

**LOGO** **PLATFORM TAILOR**

## İÇİNDEKİLER

İÇİNDEKİLER.....	2
<b>1. ÇALIŞMAYA BAŞLARKEN.....</b>	<b>3</b>
1.1. LOGO PLATFORM TAILOR KULLANIM AMACI .....	3
<b>2. PROJE OLUŞTURMA.....</b>	<b>3</b>
2.1. YENİ PROJE OLUŞTURMA ADIMLARI	
<b>Error! Bookmark not defined.</b>	
2.2. PROJE AYARLARI	
<b>Error! Bookmark not defined.</b>	
2.3. BAĞLANTI AYARLARI .....	6
<b>3. PROJEDE YERALAN KLASÖRLER VE KULLANIMLARI.....</b>	<b>7</b>
3.1. TABLOLAR .....	7
3.2. NESNELER.....	9
3.2.1. YENİ BİR NESNE OLUŞTURMAK .....	<b>Error! Bookmark not defined.</b>
3.2.2. MEVCUT BİR ÜRÜN NESNESİNİ GENİŞLETMEK .....	<b>Error! Bookmark not defined.</b>
3.3. SORGULAR .....	14
3.4. FORMLAR .....	17
3.4.1. YENİ FORM YARATMAK .....	18
3.4.2. MEVCUT ÜRÜN FORMUNU UYARLAMAK.....	25
3.5. RAPORLAR.....	29
3.6. AĞAÇLAR.....	34
3.7. PROFİLLER .....	36
<b>4. DEPLOYMENT.....</b>	<b>39</b>
4.1. ECLIPSE ÜZERİNDE DEPLOY YAPMAK.....	39
4.2. ÜRÜN İÇERİSİNDEN DEPLOY YAPMAK .....	40
4.3. DEPLOY SONRASI İŞLEMLER.....	43

## 1. ÇALIŞMAYA BAŞLARKEN

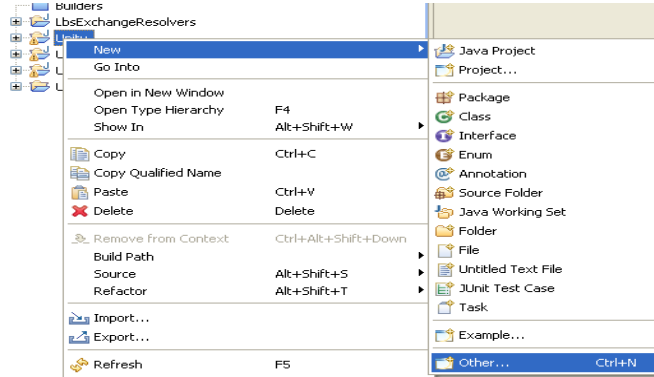
### 1.1. Logo Platform Tailor kullanım amacı

- *j-guar* ürününde kullanılan ağaç yapısı özelleştirilebilir, yeni modüller eklenebilir.
- *j-guar* ara yüzüne yeni formlar eklenebilir, mevcut formlar üzerine alanlar eklenebilir.
- Yeni sorgular eklenebilir.
- Yeni raporlar eklenebilir, mevcut raporlar genişletilebilir.
- *j-guar*'in başka sistemlerle entegrasyonu sağlanabilir. Bu amaçla standart Web servislerden yararlanabileceği gibi yeni Web Servisler de eklenebilir
- Excellent kullanımında projeye özel yeni metotlar eklenebilir.
- Yeni batch işlemler oluşturulabilmektedir.

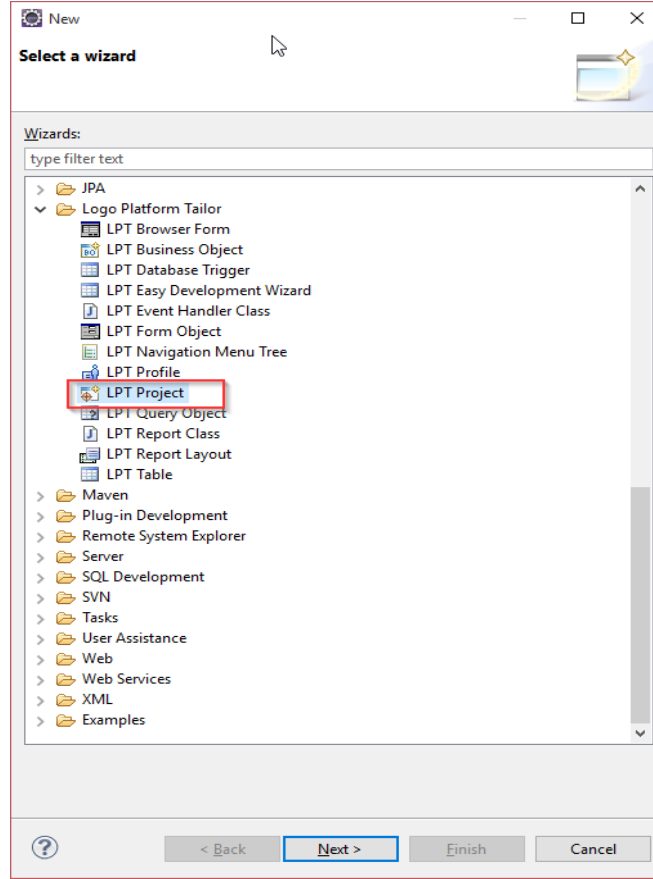
## 2. PROJE OLUŞTURMA

### 2.1. Yeni proje oluşturma adımları

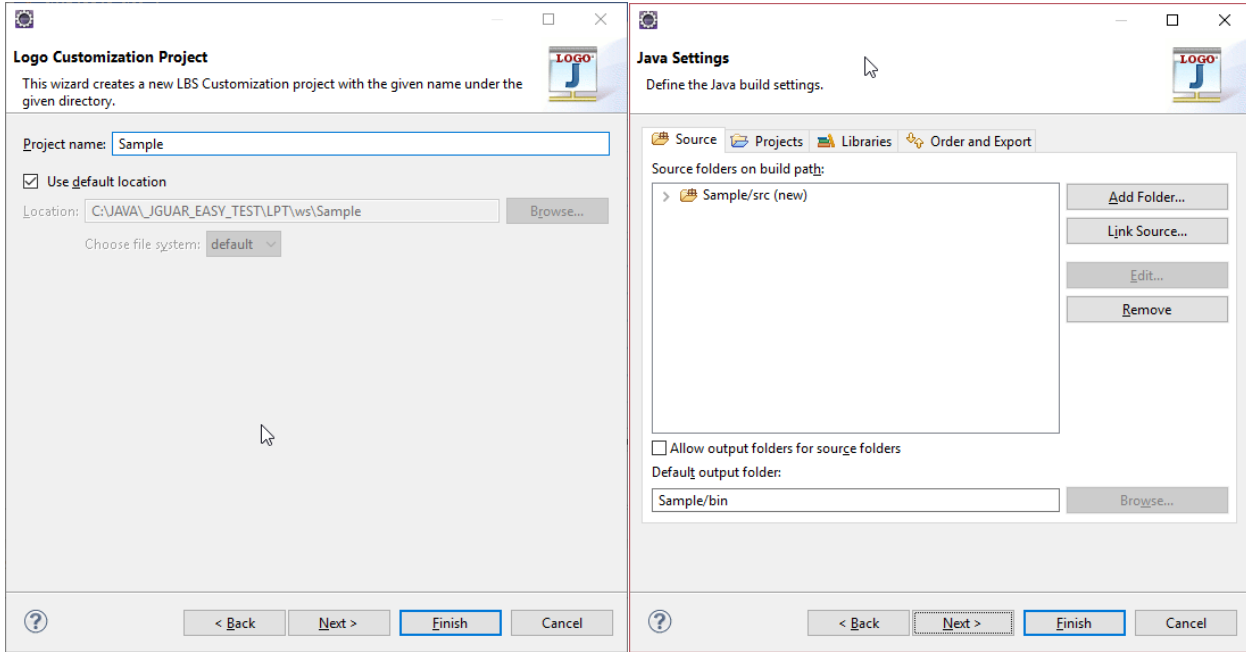
Uyarlama yapabilmek için ilk adım yeni bir uyarlama projesi oluşturmaktır. LPT'de proje oluşturmak için "Logo Customization Project" sihirbazı kullanılmaktadır. Package Explorer üzerinde sağ click ile açılacak olan listede önce "New" ardından da "Other" seçilir.

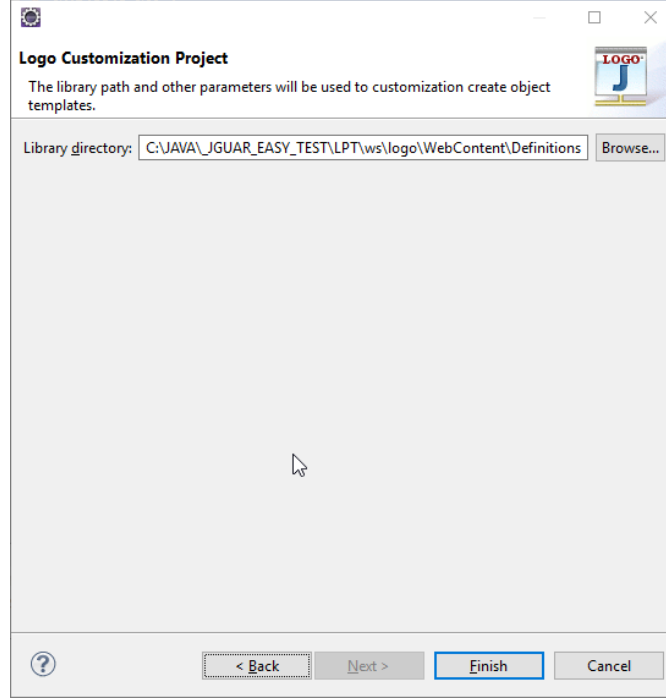


Gelen ekranda "Logo Platform Tailor" klasörünün altında uyarlama için kullanılacak sihirbazlar listelenmiştir, bu listeden proje oluşturma sihirbazının seçilmesi gerekmektedir.



Bu işlemin ardından sihirbaz üzerinde oluşturulmak istenilen uyarılama projesinin adı yazılarak devam edilir.

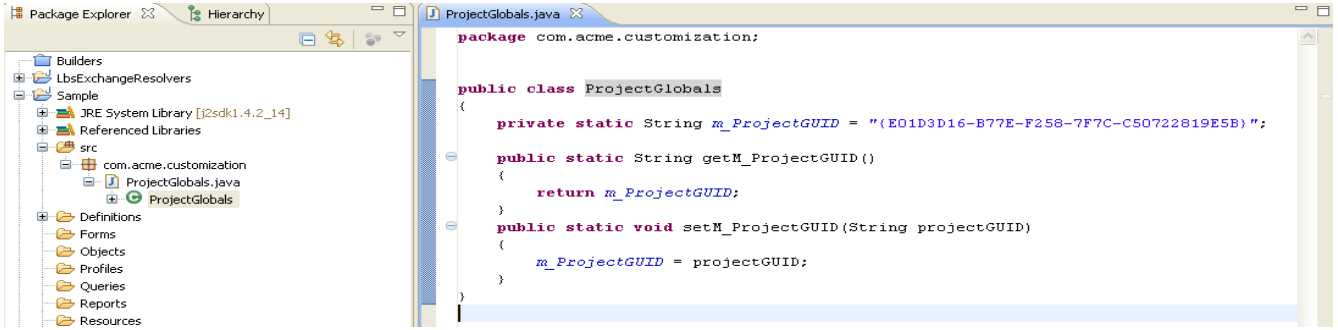




\* LbsDefinitions.jar j-guar ürününe ilişkin tüm nesne tanımlarını içeren ve versiyon bağımlı jar dosyasıdır.

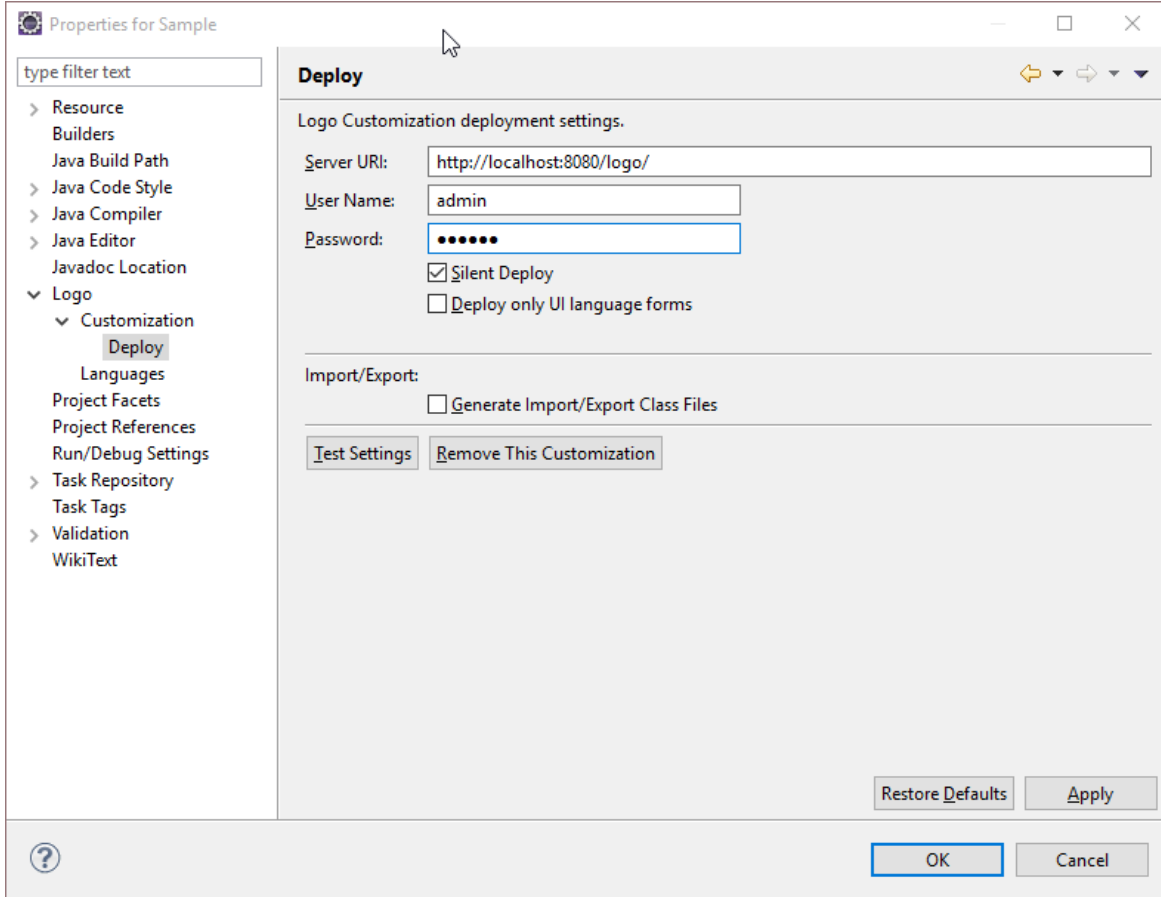
## 2.2. Proje Ayarları

Project GUID alanında gördüğümüz değer proje'nin tekil kimliğini (unique ID) ifade etmektedir. Bu değere kod yazılırken ihtiyaç duyulacağı için proje içerisindeki SRC klasörünün içinde **"ProjectGlobals"** adlı bir dosyada tutulur.



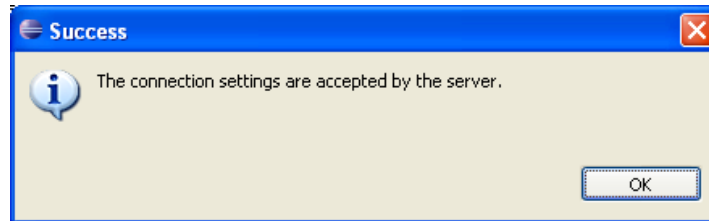
## 2.3. Bağlantı ayarları

Uyarlama Projesinin, J-GUAR ürünüyle bağlantı kurabilmesi amacıyla bağlantı ayarlarının yapılması gerekmektedir. Proje üzerinde **sağ klik** yapılarak **“Properties”** seçilir, gelen proje özellikleri penceresinin solundaki **“Logo”** ve **“Customization”** seçenekleri genişletilerek **“Deploy”** özelliğine tıklanır. İlk açılışta bağlantı bilgileri boş olarak gelmektedir, **“Restore Defaults”** düğmesi tıklanarak varsayılan bağlantı ayarları getirilebilir. Kullanıcı dilediği gibi bu ayarları düzenleyerek **“Apply”** düğmesini tıklar.



Ayrıca kullanıcı tanımlanan ayarlarla bağlantının kurulup kurulmadığını **“Test Settings”** düğmesine tıklayarak sınavabilir, bağlantının durumuyla ilgili kullanıcıya uyarı mesajı verilmektedir.

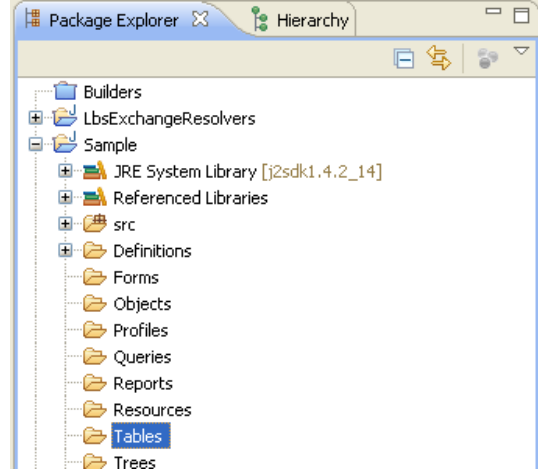
**Not:** Bağlantı ayarının sınavabilmesi için server'ın açık olması gerekmektedir.



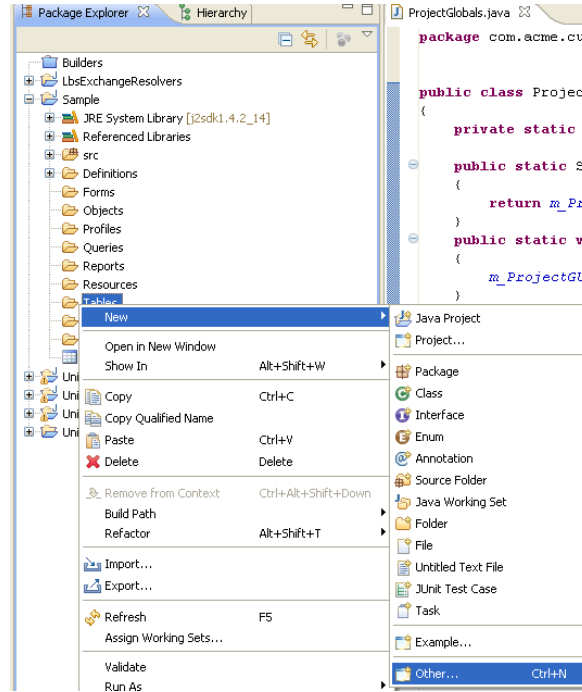
## 3. PROJEDE YERALAN KLASÖRLER VE KULLANIMLARI

### 3.1. Tablolar

Tablolar uyarlama projesi içerisinde bulunan “Tables” klasöründe yer almaktadır.

