



KURALLAR

1 – Malzeme Açıklama2 Alanını Satır Açıklaması Alanına Getirme	3
2 – Cari Hesap Son Bakiye Kuralı	3
3 - Description ile Name Alanını Deęiřtirme	3
4- e-Arřiv Faturalarında Sevkiyat Adresi Bilgilerini Getirme	4
5- Genel Açıklamalar Kısımına Cari Hesap Bakiyesini Getirme	5
6 - Detaylı Bilgi Alanını Genel Açıklama Alanına Getirme	6
7 - Karma Koli Kuralı	6
8 - Karma Koli Malzeme Kuralı	6
9 - Sipariř Numarasının Sipariř Fiř Numarasına Getirilmesi	7
10 - Barkod Numarasının Fatura Satırına Getirme Kuralı	7
11 - Sipariř No ve Tarihinin Açıklama 3 ve 4 satırından okunması.....	7
12 - Tedarikçi Kodunun Fatura Satırına Getirilmesi	8
13 - İhracat faturalarında satır miktar toplamının açıklamaya getirilmesi	8
14 - Malzemenin ingilizce kod ve açıklamasının getirilmesi.....	9
15 - Hizmet kartı açıklama 2'nin açıklamaya getirilmesi	10
16 - Tanımlı alanların Başlık ile beraber açıklamalar alanına getirilmesi	11
17 - Tesis devri kuralı	11
18 - Ana birim kodu getirme	13
19 - Ana birim çevrim kuralı	14
20 - Gelen faturada satırları tek bir malzeme ile eşleme	14
21 - Dovizli faturalarda Yalnız TL ibaresinin kaldırılması	15
22 - İş yeri bazlı adres bilgilerinin e-faturaya getirilmesi	15
23 - Detaylı bilgi alanının “;” karakteri olmadan genel açıklamaya getirilmesi.....	16
24 - Varyant kodu ve açıklamasının malzeme kodu ve açıklamasına getirilmesi	17
25 - Malzeme açıklaması ile malzeme kodunun deęiřtirilmesi (cari hesap kodu kořullu)	18
26 - Malzeme kodu ile üretici kodunun, malzeme açıklaması ile açıklama2 nin yanyana gelmesi ...	18
27 - Ürün menřei bilgisinin satır açıklamasına getirilmesi (ihracat fatura kořullu)	19
28 - Satır açıklaması dolu ise satır açıklamasının malzeme açıklamasına gelmesi	20





29 - Satır açıklamasının malzeme koduna getirilmesi (cari hesap kodu koşullu)	20
30 - Perakende satış faturasında kdvli birim fiyatı satıra getirme	21
31 - İrsaliye numarası ve tarihinin sipariş numarası ve tarihine getirilmesi	21
32 - Doküman izleme numarasının "BAYINO" alanına getirilmesi (cari hesap kodu koşullu)	21
33 - Malzeme özel kod1 değerinin satır açıklamasına getirilmesi (ihracat fatura koşullu)	22
34 - Satıcı kod ve satıcı adının genel açıklamalara getirilmesi	23
36 - E-arşiv faturada malzeme barkodunun satıra getirilmesi	24
37 - GTIP kodunun satıra getirilmesi.....	25
38 - UQUERY_GETFLD Cari Özel Kod 5 Alanının Cari Açıklama Alanına Yazılması	25
39 - UQUERY_GETFLD Sevkiyat Adresi Adres Bilgilerinin Ek Alana Eklenmesi	25
40 - UQUERY_GETFLD Metodu	26
41 - Operatörler	33





1 – Malzeme Açıklama2 Alanını Satır Açıklaması Alanına Getirme

```
sub ProcessEvent(App)
ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while ok1
name = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_DEF3")
App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.DESRIPTION", name
ok1=App.Doc.NextChild
wend
App.Doc.SaveData
end sub
```

2 – Cari Hesap Son Bakiye Kuralı

```
sub ProcessEvent(App)
ST_CIRef = App.Doc.FieldAsInteger("ARP_LOGICALREF")
ST_CHRRef = CStr(ST_CIRef)
BAKIYE = App.UQUERY_GETFLD(60,"SUM(DEBIT)-SUM(CREDIT)",24,6,FALSE,
"CARDREF="+ST_CHRRef+" AND TOTTP = 1" )
App.Doc.WriteStartElement "ADDITIONALDOCUMENTS/ADDITIONALDOCUMENT"
App.Doc.WriteElementString "ID",Replace(cstr(BAKIYE)," ",".") App.Doc.WriteElementString
"ISSUEDATE", Year(Date) & "-" & Right("0" & Month(Date),
2) & "-" & Right("0" & Day(Date), 2)
App.Doc.WriteElementString "DOCUMENTTYPE",
"BAKIYE" App.Doc.WriteEndElement
App.Doc.SaveData
end sub
```

