.Value. <b>toSqlDateTime()</b>
Tarih (SQL alanının tipi <b>DateTimeOffset</b> ise)	tarih.Value.Value. <b>toSql()</b>
Açılır Pencere	Pencere.Content.Items[0][Pencere_Columns.NAME]. <b>toSql()</b>

## Değişkenlere Değer Atama

Formül editörü içerisinde varolan değişkenler ad bilgileri ile kullanılır. Bir değişkene değer atamak için değişkenin Value özelliği kullanılmalıdır.

Değişken Tipi	Değer Atama Formülü
Metin/Büyük Metin	metin.Value
Sayısal	sayi.Value
Onay	onay.Value
Açılır liste	liste.Value
Tarih	tarih.Value

### Örnek 1

```
metin.Value="deneme";
sayi.Value=35;
liste.Value="aa";
onay.Value=true;
tarih.Value=DateTimeOffset.UtcNow;
```

### Örnek 2

1'den sayi1 değişkenine girilen sayıya kadar olan sayıların toplamını sayi2 değişkenine atama

```
int toplam=0;
for(int i=1;i<sayi1.Value.Value;i++)
{
    toplam=toplam+i;
}
sayi2.Value=toplam;
```

### Örnek 3

Saat değişkenine anın saat, dakika ve saniye bilgisini atama

```
int saat = DateTime.Now.Hour;
int dakika = DateTime.Now.Minute;
int saniye = DateTime.Now.Second;
Saat.Value= LFlowUtils.GetTime(saat,dakika,saniye);
```

## Tablo Değişkeni Roslyn Editor Kullanımı

1) Tabloda tüm satırlardaki sayısal bir kolonun (Örneğin, Tutar kolonu) toplamının alınması için *Sum* fonksiyonu kullanılabilir.

```
toplam.Value =tablo.Sum(OrnekTablo_Columns.tutar);
```

2) Tablodaki istenilen bir kolonun Min, Max değerlerinin alınması için *Min/Max* fonksiyonları kullanılabilir.

```
minDeger.Value=Convert.ToInt32(tablo.Min(OrnekTablo_Columns.tutar));
maxDeger.Value=Convert.ToInt32(tablo.Max(OrnekTablo_Columns.tutar));
```

3) Tablo içeriğinin boş olup olmadığının kontrolü için aşağıdaki formül kullanılabilir.

*if(OrnekTablo.Content!=null && OrnekTablo.Content.Rows!=null && OrnekTablo.Content.Rows.Count>0)*

4) Tablodaki toplam satır sayısının alınması için aşağıdaki gibi bir formül kullanılabilir.

*ToplamSatirSayisi.Value=OrnekTablo.Content.Items.Count;*

5) Tablo satırlarını silmek için aşağıdaki formül kullanılabilir.

*OrnekTablo .Content.Items.Clear();*

6) Tabloda herhangi bir kolonun dolu olup olmadığının kontrolü için aşağıdaki formül kullanılabilir.

*OrnekTablo.GetValue(OrnekTablo\_Columns.Teklif1, true,satirnumarası,"CODE").Assigned()*

7) Tablodaki herhangi bir kolonun değerini almak için aşağıdaki formülü kullanılabilir.

Eğer değeri alınmak istenen kolon 'Açılır Pencere' tipinde ise;

*Convert.ToString(OrnekTablo.GetValue(OrnekTablo\_Columns.Teklif1, true,satirnumarası,"Açılır pencere kolon adı"));*

Açılır Pencere dışındaki kolon tiplerinde ise;

*Convert.ToString(TalepSatirlar.GetValue(OrnekTablo\_Columns.Aciklama, true,satirnumarası));*

8) Tablonun kolonları arasında hesaplama yaparak sonucu bir başka kolona yazdırmak için; Hesaplamanın yapılmasının istendiği kolona aşağıdaki örneğe benzer şekilde kod yazılarak tutar ve adet çarpımı total alanına yazılır.