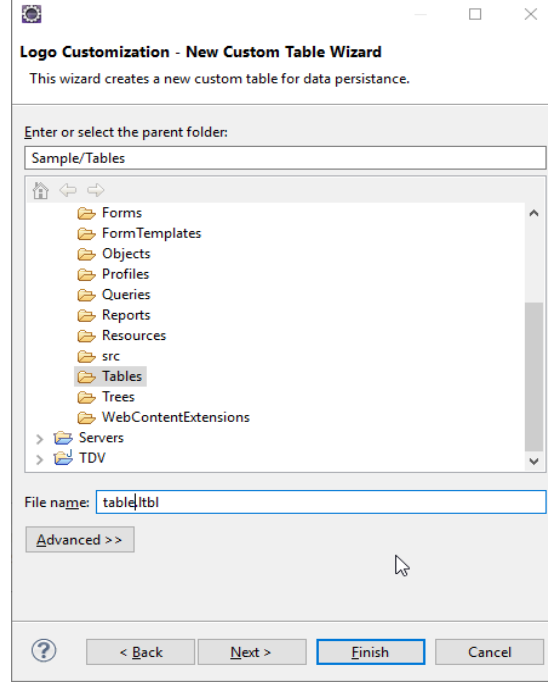
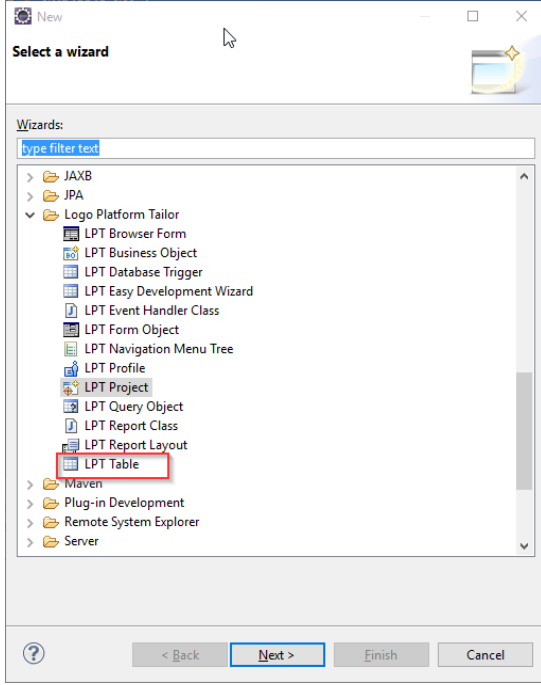


Tabloların nasıl yaratıldığını bir örnek üzerinde inceleyelim, örneğimizde makine tanımlamak için kullanılmak üzere tablo tanımı yapılmak istenmektedir, buna göre tables klasörü üzerinde sağ click yaparak menu'den “New” ardından da “Other”ı seçerek sihirbazların yer aldığı ekranı açalım.

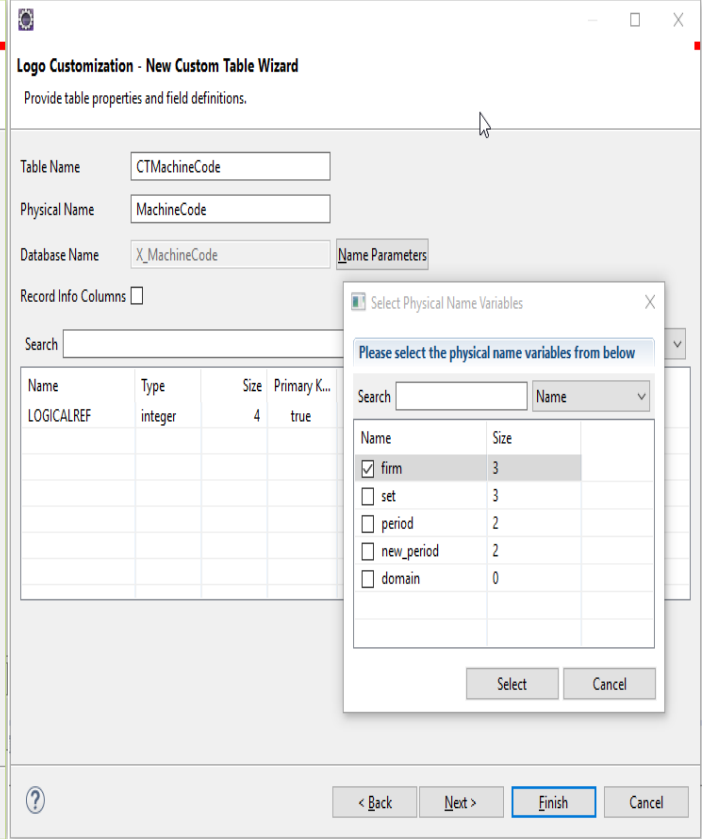
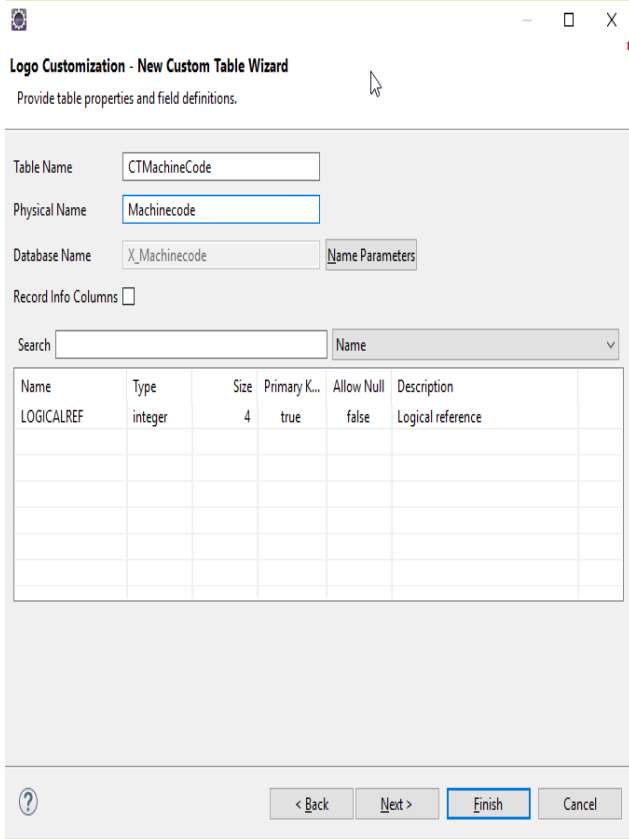


Açılan ekranda “Custom Table” seçilerek next düğmesi yardımıyla ilerleyelim gelen ekranda tablomuz için isim vermemiz gerekmektedir.

**Not:** “Bu aşamada verilen isim database üzerinde açılacak olan tablonun ismi değil sadece tablo tanımının ismidir”



**Next** düğmesi yardımıyla açılan sayfada database’de yaratılması istenilen tablonun ismi **“Physical Name Prefix”** alanına yazılır, tablonun firma ve/veya dönem tablosu olup olmadığına ilişkin bilgi **“Name Parameters”** düğmesine basılarak açılan küçük pencereden seçilir.





“Logicalref” kolonu j-guar ürününde tüm tablolarda yer alan **primary key** kolonudur, bu nedenle yaratılan her tablo için otomatik olarak oluşturulur.

Bizde “Code” ve “Description” isimli iki kolon ekleyip, tanımlamayı **finish** düğmesine basarak tablo oluşturma işlemini sonlandırılm.

Table Name: CTMachineCode  
Physical Name: MachineCode  
Database Name: X\_SV(firm)\_MachineCode  
Name Parameters  
Record Info Columns:   
Search:  Name  
Name | Type | Size | Primary K... | Allow Null | Description  
LOGICALREF | integer | 4 | true | false | Logical reference  
CODE | string | 30 | false | false |  
DESCRIPTION | string | 30 | false | false |

**Not:** “Tabloların database’de fiziksel olarak oluşabilmesi için j-guar Administrator’da uyarlama tablolarının sürüme göre güncellenmesi gerekmektedir.”

### 3.2. Nesnelere

Formlar ve tablolar arasındaki iki yönlü veri hareketi nesnelere tarafından sağlanmaktadır. Bu cümleden yola çıkarsak Nesnelere tablolar ile formlar arasındaki katmanı oluşturduğunu rahatlıkla söyleyebiliriz.

j-guar ürününde nesnelere ikiye ayrılmıştır.

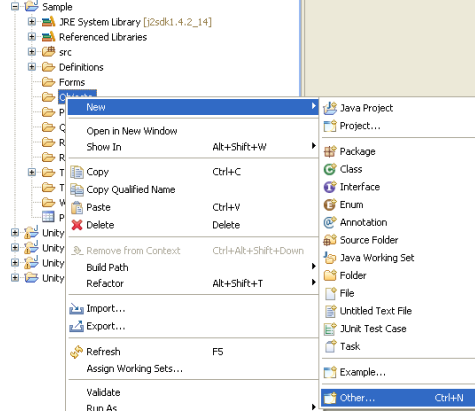
- Business Objectler
- Custom Business Objectler

İki nesne grubu arasındaki fark Business Objectler j-guar ürününe ait ürün nesnelere dir, Custom Business Objectler ise uyarlama yapan kişi tarafından oluşturulmuş yeni nesnelere dir.

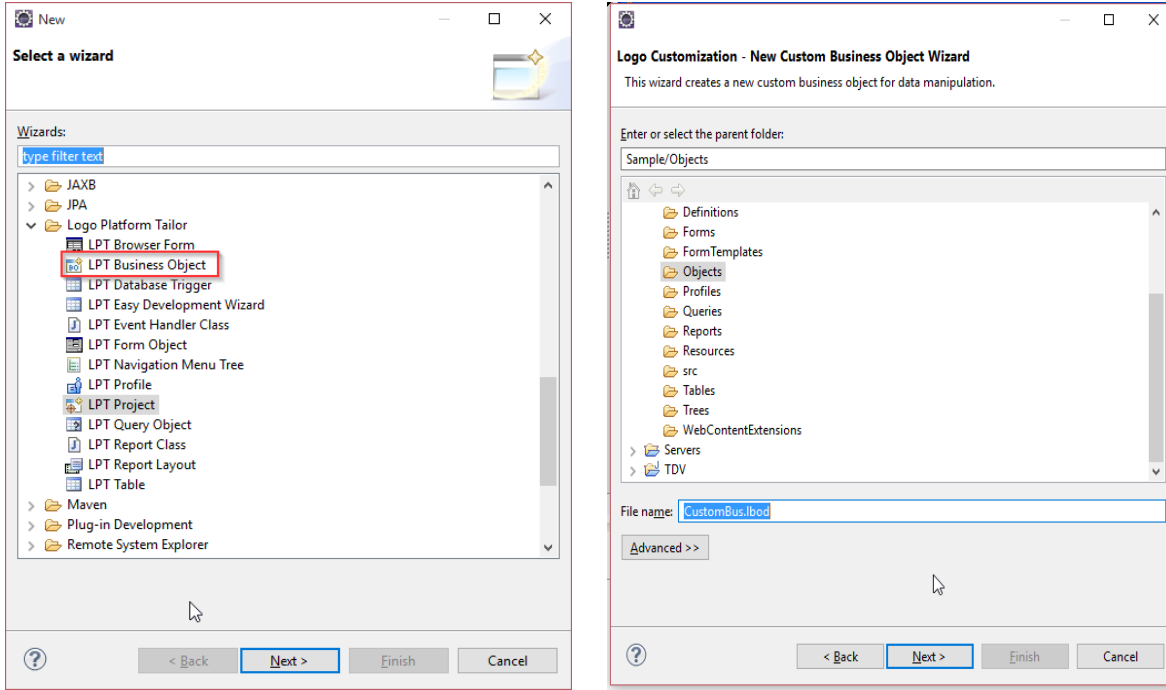
#### 3.2.1. Yeni bir nesne oluşturmak

3.1 bölümünde Makine kodları için bir tablo oluşturmuştuk şimdi bu tabloya ilişkin nesne tanımını yaparak konuyu örneklendirelim.

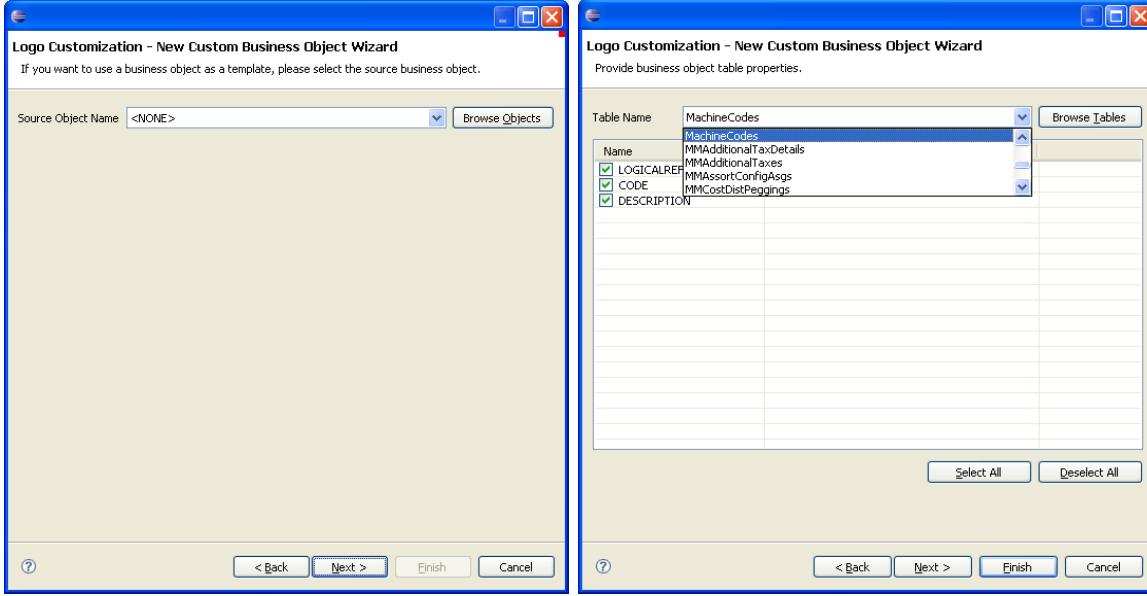
Objects klasörü üzerinde **sağ click** yaparak menu’den “New” ardından da “Other”ı seçerek sihirbazların yer aldığı ekranı açalım.



Uyarlama yapan kişi tarafından yeni oluşturulacak nesnelerin Custom Business object şeklinde olduğunu daha önce belirtmiştik. Bu nedenle seçimimizi **LPT Business Object** olarak yaparak “**Next**” düğmesine tıklayalım. Gelen ekranda bizden yaratmak istediğimiz nesneyi isimlendirmemiz istenecek, isimlendirmeyi yaparak “**Next**” düğmesine tıklayarak devam edelim.



Gelen Sayfada herhangi bir seçim yapmadan “**Next**” düğmesine basarak ilerleyelim. Şimdi bize yaratmak istediğimiz nesnenin hangi tabloyla bağlantılı olduğu sorulmaktadır. Bu aşamada 3.1 nolu madde yaratmış olduğumuz CTMachineCodes tablosunu seçelim, bu seçimin ardından ekrandaki grid’e tabloda bulunan kolonlar getirilecektir. Yaratmış olduğunuz nesneye dâhil olmasını istemediğiniz kolonlardaki check işaretini kaldırabilirsiniz yapmadan “**Next**” düğmesine basarak ilerleyelim

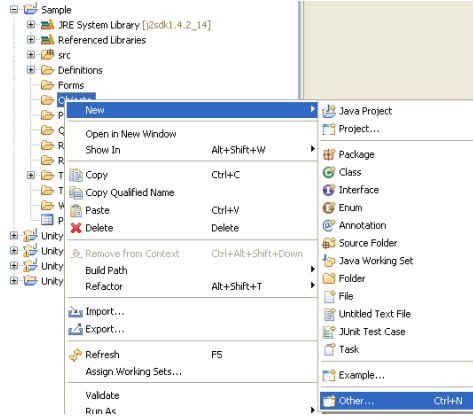


Gelen Sayfada herhangi bir şey yapmadan “Next” düğmesine basarak ilerleyelim. Sihirbazın son sayfasına ulaştık, bu sayfa üzerinde de hiçbir şey yapmadan “finish” düğmesine basarak sihirbazı sonlandıralım.

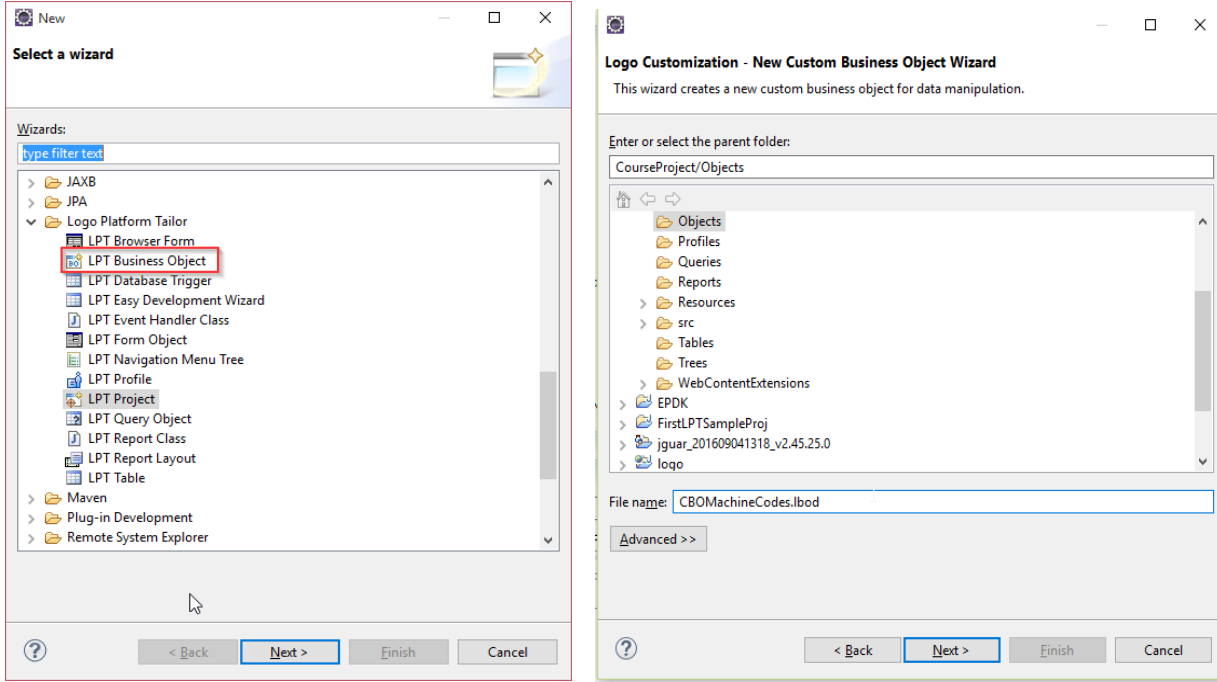
### 3.2.2. Mevcut bir ürün nesnesini genişletmek (Extension)

Bu örneğimizde de cari hesap kartı nesnesine alan ekleyerek nasıl genişletebileceğimizi inceleyelim.

Objects klasörü üzerinde sağ click yaparak menü’den “New” ardından da “Other”’ı seçerek sihirbazların yer aldığı ekranı açalım.



Bir önceki örneğimizde olduğu gibi yine LPT Business Object seçiminde bulunarak “Next” düğmesine basıyoruz. Gelen ekranda nesnemizi isimlendirerek “Next” düğmesine basarak bir sonraki sayfayı açıyoruz.



