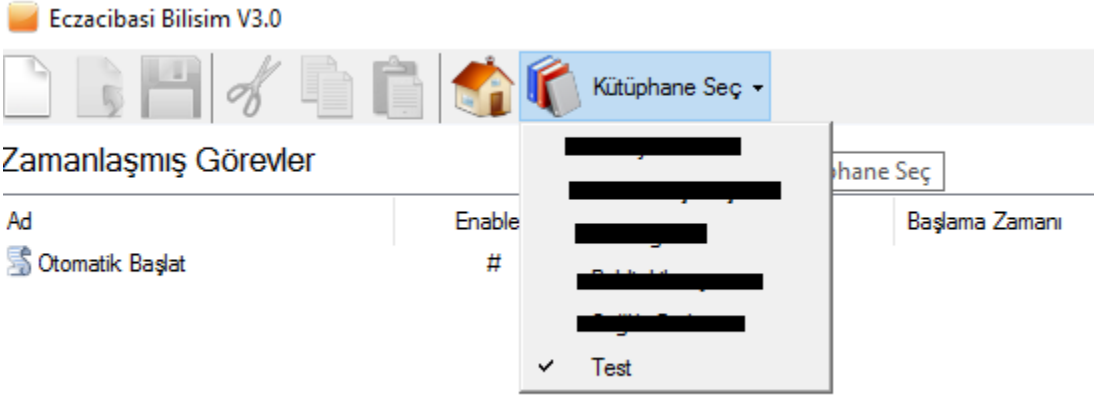


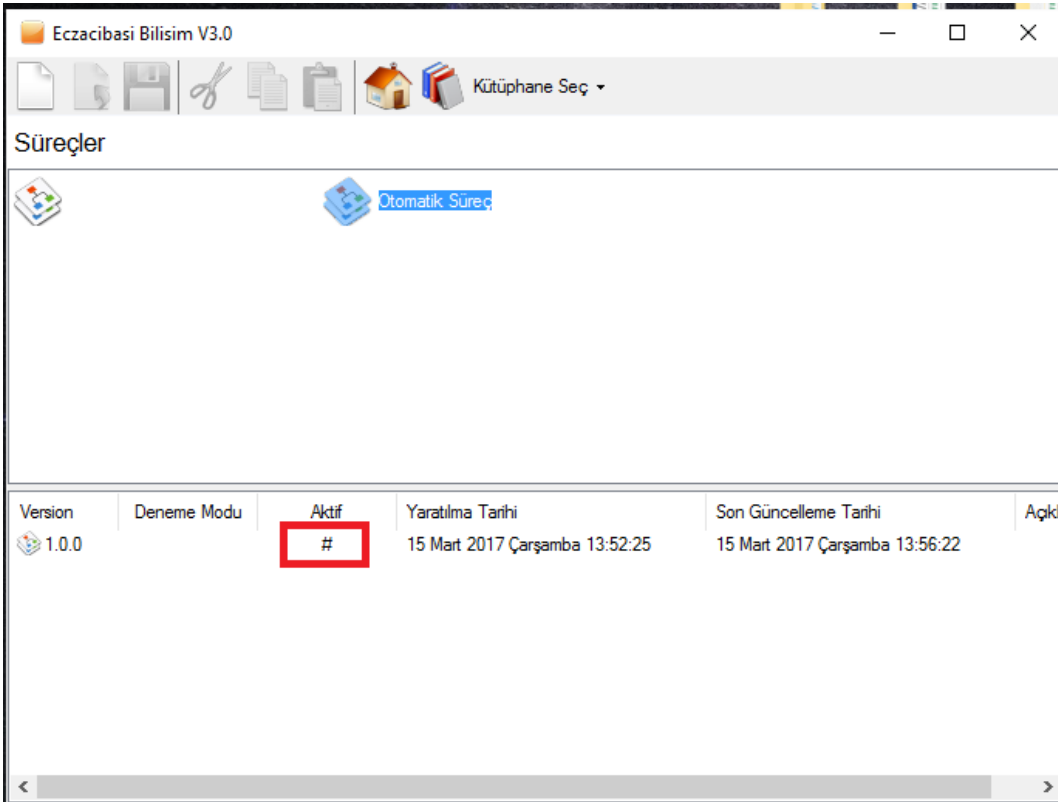
Zamanlanm görev ile otomatik süreç balatma nasıl yapılır?