*double tutar = Convert.ToDouble(OrnekTablo.GetValue(OrnekTablo\_Columns.Tutar));  
double adet = Convert.ToDouble(OrnekTablo.GetValue(OrnekTablo\_Columns.Adet));  
double result = tutar \* adet;  
OrnekTablo.SetValue(OrnekTablo\_Columns.Total, result);*

9) Tablodaki herhangi bir kolonunun refresh edilmesi için aşağıdaki komut kullanılabilir.

*OrnekTablo.RefreshWithColumn(Context,OrnekTablo\_Columns.Metin);*

10) Tablonun satırları arasında dolaşmak için while aktivitesi kullanılabilir. Aşağıda bu işlem için gereken adımlar bulunmaktadır. Aktivitenin detaylı kullanım örneği için [tıklayınız](#).

While aktivitesinin koşul bölümünde aşağıdaki komut yazılmalıdır.

*OrnekTablo.Content.CheckIsLast();*

While içerisinde Flowchart aktivitesi konularak çeşitli işlemler yapılabilir. Flowchart içerisinde ise, Kodlama aktivitesi kullanılarak aşağıdaki formüller ile tablonun o satırının istenilen kolonuna erişilebilir.

*OrnekKolonDegeri.Value= Convert.ToString(OrnekTablo.Content.CounterRow[OrnekTablo\_Columns.Kod]);*

Bir sonraki tablo satırına geçilmesi için, istenilen işlemler yapıldıktan sonra tekrar bir Kodlama aktivitesi eklenerek aşağıdaki formül yazılabilir.

*OrnekTablo.IncreaseCounter();*

11) Tablonun satırlarını kod ile doldurmak için aşağıdaki formül kullanılabilir.

```
Dictionary<string, object> satir = new Dictionary<string, object>();  
satir.Add(OrnekTablo_Columns.CariKod,"atanacak değer");  
OrnekTablo.Content.AddRow(satir);
```

*Tablo kolonu açılır pencere ise;*

```
System.Collections.Generic.Dictionary<string, object> acilirpenceredeger= new System.Collections.  
Generic.Dictionary<string, object>();
```

```
acilirpenceredeger.Add("CODE", code);
```

```
acilirpenceredeger.Add("NAME", name);
```

*//açılır pencere değişken değerini açılır pencere değişkenine ekle*

```
NPopupVariableValue popUpVal = new NPopupVariableValue();
```

```
popUpVal.Items.Add(acilirpencere);
```

*//tabloya satır değişkenine ekleme bölümü*

```
System.Collections.Generic.Dictionary<string, object> Satir= new System.Collections.Generic.  
Dictionary<string, object>();
```

```
Satir.Add(OrnekTablo_Columns.acilirpencere,popUpVal);
```

*//tabloya satır değişkenini tabloya ekle*

```
UretilecekUrunler.Content.Items.Add(dictTableRow);
```

**12)** Tablonun herhangi bir kolonunun özelliğine(salt okunur, görünür) müdahale etmek için aşağıdaki formül yapısı kullanılabilir. Bu formüller tüm roslyn bölümleri üzerinde yazılabilmektedir. Örneğin; formu yüklerken veya bir değişkenin değer değiştiğinde eventinde vs.

Tablo kolonunun görünürlük ayarı için;

```
Tablo.GetColumn(Tablo_Columns.GizlenecekKolon).Visibility=false;
```

Tablo kolonunu müdahaleye kapatmak için;

```
Tablo.GetColumn(Tablo_Columns.SaltOkunurOlacakKolon).ReadOnly=true;
```

## Açılır Pencere (Pop-up) Değişkeni Roslyn Editor Kullanımı

1) Açılır pencere değişkenini kod ile doldurmak için aşağıdaki formül kullanılabilir.

```
System.Collections.Generic.Dictionary<string, object> OrnekAcilirPencereDegeri = new System.  
Collections.Generic.Dictionary<string, object>();  
OrnekAcilirPencereDegeri.Add("ID",1);  
OrnekAcilirPencereDegeri.Add("AD", "Deneme Değer");  
OrnekAcilirPencere.Content.Items.Add(OrnekAcilirPencereDegeri);
```