Gelen ekranda herhangi bir işlem yapmadan “Next” düğmesine basıyoruz. Şimdi nesneye bağlamak istediğimiz tabloya ilişkin seçim yapıyoruz. Bu seçimin ardından tabloya ilişkin kolonlar ekrandaki grid’e gelmektedir. ve “Finish” düğmesine basarak sihirbazı sonlandırıyoruz.

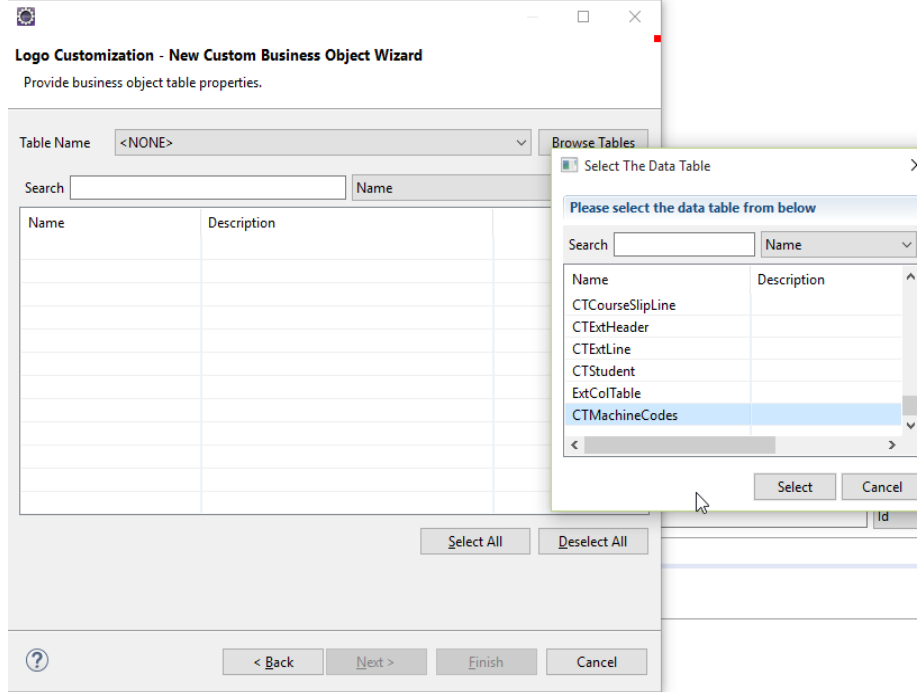


Table Name: CTMachineCodes

Search: [ ] Name: [ ]

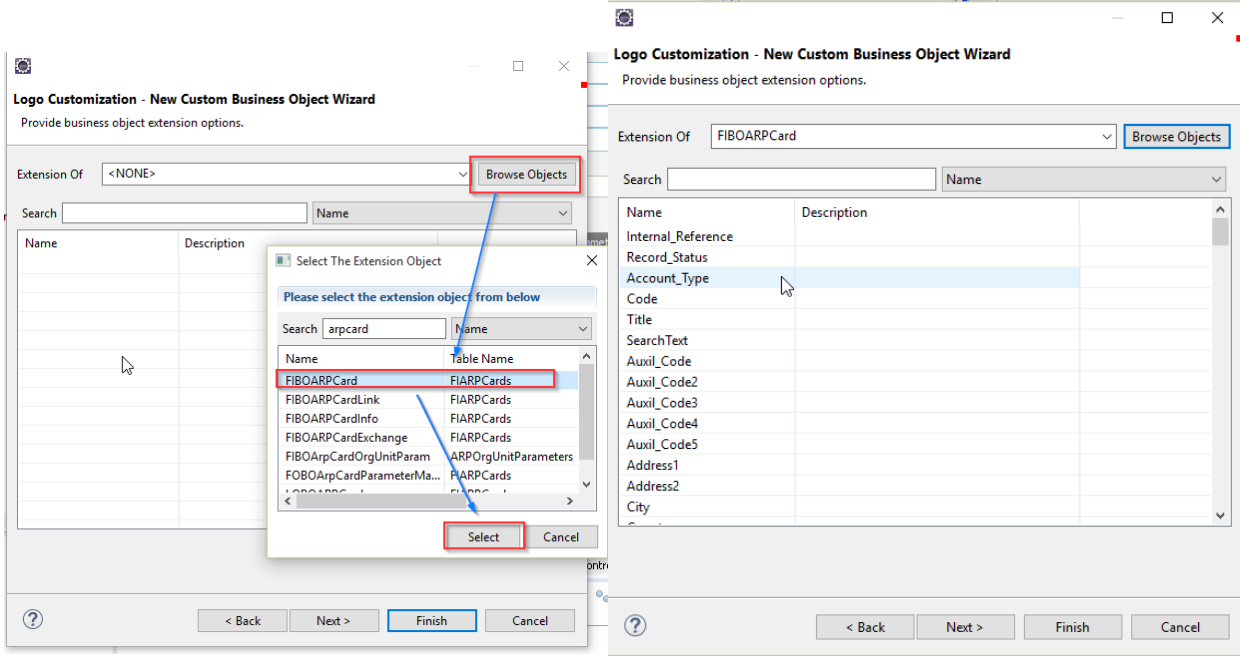
Name	Description
<input checked="" type="checkbox"/> LOGICALREF	Logical reference
<input checked="" type="checkbox"/> CODE	
<input checked="" type="checkbox"/> DESCRIPTION	

Select All Deselect All

< Back Next > Finish Cancel

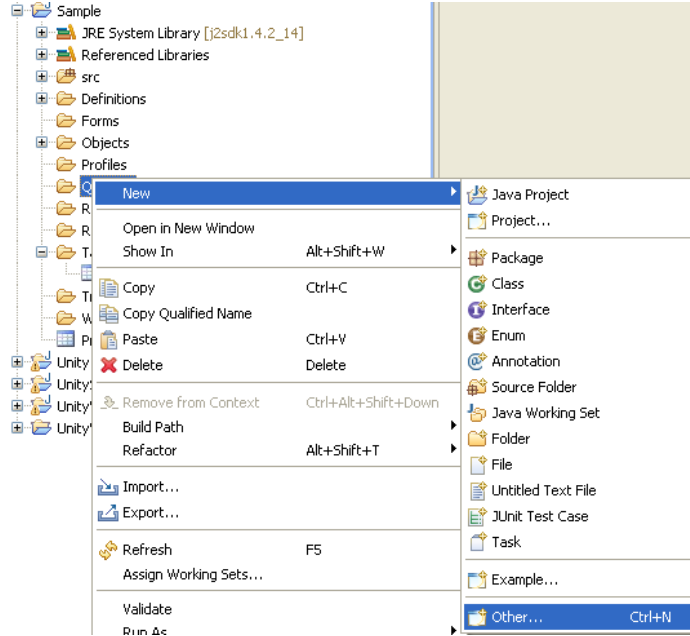
Ürün formlarına nesnelere kullanarak yeni alanlar ekleriz bunun için **Extensions** tanımı yapılması gerekir yine aynı tablo üzerinden ikinci bir nesne ekleyerek bunu cari kartına bağlayabiliriz. Form üzerinde nasıl kullanıldığını ise ürün formlarını uyarlama bölümünden inceleyebilirsiniz.

Yapacağımız işlem adımları farklı isimde nesne tanımı yapılması ve **Extension** yapılacak ürün nesnesi seçilmesi. Örnekte kullanacağımız nesne adı CBOExtMachineCodes

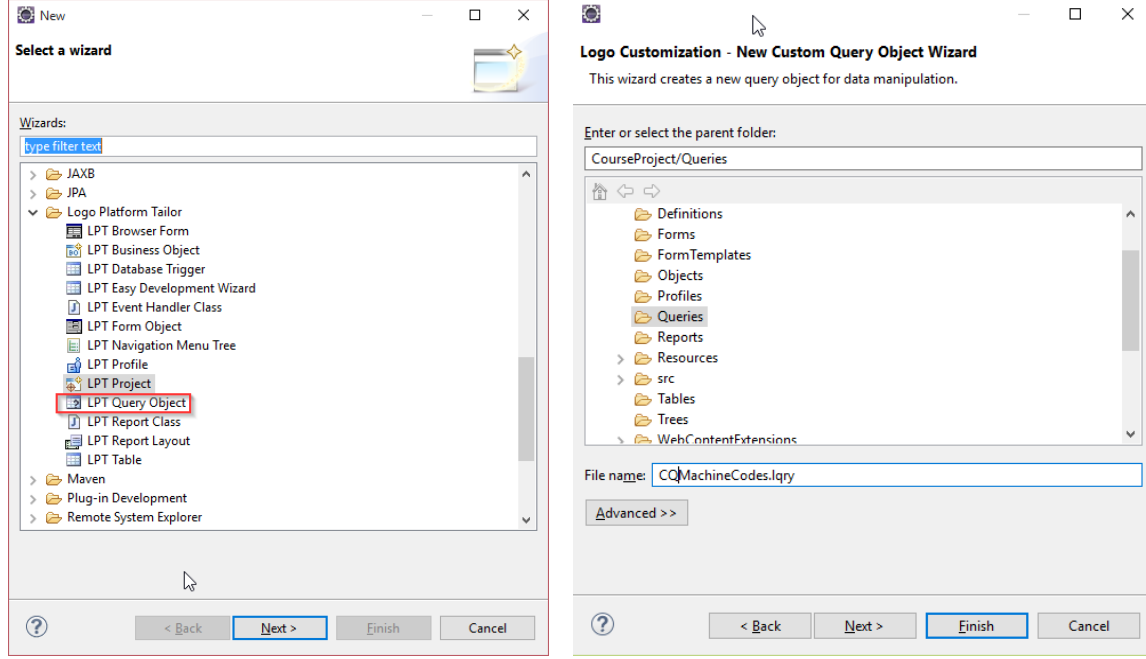


### 3.3. Sorgular

Uyarlama query nesneleri (Query Business Object) “**Queries**” klasörü içerisinde yer almaktadır. Yeni bir query nesnesi oluşturmak için sağ click yaparak menü’den “**New**” ardından da “**Other**”ı seçerek sihirbazların yer aldığı ekranı açalım.

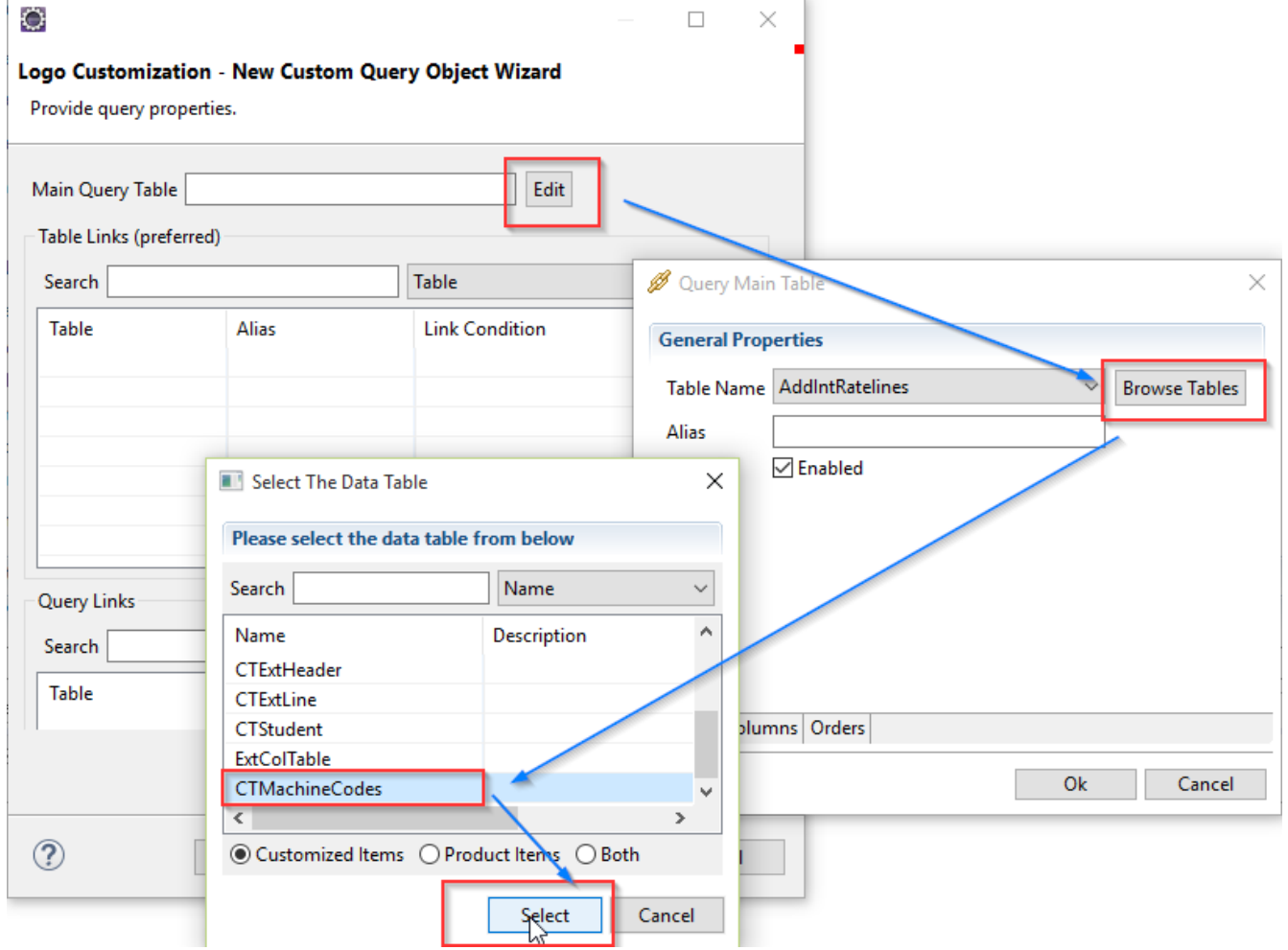


Açılan ekranda LPT Query Object seçeneğini seçerek “**Next**” düğmesine basalım, açılan sayfada bizden query nesnesi isimlendirmemiz beklenmektedir, isimlendirme işlemini yaptıktan sonra “**Next**” düğmesine basalım.



Gelen ekranda mevcut queryler listelenmektedir, bu queryler den birini şablon olarak kullanabildiğimiz gibi yeni bir query de oluşturabiliriz, bunun için “<Empty>” seçeneği seçili olarak “Next” düğmesine basalım.

Gelen ekranda boş bir grid bulunmaktadır, sağ click yaptığımızda gelen menu’de “Add” seçili iken click’leyelim. Tablo seçebileceğimiz bir pop up gelecektir. Combobox’dan “MachineCodes” tablosunu seçelim. Ve “Ok” düğmesine tıklayarak pop up’ı kapatalım.



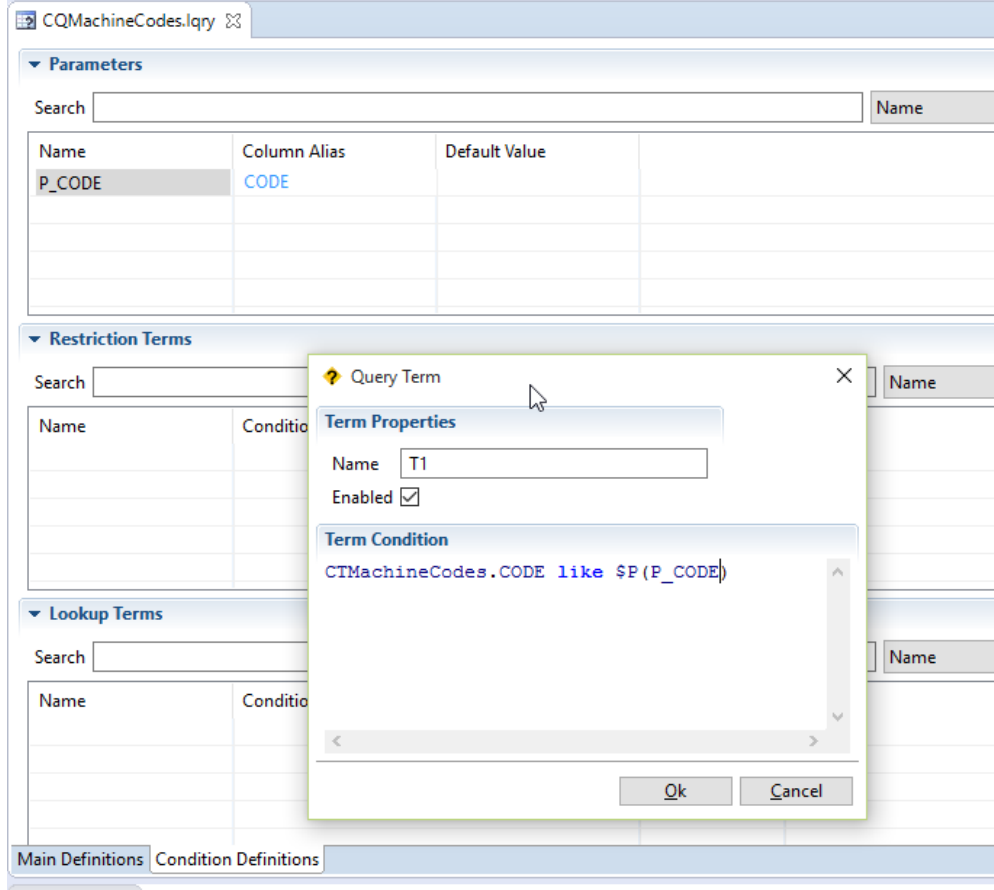
“Finish” düğmesine basarak sihirbazı kapatalım.

Sihirbaz kapatıldıktan sonra ekranda query tanımı görüntülenecektir, bu ekran üzerinde 2 adet tab bulunmaktadır.

- **Main Definition**  
Bu kısımda query’de select edilecek kolonlar, join edilecek table’lar ve order tanımları yapılabilmektedir.
- **Condition Definition**  
Bu kısımda query için koşul oluşturulabilmektedir. En üst kısımda parameter’s tanımı yer almaktadır, burada tanımlan parametre değeri orta kısımda bulunan “**Restriction Terms**” kısmında kullanılabilmektedir.

Örneğimize göre Lookup Tems üzerinden makine kodunu filtreleyebileceğimiz bir koşul hazırlayalım. İlk hareketimiz parametre değerini belirlemektir. “P\_Code” isimli parametre değerini hazırladıktan sonra bu değeri koşulda aşağıdaki gibi kullanalım...