Zamanlanm Görevler Ebiflow içerisinde belirli periyotlarla belirli işlemlerin gerçekleştirilmesi için kullanılmaktadır.

Zamanlanm Görevler penceresinde süreçlerde olduğu gibi kitaplıklara göre filtreleme vardır. Her kitaplık kendisi için oluşturulmuş görevleri içerir.



Zamanlanm Görevler içerisinde bir görev yaratmadan önce otomatik olarak balatılacak süreci hazırlayalım. Örnek olarak oluşturduğumuz sürecin adı "Otomatik Süreç". Sürecin zamanlanm görev ile çalışabilmesi için aktif moda olduğundan emin olmanız gerekiyor. Deneme modundaki süreçleri zamanlanm görevlerden balatmak isterseniz sürecin bulunamadına dair bir hata mesajı dönecektir.



Otomatik olarak süreç balatmak için Modül tipinde bir adm eklememiz gerekiyor. Eklediğimiz modülün ismini daha sonra zamanlanm görev içerisinde kullanacağız.

Otomatik Süreç .0.0 Süreç ismini zamanlanmış görev içerisinde kullanacağız.

Modül ismini zamanlanmış görev içerisinde kullanacağız.

Yeni

Otomatik Başlat

XmlScript

Tamam

Kontrol

Kontrol Kullanıcısı

Bitir

Süreç Ağacı

Workflows

Yeni

Roller

Initiator

Kontrol Kullanıcısı

Fomlar

Form

Mobile Forms

Methods

Form Versi

Localization

Adımlar

İş Başlatıcı

Normal

Modül

Alt Akış

Zamanlayıcı

Bilgilendirme

Kontroller

Temel Kontroller

Metin Alanı

Gelişmiş Metin Alanı

Sayısal Alan

Sayısal Metin Alanı

Maskeli Metin Alanı

Tarih Alanı

Açılan Liste

Kontrol Özellikleri

Modülün özelliklerini açtığımız zaman dikkat etmemiz gereken konfigürasyon yolunun Türkçe ibare içermemesidir. "Success Action" alanının mutlaka modülün sonuna eklenen işlem ile aynı isimde olmasına dikkat edilmelidir. Bu örnekte işlem adı "Tamam" olarak oluşturulduğu için "Success Action" alanı da "Tamam" olarak dolduruldu.

Dikkat

Script sekmesinde herhangi bir işlem yapmamıza gerek bulunmuyor. Ancak istenirse süreç öncesinde yapılacak istenilen işlemler de burada çalıştırılabilir. Burada dikkat edilmesi gereken eğer hata alırsa "Failure Action" ve "Exception" alanları için bir işlem tanımlanmalı ve hata denetimi yapılmalıdır.

Modül Adımı Özellikleri

Adı: Otomatik Başlat

Çoklu İşleme

Seçenekler: Script, Notlar

Configuration: /formData/OtomatikBaslat

Yeni...

SuccessAction (Tamam)

FailureAction

Exception

Oluşturduğumuz konfigürasyon verisi yalnızca İngilizce karakter içermelidir ve boşluk olmamalıdır.

Tamam Vazgeç Uygula

Zamanlanm Görevler penceresinden yeni bir zamanlanm görev oluturuyoruz.

Sürekli İşlem

Genel Script

Adı Otomatik Başlat

Kullanımda

Açıklama

Takvim

Başlangıç Tarihi 15 Mart 2017 Çarşamba 14:00

Bitiş Tarihi 15 Mart 2017 Çarşamba 13:

Tekrar adedi 0

Devam süresi 00:30:00

Tamam Vazgeç Uygula

Özellik	Açıklama
Ad	Zamanlanm görevin ad
Açıklama	Zamanlanm görev açıklamas

Takvim

Özellik	Açıklama												
Başlangıç Tarihi	Zamanlanmış görevin ilk çalışma zamanı												
Biti Tarihi	Zamanlanmış görevin çalışabileceği en son zaman												
Tekrar Adedi	Devam süresi ile belirlenen periyotta kaç adet çalışacağı bilgisi. "0" sonsuz sayıda çalışmayı temsil eder.												
Devam Süresi	Zamanlanmış görevin periyot bilgisi. <table border="1"><thead><tr><th>Giri Verisi</th><th>Açıklama</th></tr></thead><tbody><tr><td>7.00:00:00</td><td>Haftalık periyot</td></tr><tr><td>1.00:00:00</td><td>Günlük periyot</td></tr><tr><td>00:30:00</td><td>Yarı saatlik periyot</td></tr><tr><td>00:01:00</td><td>Dakikalık periyot</td></tr><tr><td>00:00:01</td><td>Saniyelik periyot</td></tr></tbody></table> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">Dikkat Zamanlanmış görevlerin çalışma sürelerinin uzaması durumunda periyotlarda sapmalar olabilir. Bu nedenle 1 dakikadan altındaki periyotlar tavsiye edilmemektedir.</div>	Giri Verisi	Açıklama	7.00:00:00	Haftalık periyot	1.00:00:00	Günlük periyot	00:30:00	Yarı saatlik periyot	00:01:00	Dakikalık periyot	00:00:01	Saniyelik periyot
Giri Verisi	Açıklama												
7.00:00:00	Haftalık periyot												
1.00:00:00	Günlük periyot												
00:30:00	Yarı saatlik periyot												
00:01:00	Dakikalık periyot												
00:00:01	Saniyelik periyot												

Zamanlanmış görevin Script sekmesinde StartWorkflow fonksiyonu ile istenilen süreç başlatılabilir. Aşağıdaki örnekte System Admin kullanıcıları ile süreç başlatılıyor.

Dikkat

Süreci başlatacak olan kişinin süreci başlatma yetkisinin olması gerekmektedir.

Sürekli İşlem

Genel Script

İş Adımı

Tasarım Kod Editörü

```
<Project Name="">
<StartWorkflow Library="Test" Process="Otomatik Süreç" Task="Otomatik Başlat" UseProcessData="True" ProcessCulture="">
  <Initiator>
    <FindUser Property="RegistryNumber">
      <Value Culture="" Target="None">99999999</Value>
    </FindUser>
  </Initiator>
  <Perform />
</StartWorkflow>
</Project>
```