2) Açılır pencere değişkeninin içeriğini silmek için aşağıdaki formül kullanılabilir.

```
OrnekAcilirPencere.Content.Items.Clear();
```

3) Form içerisinde kullanılan açılır pencereden çoklu seçim yapıldığında seçilen verilere aşağıdaki örnekteki gibi ulaşılabilir.

```
string SeciliDegerler="";  
if(OrnekAcilirPencere.Content!=null)  
{  
    for(int i=0;i<OrnekAcilirPencere.Content.Items.Count;i++)  
        SeciliDegerler += OrnekAcilirPencere.Content.Items[i][OrnekAcilirPencere_Columns.AD].ToString()  
        + " ";  
}
```

4) Tablo içerisinde kullanılan açılır pencereden çoklu seçim yapıldığında seçilen verilere aşağıdaki örnekteki gibi ulaşılabilir.

```
string SeciliDegerler="";  
for(int i=0;i<tablo.GetValue(tablo_Columns.OrnekAcilirPencere).CPopUpValue().Items.Count;i++)  
{  
    SeciliDegerler+=Convert.ToString(tablo.GetValue(tablo_Columns.OrnekAcilirPencere).CPopUpValue().  
Items[i]["NAME"])+",";  
}
```

## UBL Değişkeni Roslyn Editor Kullanımı

Ubl değişkeni için kullanılabileceğiniz metotlar aşağıda listelenmiştir. Bu metotlar kullanılarak ubl değişkenine xml veri atanıp, atanan içerikteki değerlere parse metotları yardımıyla erişilebilmektedir.

Akış tasarım aktivitelerinden; *UBL Eşleştirme* aktivitesi yardımıyla da UBL değişkenindeki değerler; form üzerindeki diğer değişkenlerle eşleştirilebilmektedir. **bnz: UBL Eşleştirme Aktivitesi**

- **string GetXml()** : Xml veriyi döndürür
- **void Clear()** : İçerik değerini temizler.
- **void SetXml(string xmlData)**: Xml(Ubl formatında) ataması yapar.
- **void SetXmlFromZipData(string base64ZipData)** : base64 formatında dönüştürülmüş bir zip dosyası içerisindeki xml dosya atması yapar.
- **object ParseAsObject(string propertyPath)**: property path olarak verilen değeri nesne içerisindeki değerine object olarak döndürür.
- **double ParseAsDouble(string propertyPath, double defaultValue = 0)**: property path olarak verilen değeri nesne içerisindeki değerine double olarak döndürür.
- **double ParseAsInt(string propertyPath, int defaultValue = 0)**: property path olarak verilen değeri nesne içerisindeki değerine integer olarak döndürür.
- **string ParseAsString(string propertyPath, string defaultValue = "")**: property path olarak verilen değeri nesne içerisindeki değerine string olarak döndürür.
- **DateTime ParseAsDateTime(string propertyPath)**: property path olarak verilen değeri nesne içerisindeki değerine datetime olarak döndürür.
- **IList<object> ParseAsList(string propertyPath)**: property path olarak verilen değeri nesne içerisindeki değerine IList olarak döndürür.
- **IList<UblType> ParseAsList<UblType>(string propertyPath)**: property path olarak verilen değeri nesne içerisindeki değerine IList olarak döndürür.
- **InvoiceType GetInvoiceData()**: Xml veriyi fatura tipine dönüştürür.

UBL değişkeni için oluşturulan metotların kullanıldığı örnek akış tasarımı ([UblOrnek.nxm](#)), tasarım içerisinde ihtiyaç duyulan xml içerik ([UblSample.xml](#)) ve pdf ([UblSample.pdf](#)) bulunmaktadır.

Akış, *xml* ve *pdf* dosyaları **C:\Temp** klasörüne kopyaladıktan sonra çalıştırılmalıdır.

e-İrsaliye akışında kullanılmak üzere oluşturulan UBL değişkeni için roslyn editör üzerinden nesne tabanlı veri alma işlemi de gerçekleştirilebilmektedir.