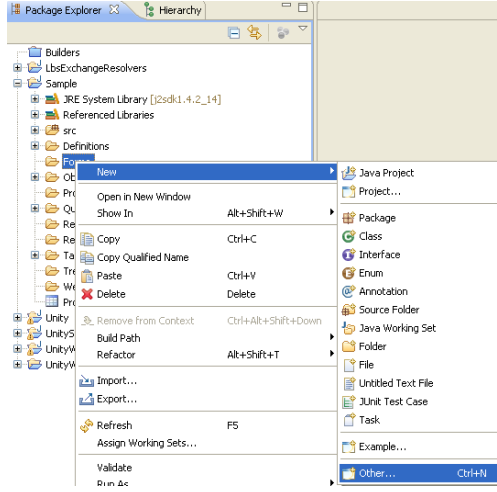




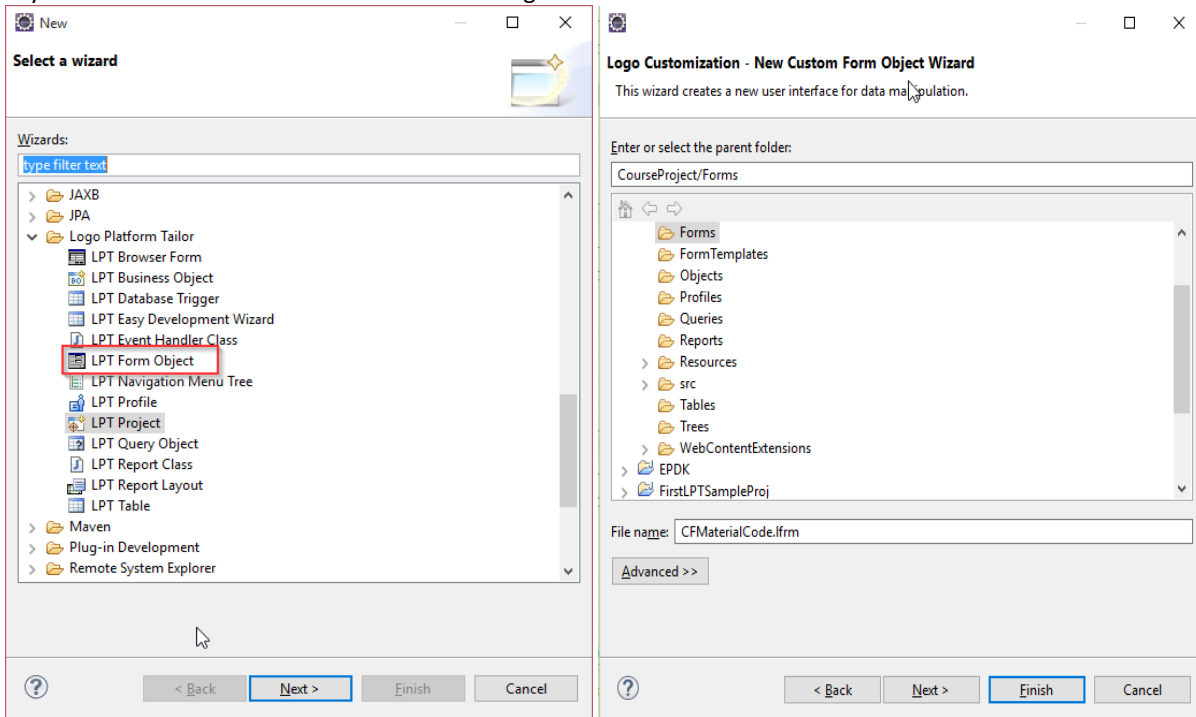
## 3.4. FORMLAR

### 3.4.1. Yeni form oluşturmak

Formlar uyarlama projesi içerisinde yer alan “**Forms**” klasörünün içerisinde bulunmaktadır. j-guar kartların ve browser’ların yer aldığı bir yapıya sahiptir, sırasıyla kart ve browser yapısı anlatılmaya çalışılacaktır. Forms klasörü üzerinde **sağ klik** yaparak menü ’den “**New**” ardından da “**Other**”ı seçerek sihirbazların yer aldığı ekranı açalım.



Sihirbazın ilk sayfasında “LPT Form Object(Table Layout)” seçeneği işaretlenerek “Next” düğmesine basılarak ilerlenir gelen sayfada form nesnesi isimlendirilerek “Next” düğmesine basılır.



Sihirbazın gelen sayfasında, yaratılmak istenilen form için şablon kullanılmak istenip istenmediği seçilir (Form şablonları “LbsDefinitions.jar” dosyasında bulunmaktadır), eğer boş bir ekran açılması isteniyorsa ön değer olarak gelen “<Empty>” seçeneği değiştirilmemelidir. Bu ekran üzerinde yer alan seçeneklerden bahsetmek gerekirse,

- **Create event handler class:** Bu seçenek form için kod yazılmasına olanak sağlayacak olan class dosyasının otomatik olarak yaratılmasına olanak sağlamaktadır. Ön değer seçili olarak gelmektedir, yani class dosyası yaratılır.
- **Clear inherited events & properties:** Eğer şablon kullanarak form yaratıyorsanız, yani ürünün formlarından birini uyarlıyorsanız, ilgili formun arkasında ki ürün kodları otomatikman yaratmış olduğunu bu yeni formda da çalışıyor olacaktır, eğer bu bahsedilen ürün kodlarının çalışması istenilmiyor ise bu seçenek işaretlenir.

“Next “ düğmesine basılarak ilerlenir.

**Form Template**

The available form templates are listed below.

Search  Name

Name	Module	Description
<EMPTY>		Empty form
CMXFCommission.jfm	Satış Yönetimi	Komisyon Tanımı
CMXFCommissionBrowser.jfm	Satış Yönetimi	Komisyon Tanımları
CMXFCommissionLineBrowser.jfm	Satış Yönetimi	Satış Temsilcisi Komisyonl...
CMXFCommissionStateChange.jfm	Satış Yönetimi	Komisyon Durum Değişikl...
CMXFCommissionUpdate.jfm	Satış Yönetimi	Satış Komisyonları Güncel...
CMXFCustomer.jfm	Satış Yönetimi	Müşteri Kartı
CMXFCustomerBrowser.jfm	Satış Yönetimi	Müşteriler
CMXFCustomerContact.ifm	Satış Yönetimi	Müşteri Bağlantıları Kartı

Create event handler class  
 Clear inherited events & properties

Layout

Form Width  Column Count   
Form Height  Row Count

Append a "FILL" column to the end (for resize issues)  
 Append a "FILL" row to the end (for resize issues)

Tanımlayacağımız formun Custom Business Objecti var ise onu seçeriz yoksa CBO dan bağımsız bir form tasarıncaksa Finish butonuyla işlem tamamlanır.

CBO bağlamak için **Browse Business Object** butonuna tıklanır, açılan ekrandan daha önceden oluşturduğumuz CBOMachineCodes seçilir secme işleme sonrası CBO içerisinde bütün alanlar tanımlama sihirbazında gelir, bu alanlardan ihtiyaçlarımız olanları seçeriz.

The Available Business Objects and Fields Listed Below.

Editor Title:

Business Object:

Name	Description	Type	Column	Components	Has L
<input type="checkbox"/> LogicalReference	Logical reference	Integer	LOGICALREF		false
<input checked="" type="checkbox"/> Code		String	CODE		false
		String	DESCRIPTION		false

Select Business Object

Please select a Business Object from the list

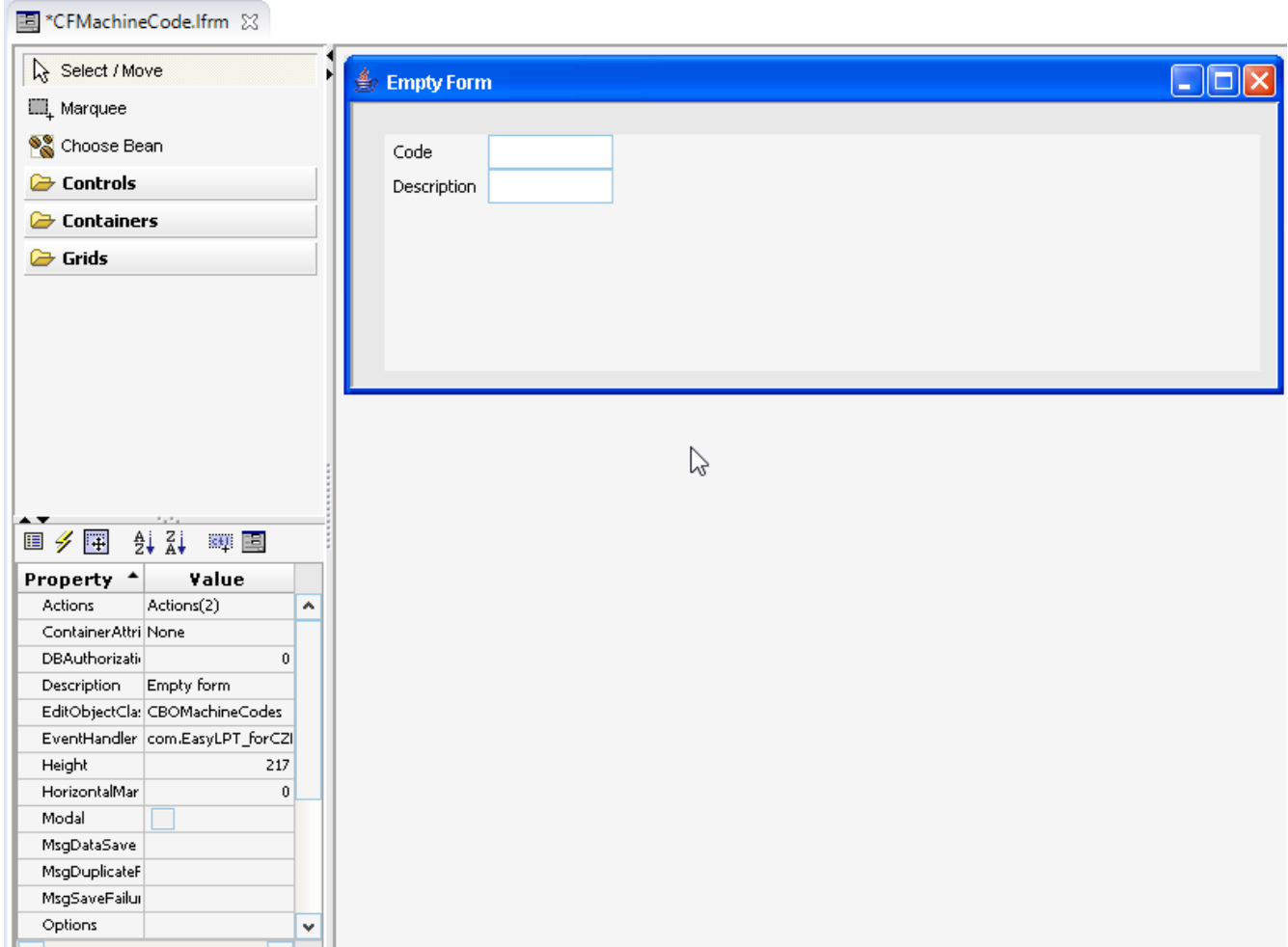
Search:  Name:

Name	Description
CBOCourseSlip	
CBOCourseSlipLine	
CBOExtHeader	
CBOExtLine	
CBOStudent	
CBOMachineCodes	

Select All Deselect All

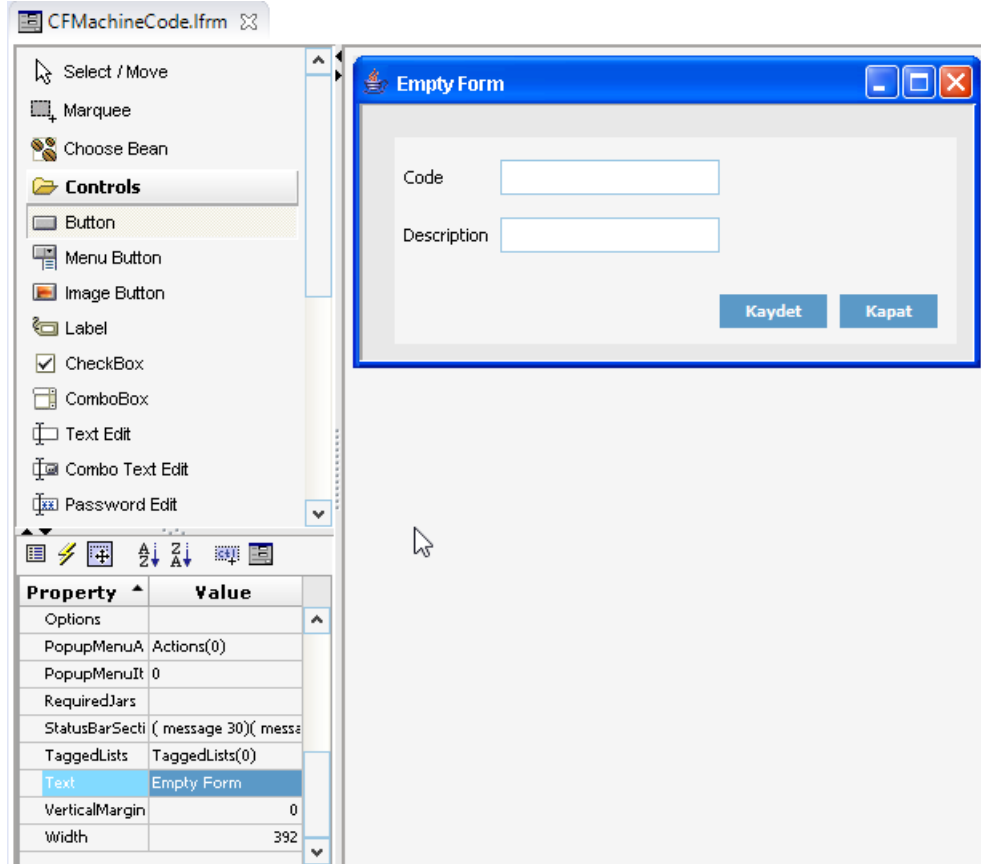


Tanımlarken belirlediğimiz alanların form tasarımında vertical olarak yerleştirildiğini görürüz. Formun sol tarafında kullanılabilecek kontrollere ilişkin bir liste bulunmaktadır. Bu liste içerisinde yer alan kontroller sürükleyip bırakarak form'a yerleştirilebilmektedir.

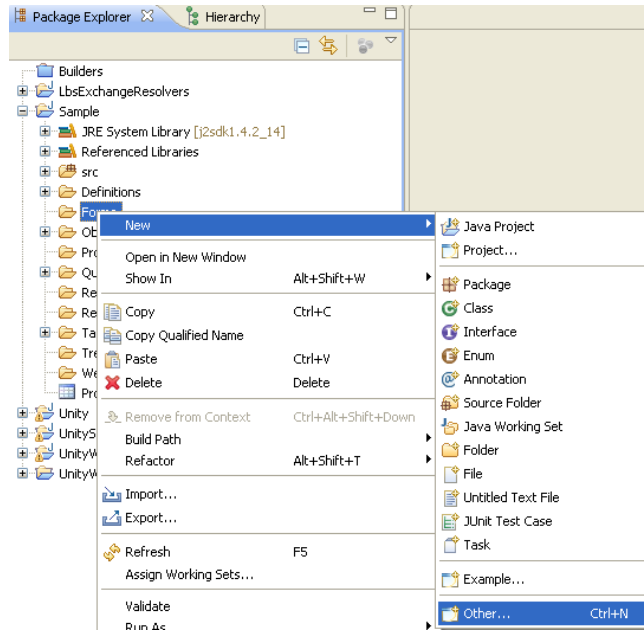


Makine kodları örneğimize devam ederek bu örnek için gerekli kart ve browser'ı yaratalım. Önce form'a 3.2.1 nolu madde de oluşturduğumuz nesnenin bağlandığını görebiliriz.

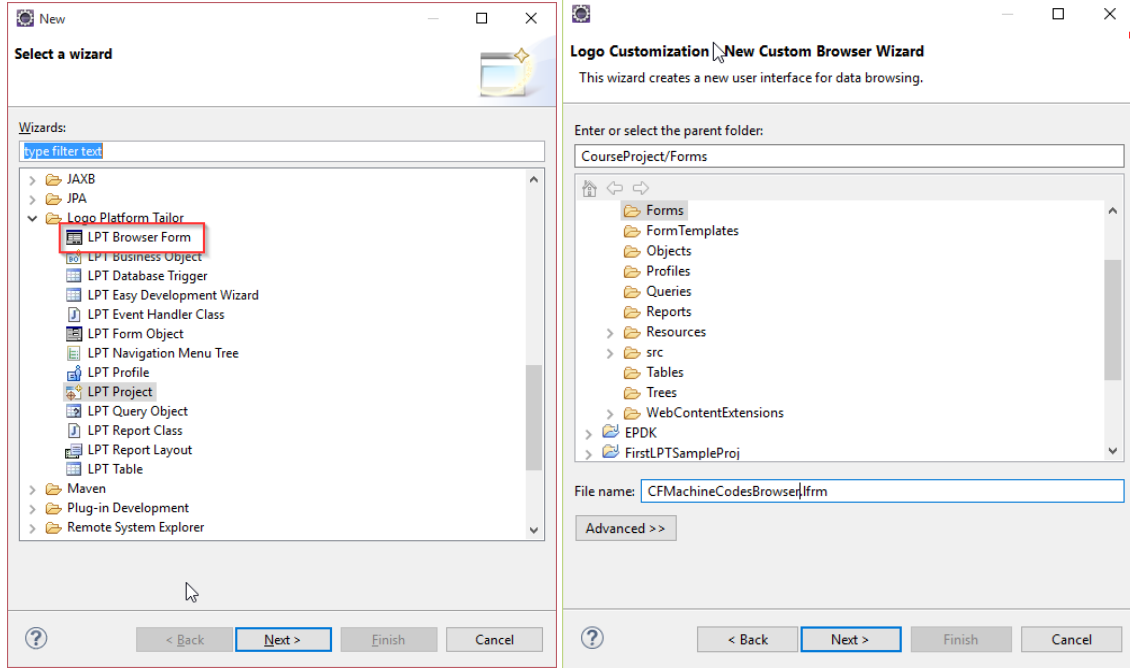
İki adet düğme koyalım ve düğmeleri "kaydet" ve "kapat" şeklinde isimlendirelim. "Kaydet" düğmesi için özellikler tabında "Action" özelliği için "Save & Close" "Kapat" düğmesi için özellikler tabında "Action" özelliği için "Close" özelliğini seçelim. Kart'ımız böylelikle tamamlanmış olacaktır.



Şimdi Browser'ımızı oluşturmaya başlayalım, Forms klasörü üzerinde sağ click yaparak menu'den "New" ardından da "Other"ı seçerek sihirbazların yer aldığı ekranı açalım.



Gelen sihirbaz ekranı üzerinde “LPT Browser Form” özelliğini seçerek “Next” düğmesine basalım. Gelen ekranda browser’ın isimlendirmesini yaparak “Next” düğmesine basarak ilerleyelim.



Gelen Ekrandan;

- **Browser Title** alanına Başlık girilir.
- **Query Name** alanına daha önceden tanımladığımız query’i seçeriz
- Query Seçme işlemi sonrası tablomuzdaki tanımlı alanlar listelenir **select all** butonuna tıklarız
- **Select DBEntryDataClass** alanına daha önce tanımladığımız BO muzı tanımlarız
- **Select Object Class** alanına daha önce tanımladığımız BO muzı tanımlarız.
- **Select DBEntryFormClass** alanına Browser üzerinden açmak istediğimiz formun seçerek next butonuna tıklarız.
- Gelen ekran üzerinden Browserda görünmesini istediğimiz alanları seçeriz ve Finish butonuyla bitiririz.

**not:** hepsinin seçilip browser design-time üzerinden istenilen alanların kaldırılması daha sağlıklı olduğundan hepsini seçip **Finish** butonuna tıklarız.



### Logo Customization - New Custom Browser Wizard

Provide browser query options.