Tamam Vazgeç Uygula

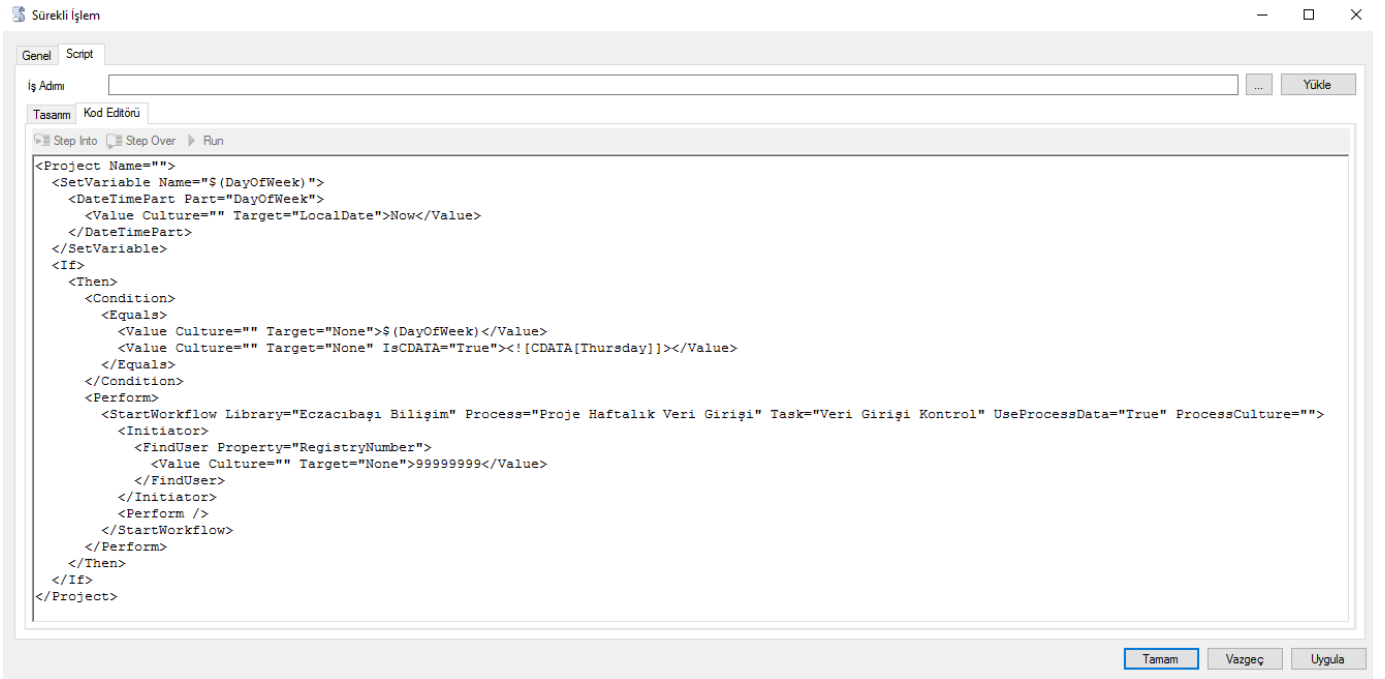
Özellik

Açıklama

Library	Sürecin olduğu kitapın ad.
Process	Balatacak sürecin ad.
Task	Balatacak süreçteki modülün ad.
UseProcessData	
ProcessCulture	
Initiator	Süreci balatacak olan kullanıcı kimliği.
Perform	Balatacak akta istenilen alanların tanımlaması gerçekleştirilir.

Örnekler

Haftanın bazı günlerinde çalışacak bir zamanlanmış görev örneği.



The screenshot shows a workflow editor window titled "Sürekli İşlem". The main area displays XML code for a project named "Sürekli İşlem". The code defines a workflow that runs on a specific day of the week (Thursday) and performs a task named "Veri Girişi Kontrol".

```
<Project Name="">
  <SetVariable Name=""$(DayOfWeek)">
    <DateTimePart Part="DayOfWeek">
      <Value Culture="" Target="LocalDate">Now</Value>
    </DateTimePart>
  </SetVariable>
  <If>
    <Then>
      <Condition>
        <Equals>
          <Value Culture="" Target="None"$(DayOfWeek)</Value>
          <Value Culture="" Target="None" IsCDATA="True"><![CDATA[Thursday]]</Value>
        </Equals>
      </Condition>
      <Perform>
        <StartWorkflow Library="Eczacıbaşı Bilişim" Process="Proje Haftalık Veri Girişi" Task="Veri Girişi Kontrol" UseProcessData="True" ProcessCulture="">
          <Initiator>
            <FindUser Property="RegistryNumber">
              <Value Culture="" Target="None">999999999</Value>
            </FindUser>
          </Initiator>
          <Perform />
        </StartWorkflow>
      </Perform>
    </Then>
  </If>
</Project>
```

The interface includes a "Genel" tab, a "Script" tab, and a "Yükle" button. The main area has a "Tasarım" tab and a "Kod Editörü" tab. The "Kod Editörü" tab is active, showing the XML code. The "Run" button is visible in the top right corner of the editor area. The bottom of the window has "Tamam", "Vazgeç", and "Uygula" buttons.

Haftann belirli bir günü çalacak görev

```
<Project Name="">
  <SetVariable Name="$(DayOfWeek)">
    <DateTimePart Part="DayOfWeek">
      <Value Culture="" Target="LocalDate">Now</Value>
    </DateTimePart>
  </SetVariable>
  <If>
    <Then>
      <Condition>
        <Equals>
          <Value Culture="" Target="None">$(DayOfWeek)</Value>
          <Value Culture="" Target="None"
IsCDATA="True"><![CDATA[Thursday]]></Value>
        </Equals>
      </Condition>
      <Perform>
        <StartWorkflow Library="Eczacba Biliim" Process="Proje Haftalk Veri
Girii" Task="Veri Girii Kontrol" UseProcessData="True" ProcessCulture="">
          <Initiator>
            <FindUser Property="RegistryNumber">
              <Value Culture="" Target="None">99999999</Value>
            </FindUser>
          </Initiator>
          <Perform />
        </StartWorkflow>
      </Perform>
    </Then>
  </If>
</Project>
```

Ayrca Baknz

StartWorkFlow

Zamanlanm Görevler

If

SetVariable

DateTimePart

Value