3 - Description ile Name Alanını Değiştirme

```
sub ProcessEvent(App)
TAX=App.Doc.FieldAsString("ARP_CODE")
If TAX= "120.01.002.0" then
rOk=App.Doc.FirstRoot
while rOk
ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while ok1
KOD =
App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_CODE")
DEF =
App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_DEF")
App.Doc.ModifyData
"TRANSACTIONS.TRANSACTION.MASTER_DEF", KOD
App.Doc.ModifyData
"TRANSACTIONS.TRANSACTION.MASTER_CODE",
DEF ok1=App.Doc.NextChild
```





```
wend  
rOk=App.Doc.NextRoot  
wend  
App.Doc.SaveData  
end If  
end sub
```

4- e-Arşiv Faturalarında Sevkiyat Adresi Bilgilerini Getirme

Aşağıdaki 2 kural ile sevkiyat adresi açıklaması, sevkiyat adresi il, ilçe ve adres bilgilerini fatura ubl'ine getirebilirsiniz;

Sevkiyat adresi açıklaması

```
sub ProcessEvent(App)  
DETAY = App.Doc.FieldAsString("SHIPLOC_DEF")  
App.Doc.WriteStartElement "ADDITIONALDOCUMENTS/ADDITIONALDOCUMENT"  
App.Doc.WriteElementString "ID", DETAY  
App.Doc.WriteElementString "ISSUEDATE", Year(Date) & "-" & Right("0" & Month(Date), 2)  
& "-" & Right("0" & Day(Date), 2)  
App.Doc.WriteElementString "DOCUMENTTYPE", "SEVKIYAT"  
App.Doc.WriteEndElement  
App.Doc.SaveData  
end sub
```

İl, ilçe, adres

```
sub ProcessEvent(App)  
DETAY =  
App.Doc.FieldAsString("EINVOICE_EXTENSIONS.EXTENSIONS.SHIPINFO_ADDRES")  
App.Doc.WriteStartElement "ADDITIONALDOCUMENTS/ADDITIONALDOCUMENT"  
App.Doc.WriteElementString "ID", DETAY  
App.Doc.WriteElementString "ISSUEDATE", Year(Date) & "-" & Right("0" & Month(Date), 2)  
& "-" & Right("0" & Day(Date), 2)  
App.Doc.WriteElementString "DOCUMENTTYPE",  
"ADRES" App.Doc.WriteEndElement  
App.Doc.SaveData  
DETAY =  
App.Doc.FieldAsString("EINVOICE_EXTENSIONS.EXTENSIONS.SHIPINFO_TOWNDIST")  
App.Doc.WriteStartElement "ADDITIONALDOCUMENTS/ADDITIONALDOCUMENT"  
App.Doc.WriteElementString "ID", DETAY  
App.Doc.WriteElementString "ISSUEDATE", Year(Date) & "-" & Right("0" & Month(Date), 2)  
& "-" & Right("0" & Day(Date), 2)  
App.Doc.WriteElementString "DOCUMENTTYPE", "İLCE"
```





```
App.Doc.WriteEndElement
App.Doc.SaveData
DETAY =
App.Doc.FieldAsString("EINVOICE_EXTENSIONS.EXTENSIONS.SHIPINFO_CITY")
App.Doc.WriteStartElement "ADDITIONALDOCUMENTS/ADDITIONALDOCUMENT"
App.Doc.WriteElementString "ID", DETAY
App.Doc.WriteElementString "ISSUEDATE", Year(Date) & "-" & Right("0" & Month(Date), 2)
& "-" & Right("0" & Day(Date), 2)
App.Doc.WriteElementString "DOCUMENTTYPE",
"SEHIR" App.Doc.WriteEndElement
App.Doc.SaveData
end sub
```

tasarıma da gelmesi için;

```
<xsl:for-each select="//n1:Invoice/cac:AdditionalDocumentReference"> <xsl:if
test = "cbc:DocumentType = 'SEVKIYAT'"> <xsl:value-of
select="cbc:ID"></xsl:value-of> </xsl:if> </xsl:for-each>
<xsl:for-each select="//n1:Invoice/cac:AdditionalDocumentReference"> <xsl:if test
="cbc:DocumentType = 'ADRES'"> <xsl:value-of select="cbc:ID"></xsl:value-of>
</xsl:if> </xsl:for-each>
<xsl:for-each select="//n1:Invoice/cac:AdditionalDocumentReference"> <xsl:if
test = "cbc:DocumentType = 'SEHIR'"> <xsl:value-of select="cbc:ID"></xsl:value-
of> </xsl:if> </xsl:for-each>
<xsl:for-each select="//n1:Invoice/cac:AdditionalDocumentReference"> <xsl:if test
="cbc:DocumentType = 'ILCE'"> <xsl:value-of select="cbc:ID"></xsl:value-of>
</xsl:if> </xsl:for-each>
```