Browser Title:

Query Name:

Search:  Alias:

Alias	Column Name	Description
<input checked="" type="checkbox"/> LOGICALREF	LOGICALREF	
<input checked="" type="checkbox"/> CODE	CODE	
<input checked="" type="checkbox"/> DESCRIPTION	DESCRIPTION	

Identify Filter options

Create Event Handler Class

Select DBEntryDataClass:

Select ObjectClass:

Select DBEntryFormClass:

### Logo Customization - New Custom Browser Wizard

Provide browser filter and search options.

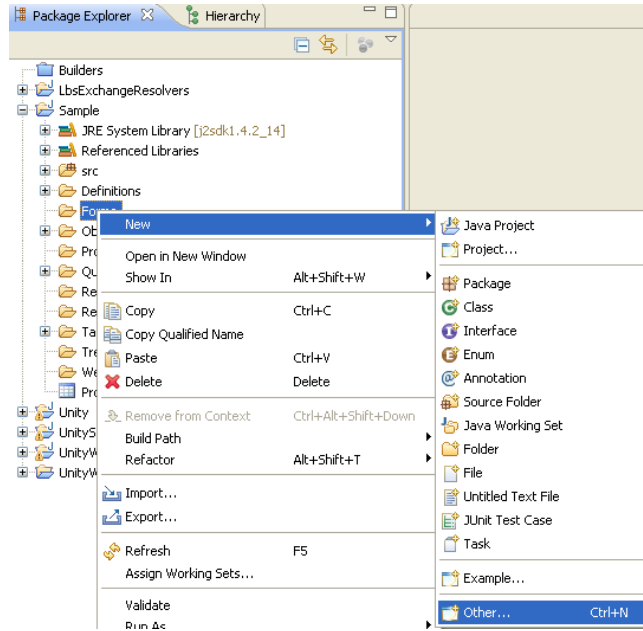
Query Name:

Search:  Filter Column:

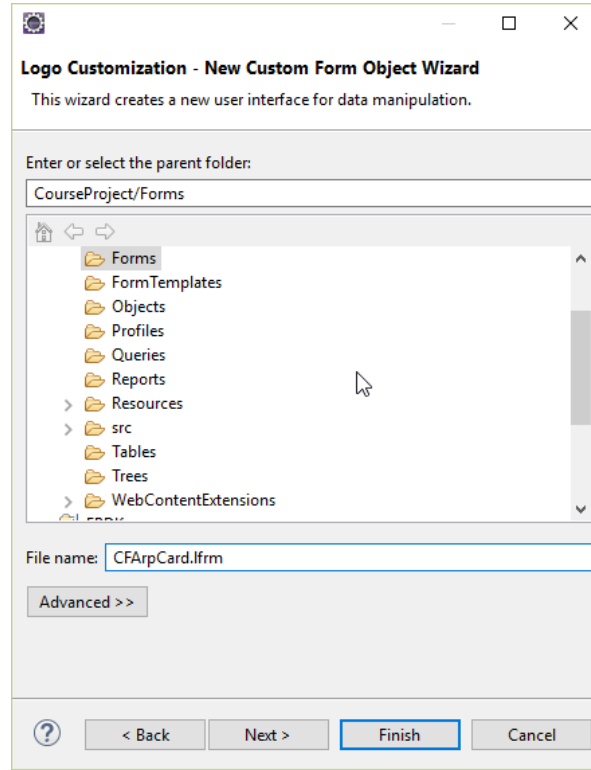
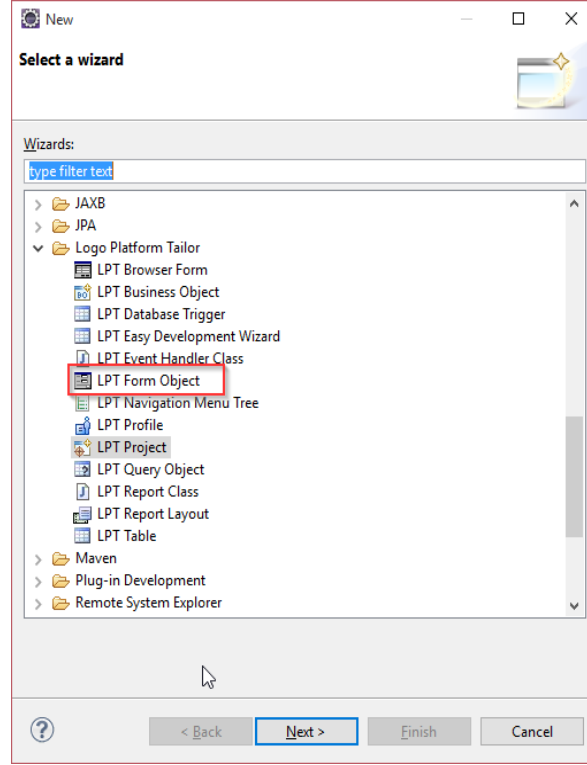
Filter Column	Type	Generate Type
<input checked="" type="checkbox"/> LOGICALREF	Normal	Data-Bound
<input checked="" type="checkbox"/> CODE	Normal	Data-Bound
<input checked="" type="checkbox"/> DESCRIPTION	Normal	Data-Bound

### 3.4.2. Mevcut ürün formunu uyarlamak

Konuyu örneklendirerek devam edelim, forms klasörü üzerinde sağ click yaparak menü 'den "New" ardından da "Other"'ı seçerek sihirbazların yer aldığı ekranı açalım.



Açılan sihirbaz ekranında “LPT Form Object” seçili iken “Next” butonuna basarak ilerleyelim, gelen sayfada formumuzu isimlendirerek yine “Next” düğmesine basıyoruz.



Açılan sayfada uyarlamak istediğimiz ürün formunu seçiyoruz, örneğimiz için bu cari hesap kartı (FIXFARPCard.jfm) "Finish" düğmesine basarak sihirbazı sonlandırıyoruz.

**Form Template**  
The available form templates are listed below.

Search: arpcard Name: [v]

Name	Module	Descrip
FIXFARPCard.jfm	Finans	Cari He
FIXFARPCardBrowser.jfm	Finans	Cari He
FIXFARPCardCampaignPoints.jfm	Finans	Temina
FIXFARPCardEntryForForms.jfm	Finans	Form B
FIXFarpCardOrgUnitParams.jfm	Finans	Cari He
FIXFARPCardStateChange.jfm	Finans	Cari He
FIXFARPCardStateChange.jfm	Finans	Cari He

Create event handler class  
 Clear inherited events & properties

Layout

Form Width	400	Column Count	4
Form Height	300	Row Count	3

Append a "FILL" column to the end (for resize issues)  
 Append a "FILL" row to the end (for resize issues)

< Back Next > Finish Cancel

Sihirbazın sonlandırılmasıyla ürüne ait cari hesap kartını görüntülüyoruz.

**Cari Hesap**

Kod  
Unvan  
Arama Metni  
Üst Seviye Birlik  
Organizasyonel Birim

Özel Kod  
Yetki Kodu  
Ticari İşlem Grubu  
Muh. Bağlantı Grubu

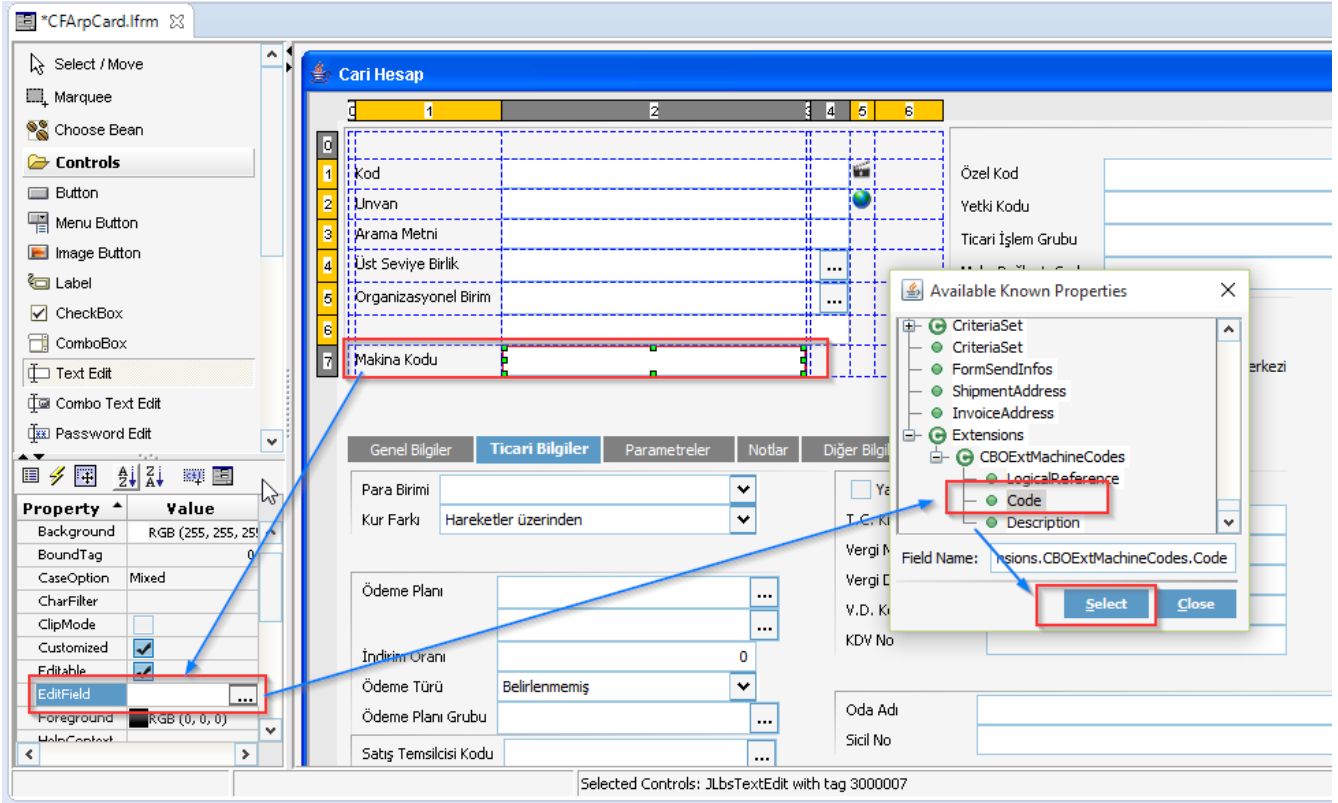
Onay Durumu: Onaysız  
 Potansiyel  Yüklenici  Merkezi

**Genel Bilgiler** | Ticari Bilgiler | Parametreler | Notlar | Diğer Bilgiler | Sevkiyat Günleri

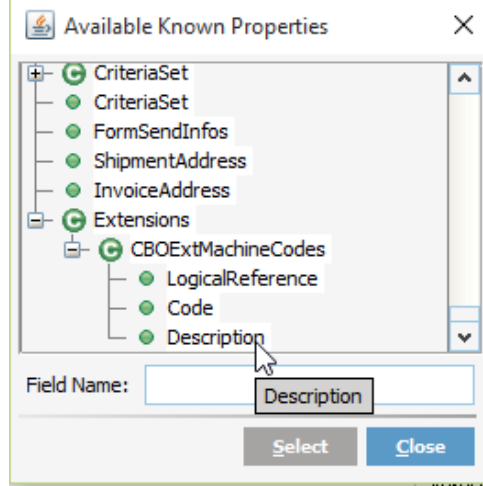
Adres  
Semt  
İlçe  
İl  
Eyalet/Bölge  
Ülke  
Posta Kodu  
Telefon No (1)  
Telefon No (2)  
Telefon No (2)  
Faks No

e-Posta Adresi  
İnternet Adresi  
Yazışma Dili  
Abone Durumu  
Abone Ek Bilgi  
Otom. Ödeme Bankası  
Mağaza Kartı No  
Teslim Şekli  
Taahhütçü Firma  
Kurumumuzun Kodu  
Fiyat Grubu Kodu

- Cari hesap kartında makine kodu için label ve textbox'ları ekleriz.
- Nesnemizdeki alaları textbox'ımıza bağlarız.
- Textbox'ımızı seçtikten sonra özellikler kısmındaki "EditField" alanına tıklayıp açılan popup ekranında nesnemize ait elemanların geldiğini görürüz.

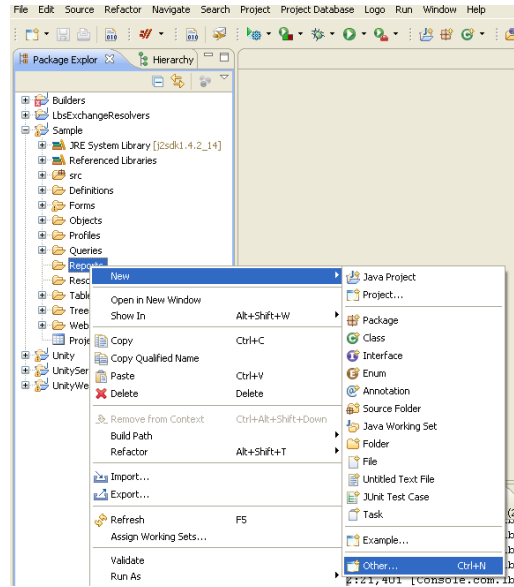


- Nesnenin en alt bölümünde "Extensions" yani uyarlama ile dâhil edilmiş elemanlar gelmektedir. İşte sırasıyla bu elemanları formda uyarlama ile eklediğimiz alanlara bağlayarak sonlandırıyoruz.



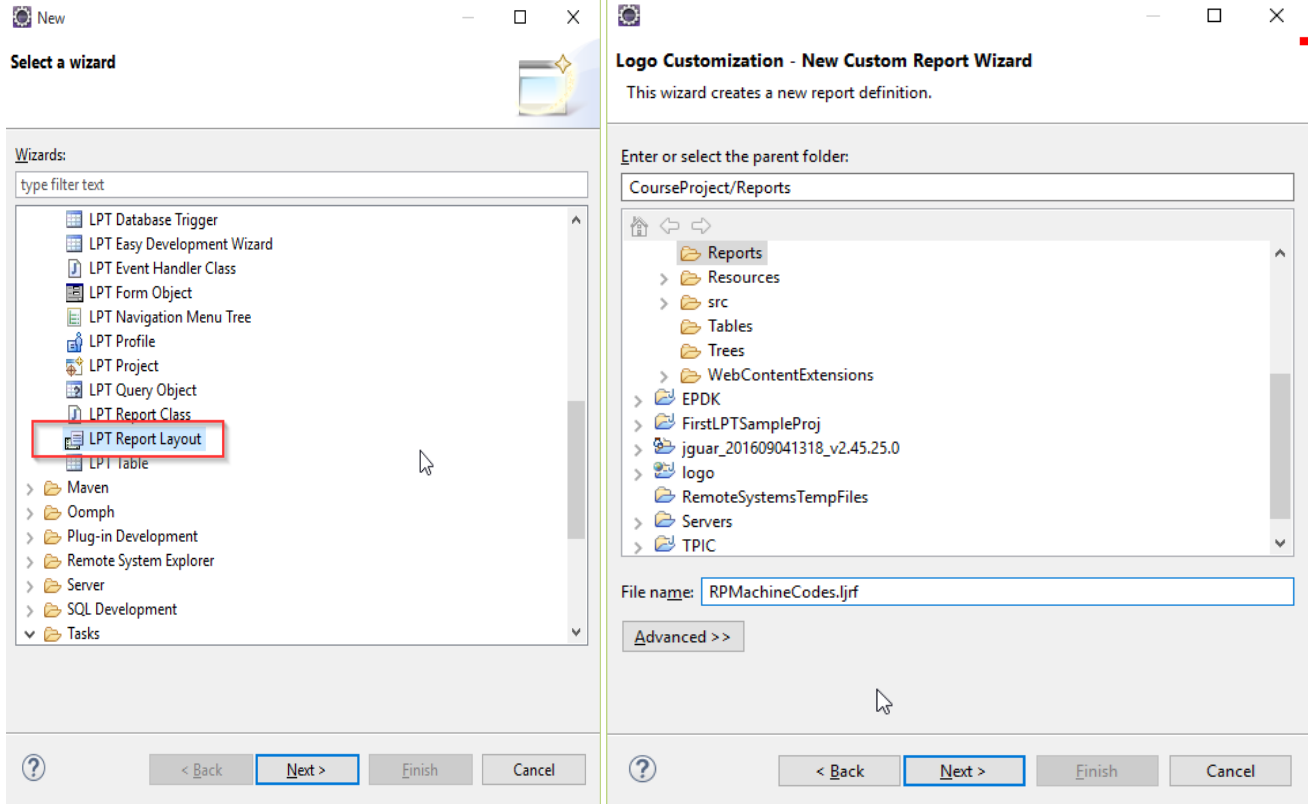
### 3.5. Raporlar

Reports klasörü üzerinde sağ click yaparak menüden “New” ardından da “Other”ı seçerek sihirbazların yer aldığı ekranı açalım.



Makine kodlarına yönelik hazırladığımız örnek üzerinden giderek ilgili tabloya ilişkin kayıtların raporunun nasıl alınabileceğini inceleyelim.

- Sihirbaz üzerinde “LPT Report Layout” seçiliyken “Next” düğmesine basarak ilerleyelim.
- Gelen sayfada raporumuzu isimlendirelim ve “Next” düğmesine basalım.



- Rapor için şablon seçilir, eğer şablon kullanılmayacaksa “<EMPTY>” seçeneği işaretli olarak “Next” düğmesine basılarak ilerlenir.
- Rapor tanımında kullanılmak üzere rapor adı ve rapor açıklamasının isimlendirilmesi beklenmektedir.
- Gelen kayıtların düz bir şekilde basılması için “Regular Query Report” seçeneği seçilir. Bu kısımda yapılan seçim class dosyasında kodlanmış şekilde yer alacaktır. “Next” düğmesiyle ilerlenir.

### Report Template

The available report templates are listed below.

Search  Name v

Name	Description	File
<EMPTY>	Empty Report	
AMFRMTransferSlip	Org. Birimler Arası Transfe...	AMFRMTransferSlip.j
AMFRMAITransferSlip	Org. Birimler Arası Transfe...	AMFRMTransferSlip.j
CARPResourcesList	Maliyet Kaynakları Listesi	CARPResourcesList.ji
CARPActivitiesList	Maliyet Aktiviteleri Listesi	CARPActivitiesList.jif
CARPDistributionDefinitionsList	Maliyet Dağıtım Tanımlar...	CARPDistributionDef
CARPDistributionTemplatesList	Maliyet Dağıtım Şablonlar...	CARPDistributionTer
CARPAssignedCostsList	Yüklenen Maliyetler Dökü...	CARPAssignedCostsl
CARPDistributedCostsList	Dağıtılan Maliyetler Dökü...	CARPDistributedCos
CARPActivityTypesList	Maliyet Aktivite Türleri Lis...	CARPActivityTypesLi
CARPResourceTypesList	Maliyet Kaynak Türleri Lis...	CARPResourceTypes
DERPDemandList	Talep Listesi	DERPDemandList.jif
DERPDemandMeetingAnalysis	Talep Karşılama Raporu	DERPDemandMeetin
LORPPurchaseInvoiceList	Satınalma Faturaları Dökü...	LORPPurchaseSaleslr
LORPPurchaseTable	Alımlar Tablosu	LORPPurchaseSaleTa

Create Default Report Definition

< Back Next > Finish Cancel

### Report Template & Handler

Please select necessary options to build up your report layout and handler.