- **GetWayBillData()** : UBL değişkeninde tutulan değeri *DespatchAdviceType* tipinde irsaliye nesnesine dönüştürür.

GetWayBillData metodunun kullanıldığı örnek akış tasarımı ([UBLFaturaIRS.nxm](#)) ve tasarım içerisinde ihtiyaç duyulan xml içerik ([EWayBillSample.xml](#)) bulunmaktadır.

Akış, *xml* **C:\Temp** klasörüne kopyaladıktan sonra çalıştırılmalıdır.

## Resim Değişkeni Roslyn Editör Kullanımı

Tablodaki ve forma koyulan resim değişkeni veri kaynağı üzerinden de dinamik olarak doldurulabilmektedir. Örnek için [tıklayınız](#).

- 1) Resim değişkeni içerisindeki data base64string olarak alınabilir.

```
string base64Data = ResimDegiskeni.GetImageAsBase64();
```

- 2) Resim değişkeni içerisindeki data byte[] olarak alınabilir.

```
byte[] dataImage = ResimDegiskeni.GetImageAsByteArray();
```

- 3) Resim değişkenine base64string formatındaki değeri atamak için kullanılır.

```
SetImage(string imageBase64Data, string ImageName, bool changeSize)
```

*imagename*: uzantı içerecek şekilde gönderilmelidir.

*changesize*: resmin gerçek boyutu form üzerinde gösterilmek istenirse *true*; form tasarımındaki değişken boyutunda gösterilmek istenirse *false* gönderilmelidir.

```
YeniResimDegiskeni.SetImage(base64Data,"Resim.png",false);
```

- 4) Resim değişkenine byte[] formatındaki değeri atamak için kullanılır.

```
SetValue(byte[] data, string ImageName)
```

*imagename*: uzantı içerecek şekilde gönderilmelidir.

```
YeniResimDegiskeni.SetValue(dataImage,"Resim.png");
```

- 5) Tablo kolonundaki resim değişkenine değer atamak için kullanılır. Herhangi bir kolonun değer değiştiğinde yazılabilir.

```
TabloAdı.SetValue(Tablo kolonu, base64 türünde veri, tablonun aktif satır bilgisi)
```

Byte[] data; *herhangi bir yerden getirilebilir*.

```
string base64Data = Convert.ToBase64String(data, 0, data.Length);
```

```
Tablo.SetValue(TabloInlineRoslyn_Columns.Resim ,base64Data,Tablo.ActiveRowid);
```

## HTML Editör Değişkeni Roslyn Editör Kullanımı

1) Word belgesini HTML Editör değişkeninde göstermek için aşağıdaki metotlar kullanılır.

```
htmleditor.SetDocx(string base64)
```

```
htmleditor.SetDocx(byte[] docx)
```

2) HTML Editör değişkenine girilen veya set edilen veriyi docx formatına dönüştürmek için kullanılır.

```
htmleditor.ExportAsDocs()
```

3) Byte[] tipinde bir word belgesini html string e dönüştürmek için kullanılır.

```
string html= LFlowUtils .ConvertDocxToHtml(byte[] docx)
```

4) String olarak verilen html içeriği byte [] tipinde bir word belgesine dönüştürmek için kullanılır.

```
byte[] docx=LFlowUtils .ConvertHtmlToDocx(string html)
```

5) String olarak verilen html içeriği byte [] tipinde bir pdf belgesine dönüştürmek için kullanılır.

```
byte[] pdf= LFlowUtils .ConvertHtmlToPdf(string html)
```

6) HTML Editör üzerindeki özelliklere form yüklendiğinde müdahale edebilmek için kullanılır.

```
htmlEditor.Tools="bold,italic,underline,undo,redo,striethrough,unlink,tableWizard,createLink";
```

7) HTML Editör içeriğinin render edilerek html olarak gösterilmesi için kullanılır.

```
htmlEditor.IsShowHtml = true;
```