Report Definition:

Report Name:

Report Description:

Layout Properties:

Generate Header Section

Event Handler:

Query Independent Report

Regular Query Report

Master-Detail Query Report

< Back Next > Finish Cancel

- Gelen ekrandan raporun class dosyasının isimlendirilmesi istenilir.
- “Next” düğmesiyle rapor sihirbazının son sayfasına ulaşılır bu sayfa üzerinde rapor ’un sorgusu seçilir. **3.3** nolu maddede hazırlamış olduğumuz sorguyu rapor sorgusu için seçiyoruz ve “Finish” düğmesine basarak sihirbazı sonlandırıyoruz.

### Java Class

Create a new Java class.

Source folder:  Browse...

Package:  Browse...

Name:

Superclass:  Browse...

Interfaces: Add...

Remove

< Back Next > Finish Cancel

### Query Template

The available query objects are listed below.

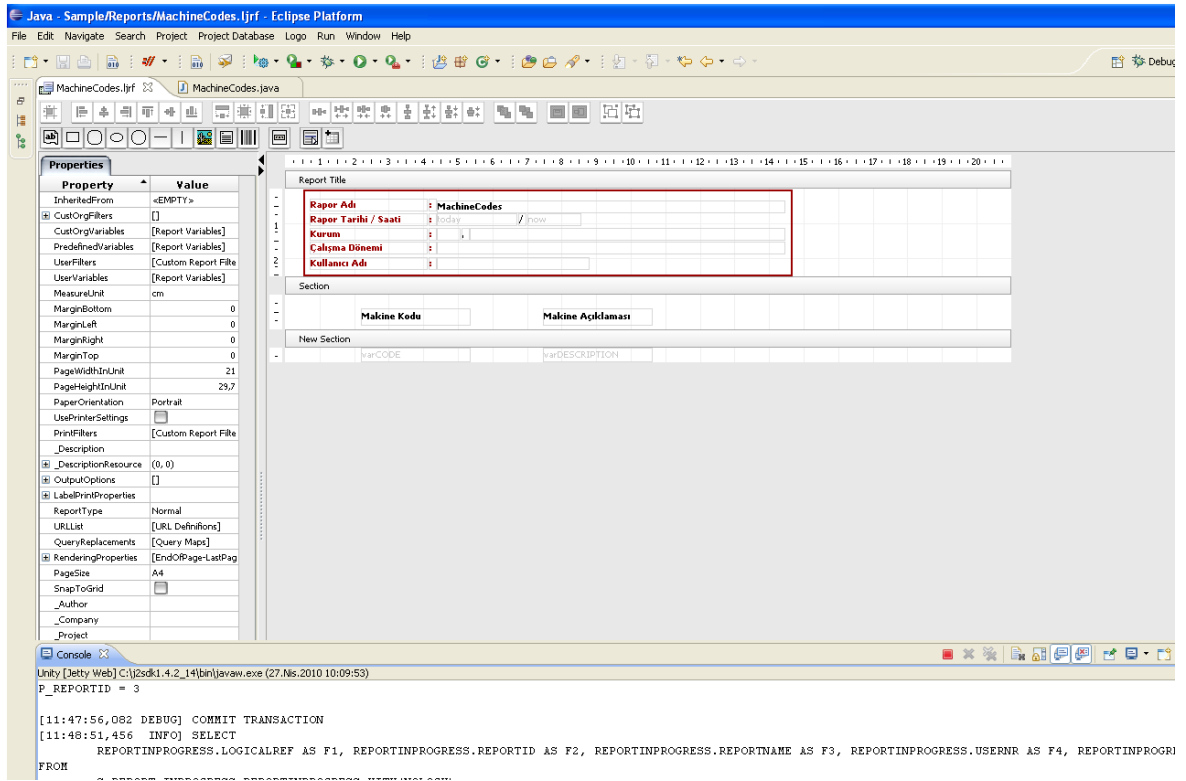
Query Object

Search  Name v

Name	Main Table	Description
CQClassroomBrowser	CTClassroom	
CQCourseBrowser	CTCourse	
CQCourseSlipBrowser	CTCourseSlip	
CQCourseSlipLines	CTCourseSlipLine	
CQInvoiceBrowser	LOInvoiceHeaders	
CQRBudgDisconnArpTrans	FITransactions	
CQRBudgDisconnBankTrans	FIBankTransactions	
CQStudentBrowser	CTStudent	
CQMachineCodes	CTMachineCodes	
AccTotalsLocalSub	GLTransTotsGeneralIV	
AccTotalsRCSub	GLTransTotsGeneralIV	
AccTotalsTCSub	GLTransTotsGeneralIV	
ADQOBusinessSectors	GOBusinessSectors	
ADQOSearchPersonInPosition	HRPersons	

< Back Next > Finish Cancel

- Sihirbaz sonlandırıldıktan sonra rapor tasarım ekranı açılmaktadır. Ekran boş olarak gelmektedir. İlk bölüm(section) başlık bölümü olsun. Bu bölüme makine kodu ve makine açıklaması için iki adet rectangle koyalım. Bölüm tipini (section Type) header olarak ayarlayalım.
- İkinci bir bölüm(section) ve bu bölüme iki adet datanın geleceği alan(field) ekleyelim.
- Rapor tasarımı üzerinde sağ klik yapıp **“import query fields”** seçeneğini seçelim. Böylece sihirbaz 'da seçmiş olduğumuz sorgunun alanları **“user variables”** alanına gelecektir. Bu variable'ları alan(fields) özelliklerinde bulunan **“VariableName”** için seçelim.



- Rapora ilişkin class dosyasına bakıldığında otomatik olarak bir kod bloğu oluşturulmuştur. Bu aşamada datanın geleceği bölüm(section) ID' si(örneğimize göre 2)kod da gerekli yerde düzeltilerek yazılırsa raporumuz hazır duruma gelmiş olacaktır.



```
RPMachineCodes.java
1 package com.EasyLPT_forCZRT.customization;
2
3 import com.lbs.reporting.JLbsReportLayout;
4
5
6
7 public class RPMachineCodes extends JLbsQueryReportBase {
8
9     public RPMachineCodes(JLbsReportLayout layout, IReportRenderer renderer) {
10         super(layout, renderer);
11         // TODO Auto-generated constructor stub
12         setMainQueryName("CQMACHINECODES");
13
14         m_BodySection = 1;
15     }
16 }
17 }
18 }
```

Tanımlar(Definitions) klasöründe bulunan Reports.lrdf içerisinde raporun layout ve class bilgileri bulunmaktadır. Burada bulunan bilgiler sihirbaz tarafından otomatik olarak oluşturulmaktadır.

The screenshot shows the Logo Platform Tailor interface. On the left, the Project Explorer displays the 'Definitions' folder containing various report-related files, including 'reports.lrdf'. The main window shows the 'Report Definitions' table with the following data:

Name	Class Name	Layout File	Description
StudentQueryIndepen...	com.LPT.customizati...	Reports/CRStudentQu...	Student QueryIndependent Exam...
StudentMasterQuery E...	com.LPT.customizati...	Reports/CRStudentM...	StudentMasterQuery Example
StudentMasterDetailQ...	com.LPT.customizati...	Reports/CRStudentM...	Student MasterDetailQuery Exam...
RPMachineCodes	com.EasyLPT_forCZR...	Reports/RPMachineC...	RPMachineCodes

A 'Report Definition Properties' dialog box is open, showing the following details for the 'RPMachineCodes' report:

- Report Name: RPMachineCodes
- Layout File: Reports/RPMachineCodes.ljrf
- Report Class: com.EasyLPT\_forCZRT.customization.RPMachineCodes
- Description: RPMachineCodes

## 3.6. AĞAÇLAR

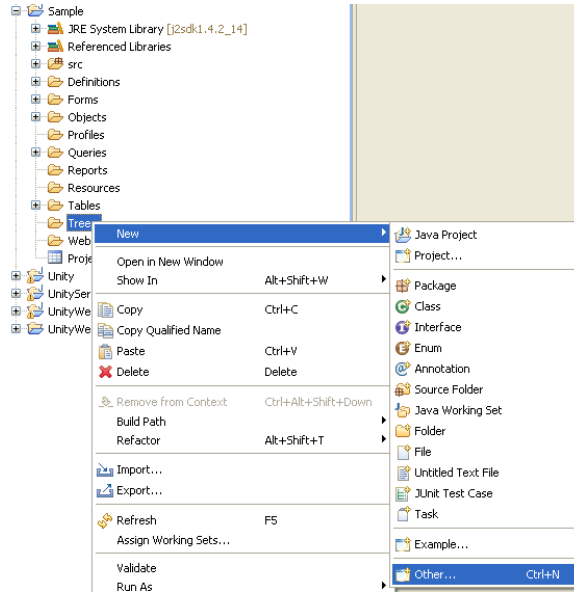
j-guar ürününde menü bir ağaç yapısının şeklindedir ve modüllere ayrılmış bulunmaktadır. Bu modüllerin altında genel yapı itibariyle

- Tanımlar
- Hareketler
- İşlemler
- Rapor

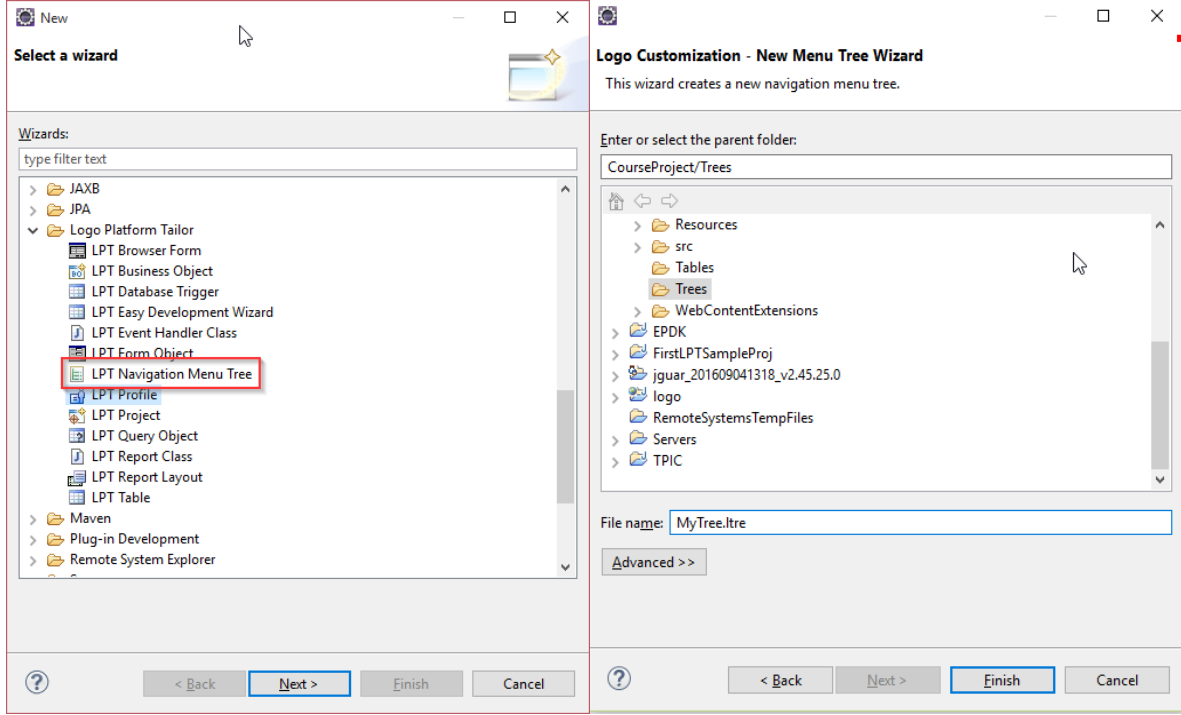
Şeklinde alt modüller yer almaktadır.

Bu ağaca uyarılama yapan kişi tarafından yeni modüller ve alt modüller eklenebilir.

**Trees** klasörü üzerinde sağ click yaparak menüden **“New”** ardından da **“Other”**ı seçerek sihirbazların yer aldığı ekranı açalım.

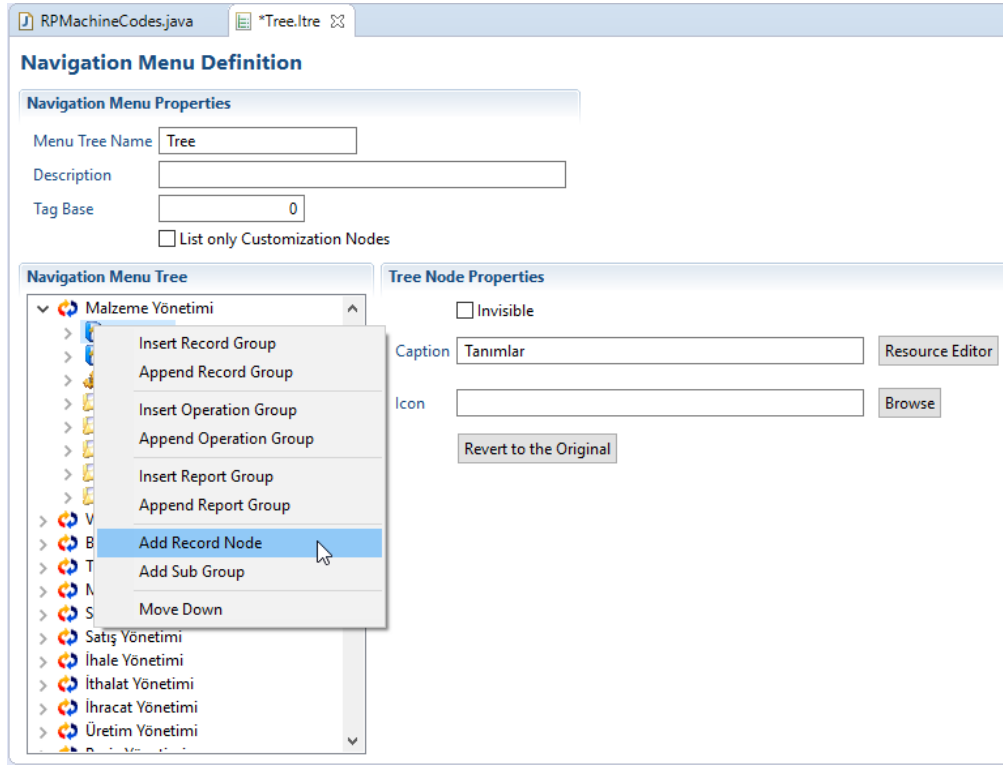


Sihirbazın ilk sayfasında **“LPT Navigation Menu Tree”** seçeneğin işaretleyerek **“Next”** düğmesine basarak ilerleyelim. Gelen sayfada ağacımızı isimlendirerek **“Finish”** düğmesine basarak sihirbazı sonlandıralım.



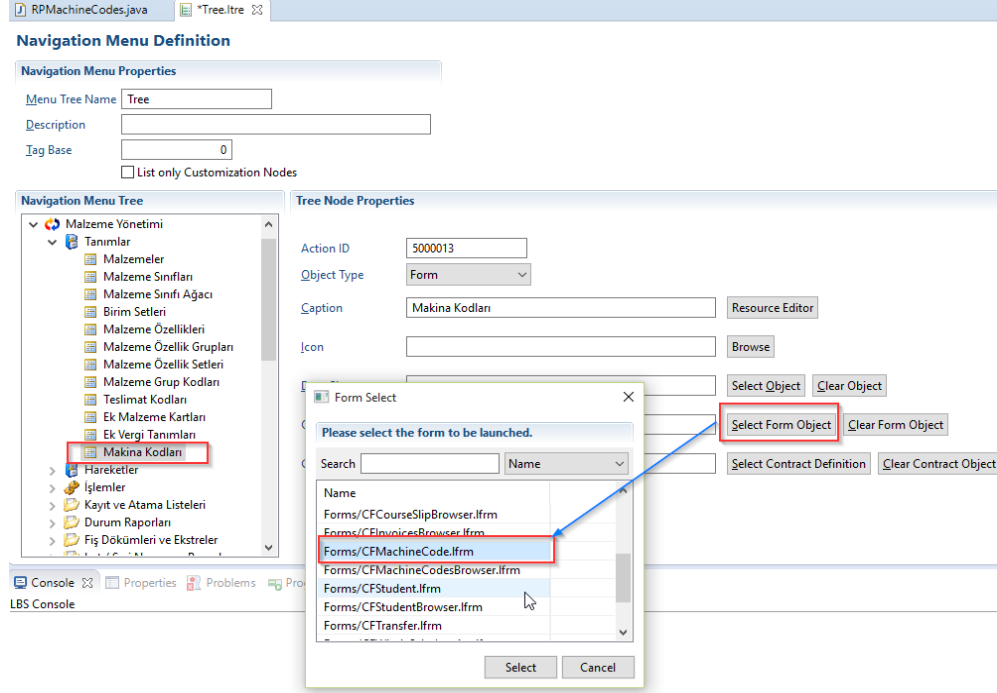
Ağaç yapısı ekrana gelecektir.

Örneğimize devam edelim makine kodları browser ekranını Malzeme Yönetimi altında bulunan tanımlar alt grubuna eleman olarak takalım bunun için ağaç üzerinde Malzeme Yönetimine tıklayarak altındaki grupları görünür hale getirelim ardında Tanımlar üzerinde sağ click yaparak gelen menüde **“Add Record Node”** seçeneğini seçelim.



Açılan tanım ekranında **“Object Type”** özelliğini form olarak belirleyelim.

Başlık için “Caption” özelliğine Makine Kodları Yazalım.  
Son olarak açılmasını istediğimiz form’u seçerek işlemimizi tamamlayalım.

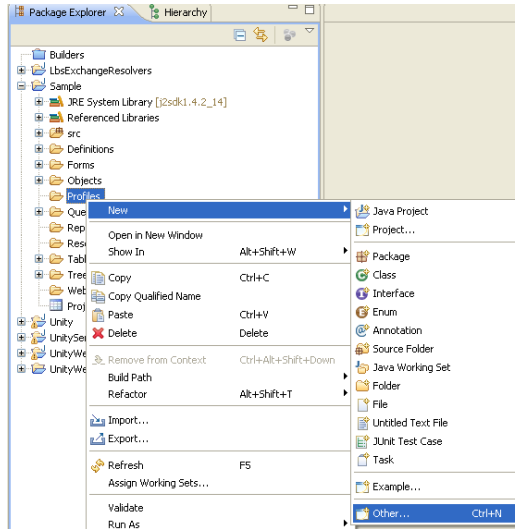


### 3.7. Profil

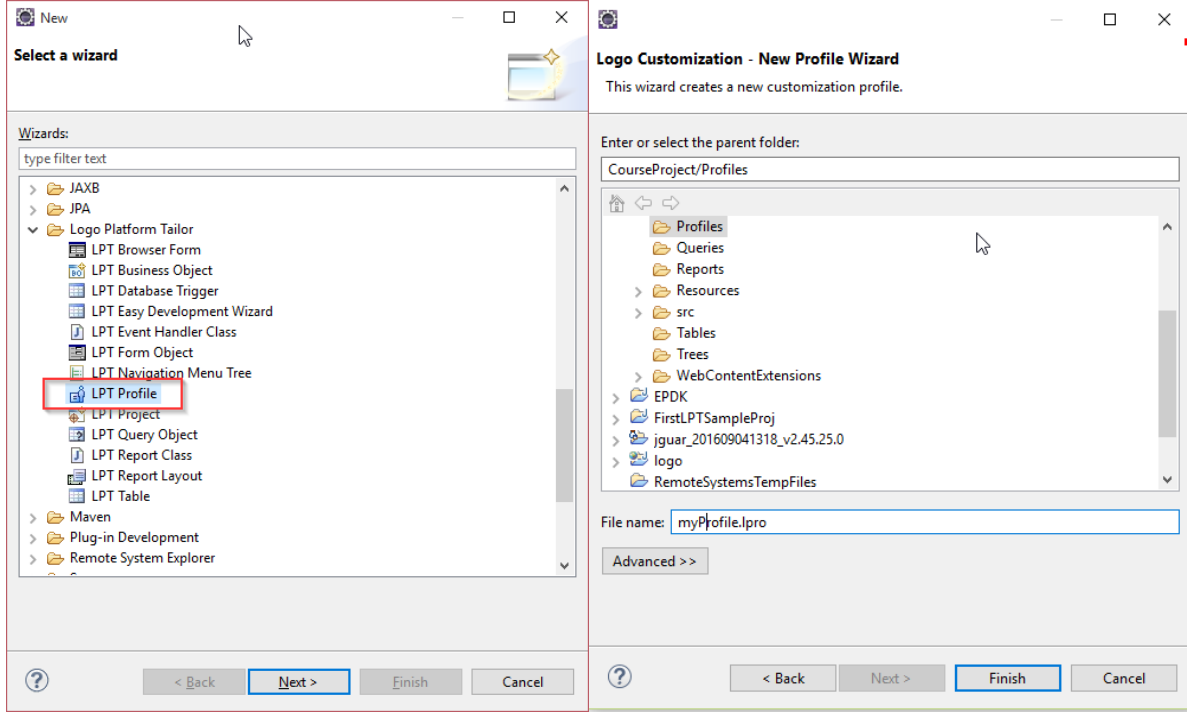
Profiller j-guar kullanıcılarına yönelik uyarlama içerikleri barındıran yapılardır. Her bir profil bir tree yapısına gereksinim duyar. Tree içerisinde tanımlanmış olan uyarlamalar sadece ilgili profil takılı olan kullanıcılar için geçerli olur.

**Önemli Not:** bir uyarlama projesinde birden fazla profil tanımlanabilir.

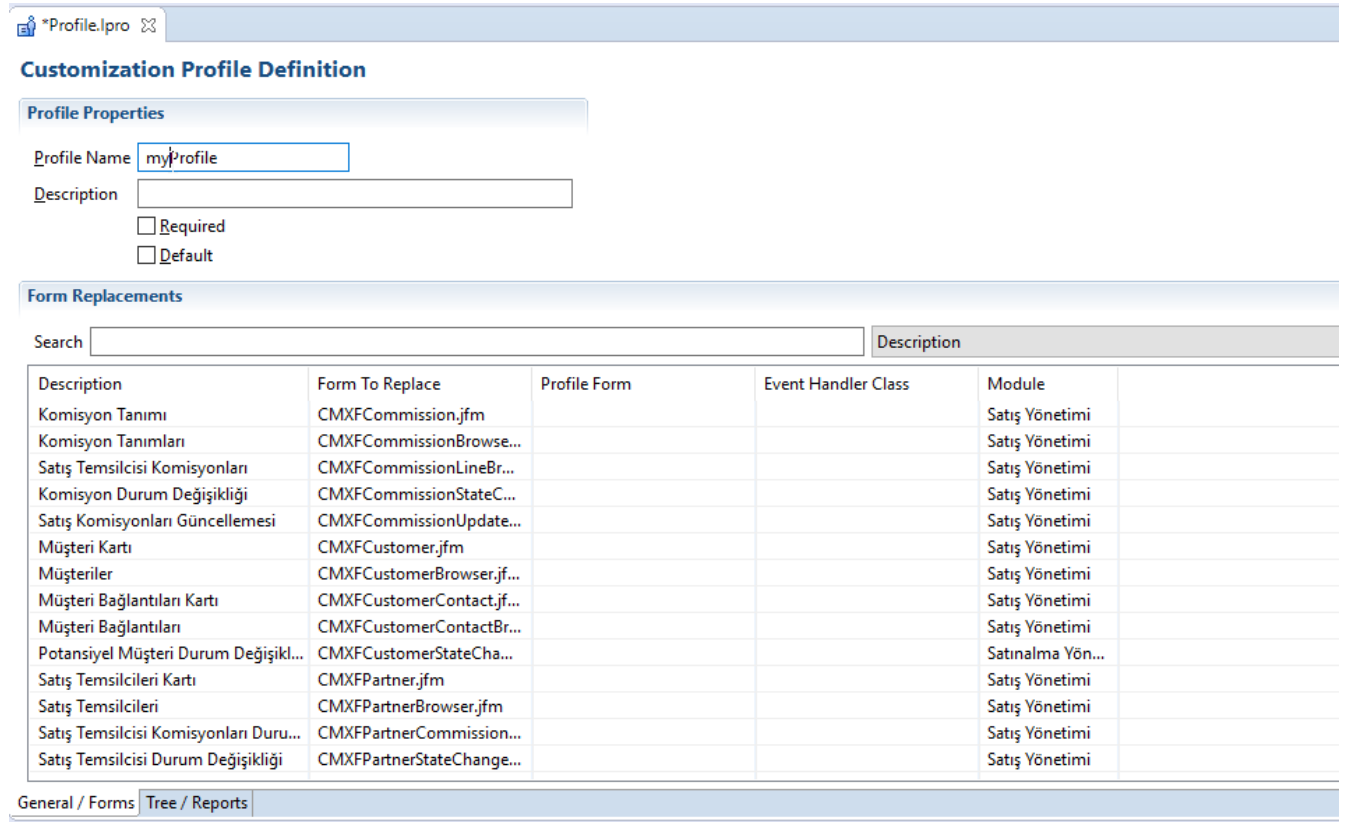
Şimdi profiline nasıl tanımlanacağına bakalım. **Profiles** klasörü üzerinde sağ click yaparak menüden “New” ardından da “Other”ı seçerek sihirbazların yer aldığı ekranı açalım.



Gelen sihirbaz sayfasında “LPT Profile” seçeneğini işaretleyelim ve “Next” düğmesine basalım. Gelen sayfada profili isimlendirerek “Finish” düğmesine basalım ve sihirbazı sonlandıralım.



Sihirbaz kapandığında profil sayfası ekrana gelecektir.



Profil sayfasında j-guar ürününde bulunan bütün form objeleri yer almaktadır. Bu formlar uyarlama olarak tanımlanmış olan formlarla değiştirilebilir.

Örneklendirelim. **3.4.2.** nolu bölümde Cari hesap kartına ilişkin bir uyarlama yapmıştık şimdi üründe cari hesap kartı çağırıldığında orjinal kartın değil uyarlama kartımızın gelmesini sağlayalım. Öncelikle cari hesap kartını listede bulalım ve üzerinde çift klik yapalım açılan pop up ekranında uyarlama form listesi gelecektir, bu listeden uyarlama cari hesap kartımızı seçelim ve “**Select**” düğmesine basarak pop up’ı kapatalım.

**Customization Profile Definition**

**Profile Properties**

Profile Name: myProfile

Description: [Empty]

Required

Default

**Form Replacements**

Search: arpCard

Description	Form To Replace	Profile Form	Event Handler Class	Module
Cari Hesap	FIXFARPCard.jfm			Finans
Cari Hesaplar	FIXFARPCardBrowser.jfm			Finans
Teminatlar	FIXFARPCardCampaignPo...			
Form Basımı İçin Cari Hesap Girişi	FIXFARPCardEntryForFor...			
Cari Hesap Org.Bir. Parametreleri	FIXFARPCardOrgUnitPara...			
Cari Hesap Durum Değişikliği	FIXFARPCardStateChange...			
Cari Hesaplar - İlişkili Kayıtlar - Ha...	FIXFARPCardTransBrowse...			

**Replacing New Form**

Search: [Empty] Name: [Dropdown]

Forms:

- Forms/AlinanHizmet.lfrm
- Forms/CFArpCard.lfrm
- Forms/CFArpCardBrowser.l...
- Forms/CFClassroom.lfrm
- Forms/CFClassroomBrowse...
- Forms/CFCourse.lfrm
- Forms/CFCourseBrowser.lfr...

List only appropriate forms

**Select** **Cancel**

Profiller için tree tanımını yapmamız gerekmektedir, bunun için gridin altında bulunan tab'dan “Tree/Reports” seçeneğine basarak “**Extension Tree and Report**” sayfasını açalım ve burada “**Tree Extensions**” alanına daha önce tanımını yapmış olduğumuz tree'yi seçelim. Bu aşamada şunu belirtmek gerekir grid'de j-guar ürününe ait tüm raporlar yer almaktadır. Bu raporlar uyarlama raporlar ile değiştirilebilir.

**Extension Tree and Reports**

Extension Tree

Tree Extension: Trees/Tree.ltre

Report Replacements

Description	Report To Replace	Profile Report	Module
Org. Birimler Arası Transfer Fişi	AMFRMTransferSlip		Varlık Yönetimi
Org. Birimler Arası Transfer Fişleri	AMFRMAltTransferSlip		Varlık Yönetimi
Maliyet Kaynakları Listesi	CARPResourcesList		Maliyet Muha...
Maliyet Aktiviteleri Listesi	CARPAactivitiesList		Maliyet Muha...
Maliyet Dağıtım Tanımları Listesi	CARPDistributionDefinitio...		Maliyet Muha...
Maliyet Dağıtım Şablonları Listesi	CARPDistributionTemplat...		Maliyet Muha...
Yüklenen Maliyetler Dökümü	CARPAAssignedCostsList		Maliyet Muha...
Dağıtılan Maliyetler Dökümü	CARPDistributedCostsList		Maliyet Muha...
Maliyet Aktivite Türleri Listesi	CARPActivityTypesList		Maliyet Muha...
Maliyet Kaynak Türleri Listesi	CARPResourceTypesList		Maliyet Muha...
Talep Listesi	DERPDemandList		Talep Yönetimi
Talep Karşılama Raporu	DERPDemandMeetingAn...		Talep Yönetimi
Satınalma Faturaları Dökümü	LORPPurchaseInvoiceList		Satınalma Yön...
Alımlar Tablosu	LORPPurchaseTable		Satınalma Yön...
Satınalma Fiyat Analizi	LORPPurchasePriceAnalysis		Satınalma Yön...
Satınalma Vade Farkı Faturası Formu	LOFRMDueDiffInvoice		Satınalma Yön...
Satınalma Sözleşmeleri Formu	LOFRMPurchaseContract		Satınalma Yön...
Toplu Satınalma Sözleşmeleri For...	LOFRMAltPurchaseContra...		Satınalma Yön...

General / Forms Tree / Reports

## 4. DEPLOYMENT

### 4.1. Eclipse üzerinde deploy işlemi

Yapmış olduğumuz uyarlamaların server'a nasıl deploy edileceğini inceleyelim.

Eclipse üzerinden deploy yapabilmek için izlenmesi gereken yol bağlantı ayarlarının kontrol edilmesiyle başlar bağlantının sağlanmasıyla ilgili detaylı bilgi 2.3 nolu maddede anlatılmıştır. Bağlantı kontrolü yapıldıktan sonra package explorer üzerinden projeye tıklanarak aktif hale geçirilir ve **deploy** tuşuna basılır.

Java EE - CourseProject/Profiles/Profile.lpro - Eclipse

File Edit Navigate Search Project LPT Run Window Help

Project Explorer

- src
  - (default package)
  - com.acme.customization
  - com.EasyLPT\_forCZRT.customization
    - AlinanHizmetHandler.java
    - CFArpCardHandler.java
    - CFMachineCodeHandler.java
    - CFMachineCodesBrowserHandler.java
    - RPMachineCodes.java
    - TestformHandler.java
  - com.LPT.customization
- Web App Libraries [logo]
- Definitions
- Forms
- FormTemplates
- Objects

Extension Tree and Reports

Extension Tree

Tree Extension: Trees/Tree.ltre

Report Replacements

Search

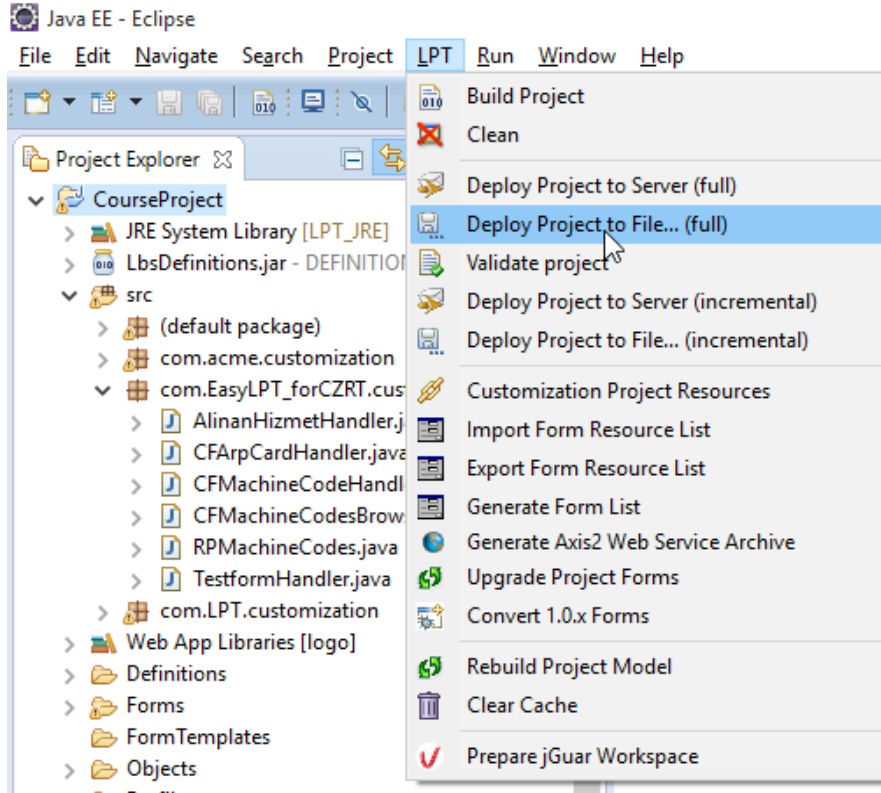
Description	Rep
Org. Birimler Arası Transfer Fişi	AMF
Org. Birimler Arası Transfer Fişleri	AMF
Maliyet Kaynakları Listesi	CAR
Maliyet Aktiviteleri Listesi	CAR
Maliyet Dağıtım Tanımları Listesi	CAR

Deploy işlemi başladıktan sonra eclipse'in sağ alt kısmında uyarlamamanın sever'a gönderimi ile ilgili ilerleme durumu takip edilebilir.



## 4.2. Ürün içerisinde deploye işlemi

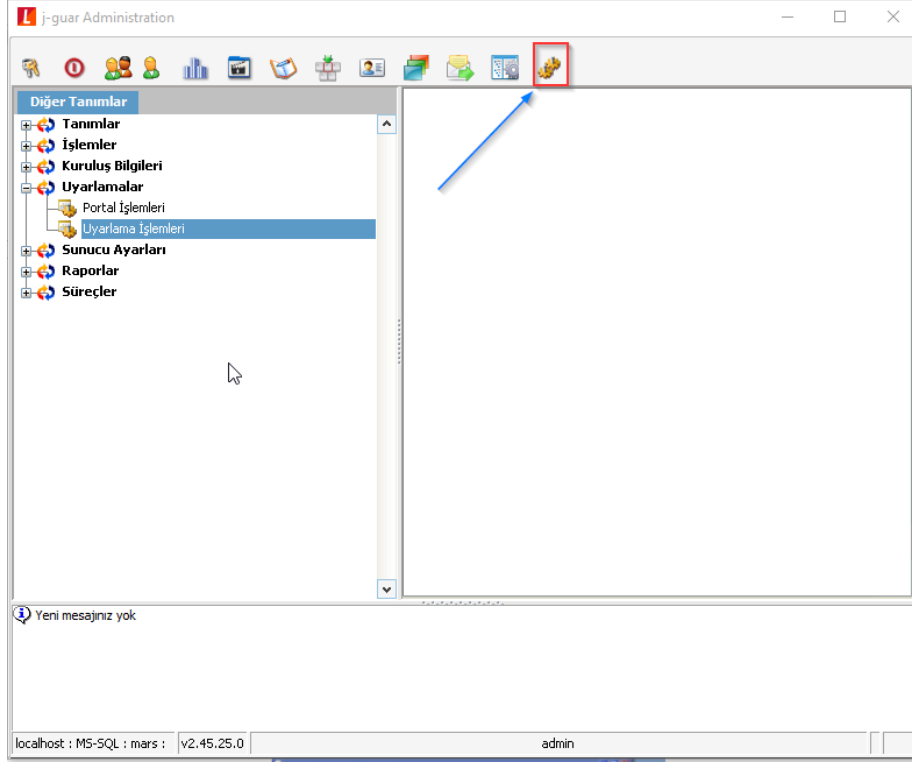
Ürün içerisinden **deploy** yapabilmek için öncelikle **lpdd** uzantılı uyarlama dosyası oluşturulması gerekmektedir. Bunun için eclipse'in logo menüsünden "**Deploy project to file (full)**" elemanı seçilir.



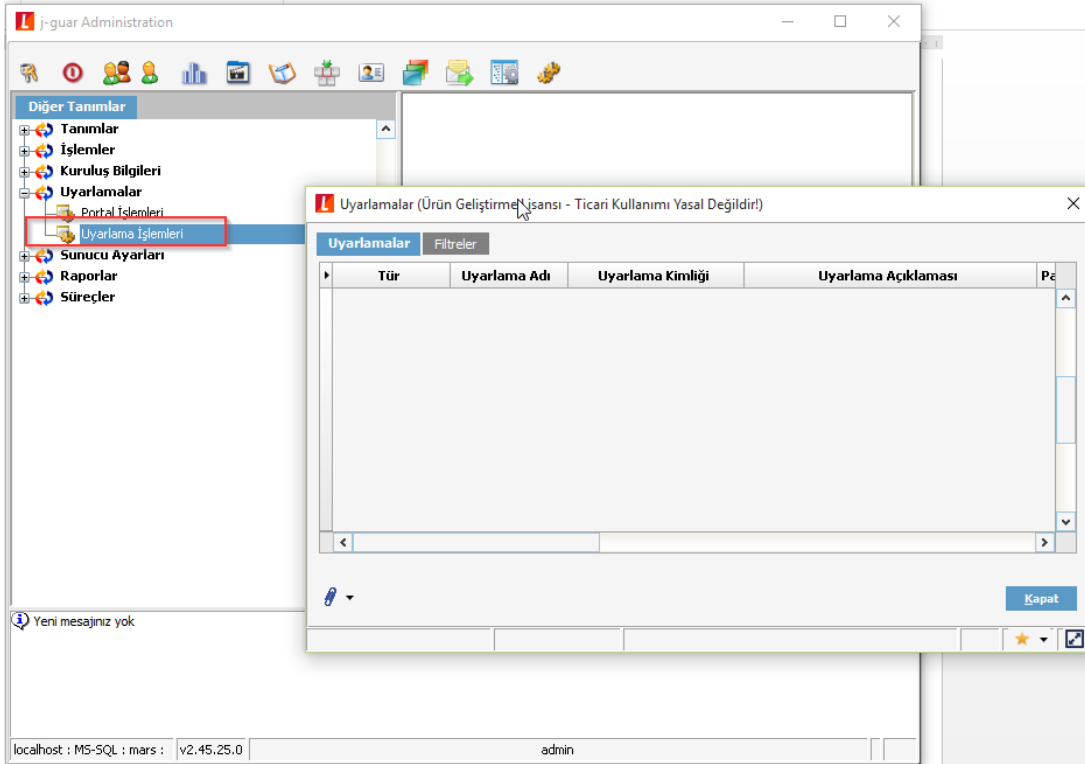
lpdd dosyasının kaydedileceği yer belirlenir, yeri gelmişken lpdd dosyası için isimlendirme otomatik olarak yapılmaktadır.

lpdd uzantılı dosya oluşturulduktan sonra j-guar Administrator uygulaması açılır. Uygulamada çark imgesi olan diğer tanımlar düğmesine basılır.

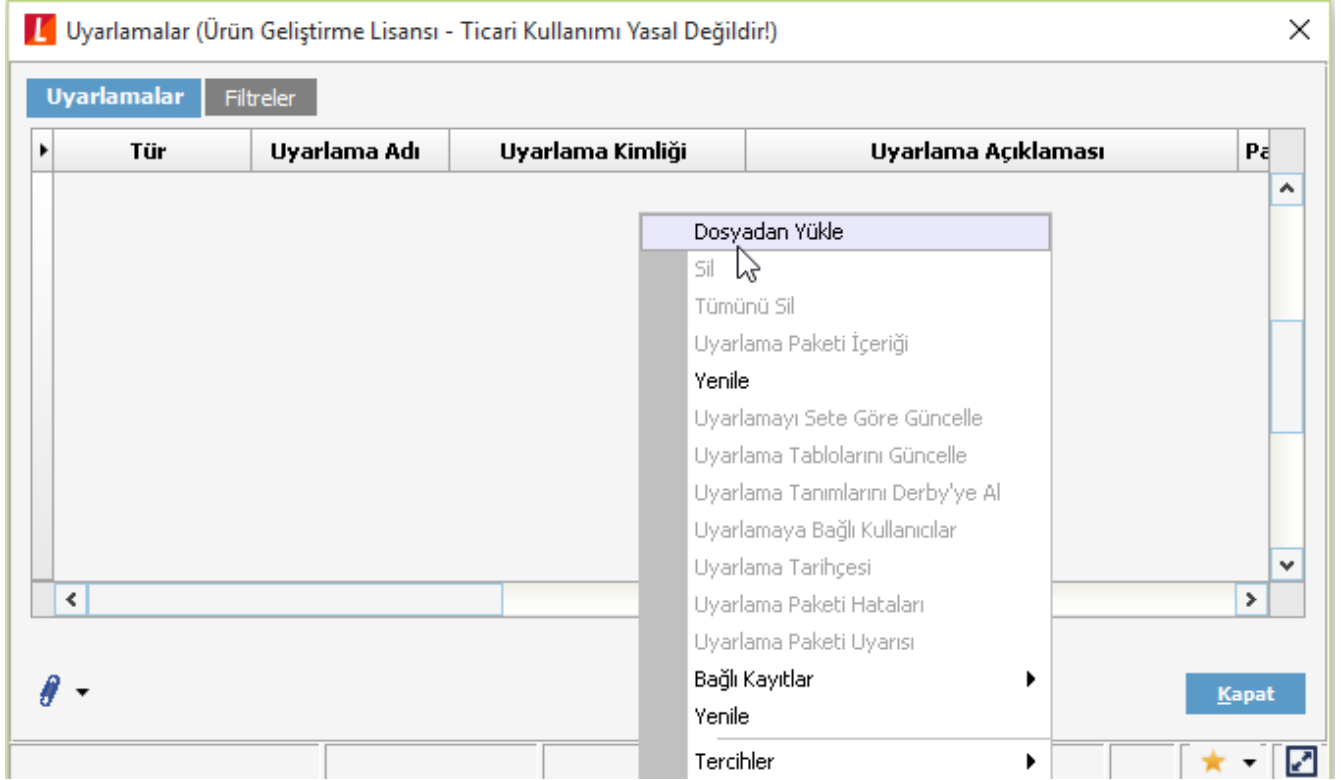




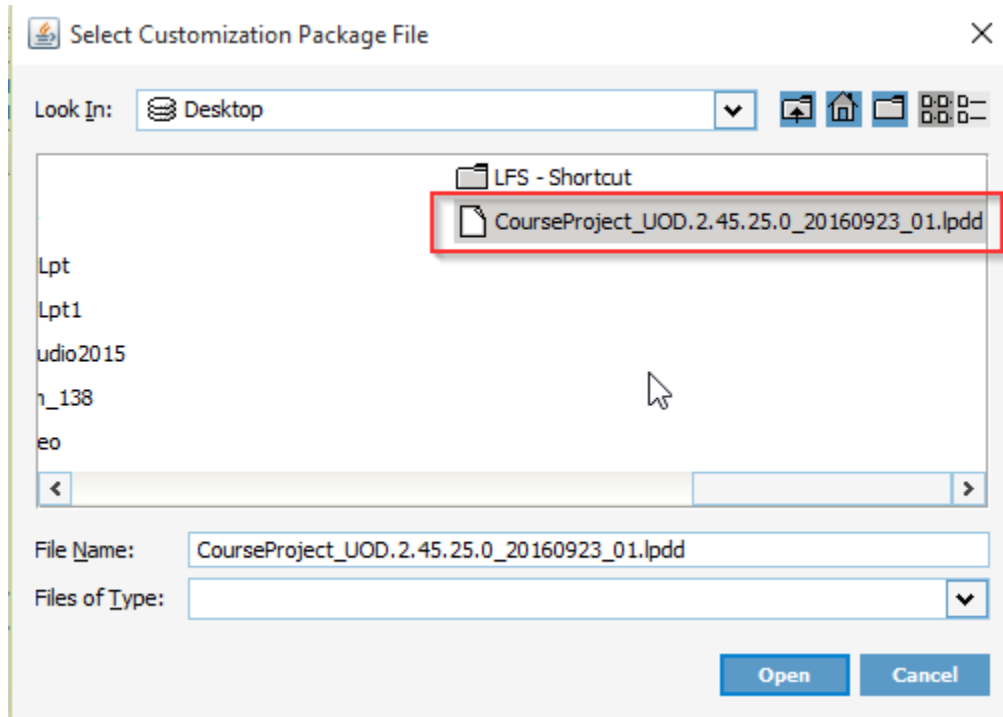
Sol tarafta gelen diğer tanımlar tabında “Uyarlamalar” ardından da “Uyarlama İşlemleri” elemanına tıklanır. Böylelikle uyarlama paketlerin bulunduğu browser açılır..



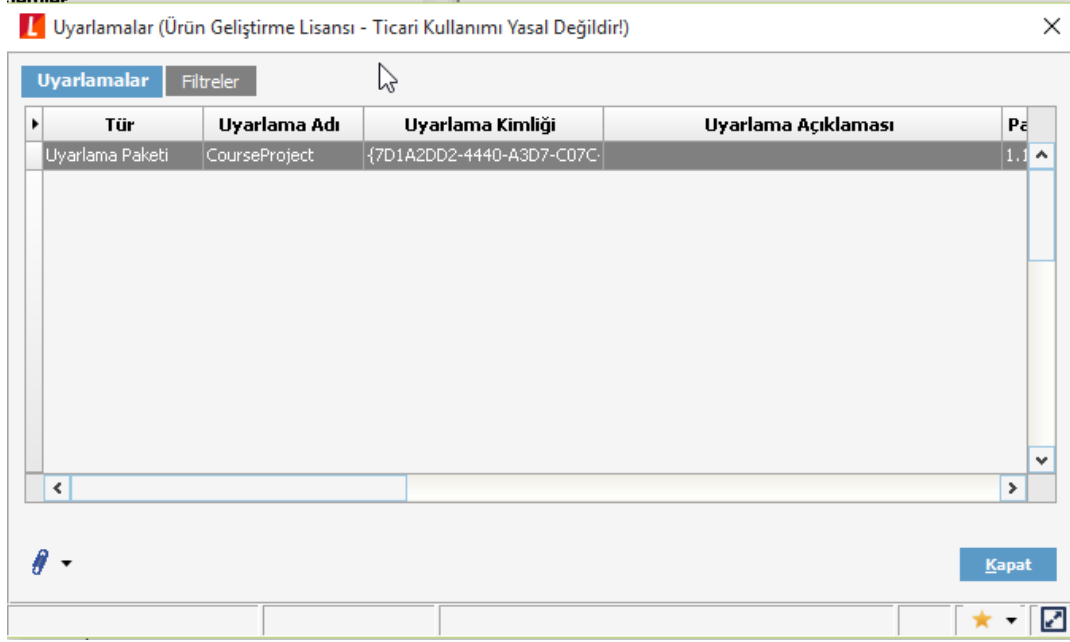
Uyarlamalar browser'ı üzerinde sağ click yapılarak “Dosyadan Yükle” seçeneği ile seçilir.



Lpdd uzantılı uyarlama dosyamızı seçerek **open** düğmesine basılır.

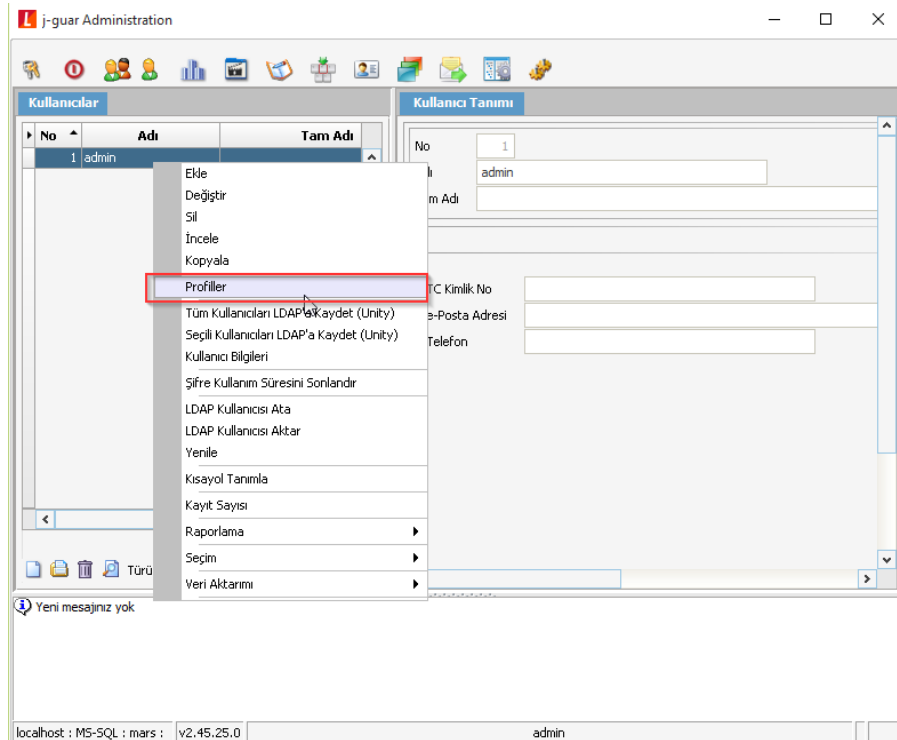


Kısa bir süre içerisinde uyarlama dosyası browser'da yer alacaktır. Böylelikle uyarlama **deploy** edilmiş olacaktır.

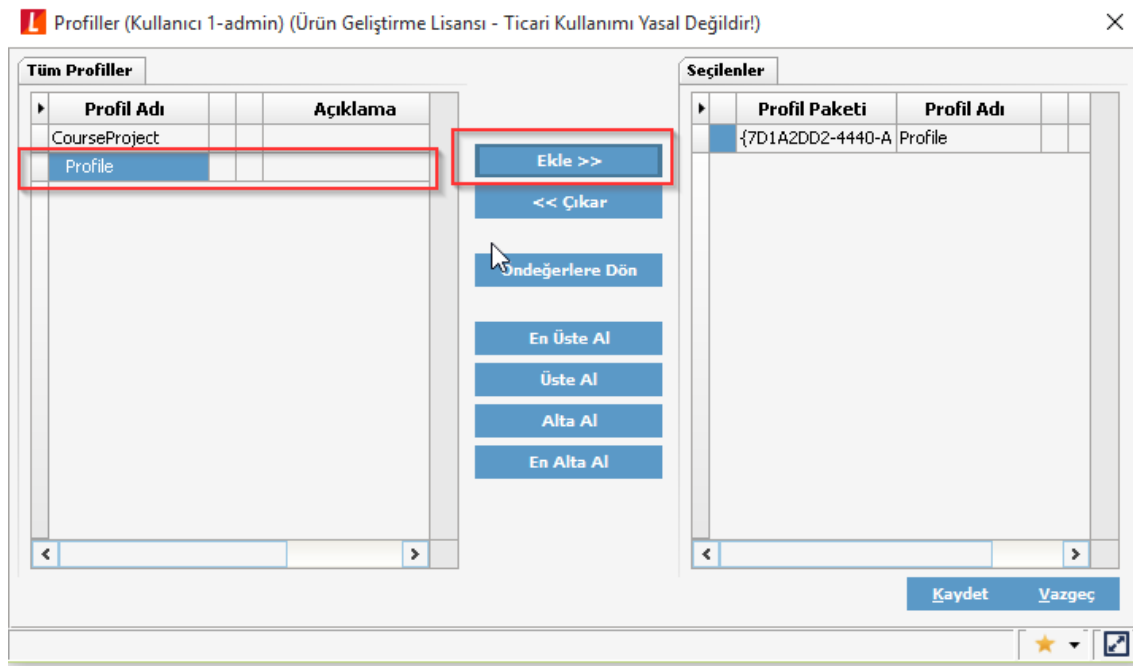


### 4.3. Deploy sonrası işlemler

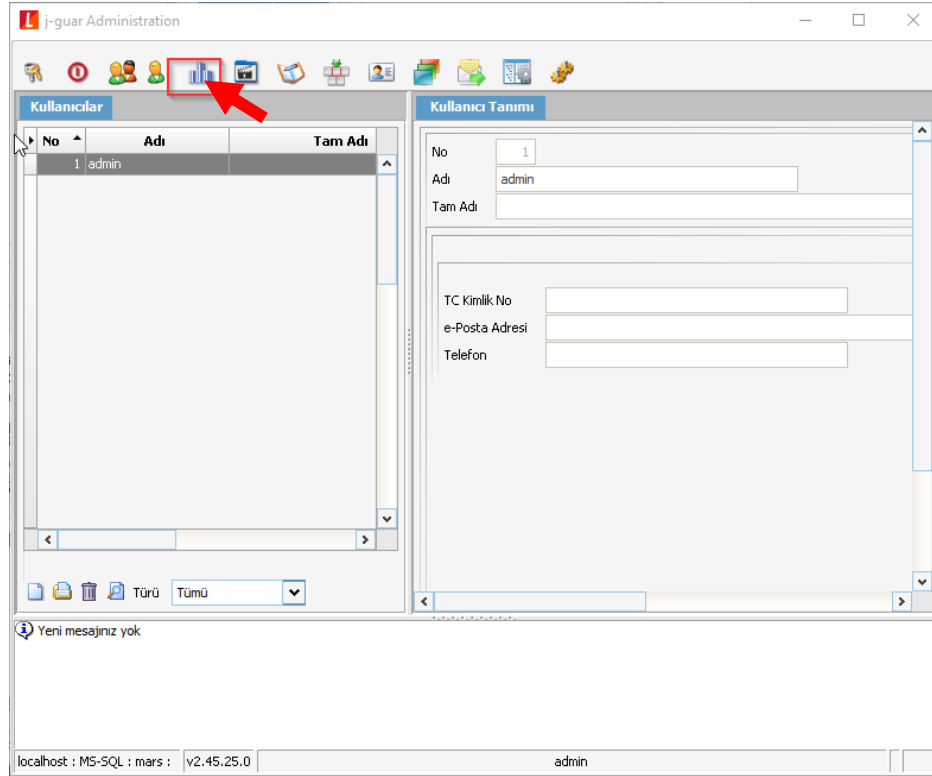
İster eclipse'den yapılınsın ister ürün üzerinden yapılınsın **deploy** sonrası yapılacak işlemler ortaktır. Öncelikle deploy etmiş olduğumuz uyarlamaları bir kullanıcıya profil olarak takmamız gerekmektedir. Kullanıcılar üzerinde sağ klik yapılarak "**Profiller**" seçilir.



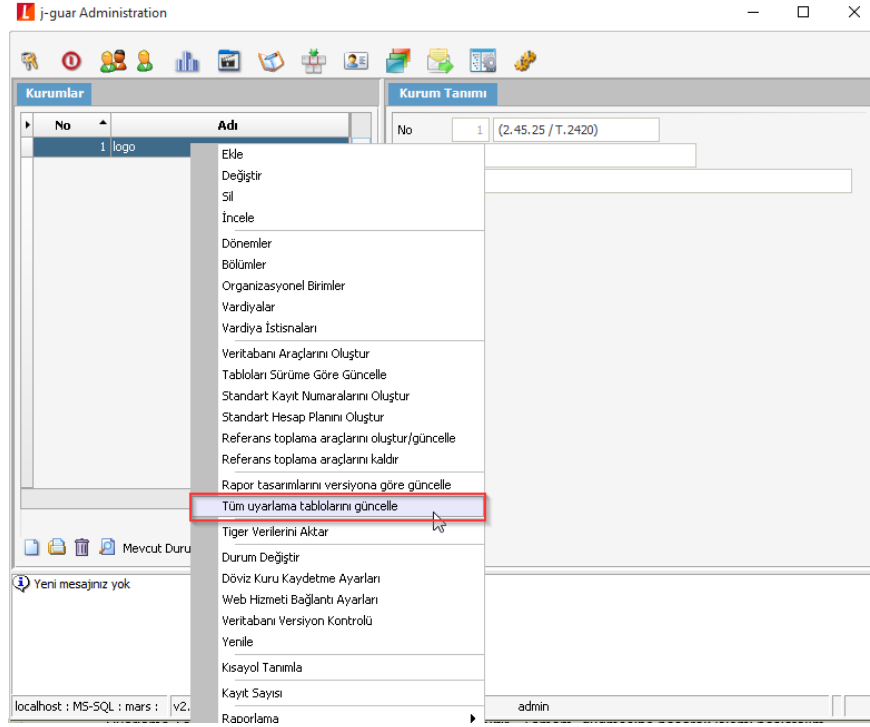
Açılan profil browser'ında sol gridde tüm uyarlama profilleri, sağ grid'de ise sadece seçili kullanıcıya bağlanmış uyarlama profilleri görüntülenmektedir. Sol griddeki "CourseProject" profilini seçtikten sonra ekle düğmesine basarak kullanıcıya profili ekliyoruz ve kaydet düğmesine basarak profil ekleme işlemini sonlandırıyoruz.



Eğer uyarlama olarak yeni bir tablo eklenmişse bu tablonun database'de açılması için deploy etmek yeterli değildir. Deploy yapıldıktan sonra uyarlama tablolarının güncellenmesi gerekmektedir. Bunun için Kurumlar düğmesine basalım.



Ekran gelen kurum tanımımız üzerinde sağ klik yaparak “Tüm Uyarlama Tablolarını Güncelle” seçeneğini seçelim.



Alt kısımda bulunan alanda tabloların güncellenme durumlarıyla ilgili bilgi verilmektedir. Tabloların güncelleme işlemi esnasında yaşanan sıkıntılar bu alanda kullanıcıya iletilmektedir.