5- Genel Açıklamalar Kısımına Cari Hesap Bakiyesini Getirme

```
sub ProcessEvent(App)
ST_CIRef = App.Doc.FieldAsInteger("ARP_LOGICALREF")
ST_CHRRef = CStr(ST_CIRef)
BAKIYE = App.UQUERY_GETFLD(60,"SUM(DEBIT)-SUM(CREDIT)",24,6,FALSE,
"CARDREF="+ST_CHRRef+" AND TOTTYP = 1" )
App.Doc.WriteStartElement "ADDITIONALDOCUMENTS/ADDITIONALDOCUMENT"
App.Doc.WriteElementString "ID",Replace(cstr(BAKIYE)," ",".") App.Doc.WriteElementString
"ISSUEDATE", Year(Date) & "-" & Right("0" & Month(Date),
2) & "-" & Right("0" & Day(Date), 2)
App.Doc.WriteElementString "DOCUMENTTYPE",
"BAKIYE" App.Doc.WriteEndElement
App.Doc.SaveData
end sub
```





6 - Detaylı Bilgi Alanını Genel Açıklama Alanına Getirme

```
sub ProcessEvent(App)
DETAY = App.Doc.FieldAsString("ITEXT")
App.Doc.ModifyData "NOTES5",DETAY
App.Doc.SaveData
end sub
```

7 - Karma Koli Kuralı

“Karma Koli satırına KDV Oranı ve Toplamı gelsin fakat alt satırlar silinsin, e-Fatura gönderirken ise dizaynda ve UBL’de bu satırlar silinsin” isteniyorsa eğer, aşağıdaki kural kullanılmalıdır;

```
sub ProcessEvent(App)
okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
While okay
sTip=App.Doc.FieldAsInteger("TRANSACTIONS.TRANSACTION.TYPE")
if sTip=6 Then
App.Doc.ModifyData
"TRANSACTIONS.TRANSACTION.VAT_RATE",18 elseif sTip=7 Then
App.Doc.RemoveChild "TRANSACTION.TYPE"
end if
okay = App.Doc.NextChild
Wend
App.Doc.SaveData
end sub
```

8 - Karma Koli Malzeme Kuralı

Karma Koli’nin içerisindeki malzemelerin faturada görünmemesi için aşağıdaki kural kullanılabilir.

Doküman: e-Fatura/e-Arşiv

Olay: Onaylanmadan/İmzalanmadan Önce

```
sub ProcessEvent(App)
okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
While okay
sTip=App.Doc.FieldAsInteger("TRANSACTIONS.TRANSACTION.TYPE")
if sTip=6 Then
App.Doc.ModifyData
"TRANSACTIONS.TRANSACTION.VAT_RATE",18 elseif sTip=7 Then
```





```
App.Doc.RemoveChild "TRANSACTION.TYPE"  
end if  
okay = App.Doc.NextChild  
Wend  
App.Doc.SaveData  
end sub
```

9 - Sipariş Numarasının Sipariş Fiş Numarasına Getirilmesi

```
sub ProcessEvent(App)  
ORDER =  
App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.ORDER_FICHENO")  
ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")  
while ok1  
App.Doc.ModifyData "CUSTORDNO", ORDER  
ok1=App.Doc.NextChild  
wend  
App.Doc.SaveData  
end sub
```

10 - Barkod Numarasının Fatura Satırına Getirme Kuralı

```
sub ProcessEvent(App)  
rOk=App.Doc.FirstRoot  
while rOk  
ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")  
while ok1  
barkod = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.BARCODE")  
App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.ITEM_SUPPLIERCODE", barkod  
ok1=App.Doc.NextChild  
wend  
rOk=App.Doc.NextRoot  
wend  
App.Doc.SaveData  
end sub
```

11 - Sipariş No ve Tarihinin Açıklama 3 ve 4 satırından okunması

```
sub ProcessEvent(App)  
rOk=App.Doc.FirstRoot  
while rOk  
ST_DEGISKEN1 =App.Doc.FieldAsString("NOTES3")  
ST_DEGISKEN2 = App.Doc.FieldAsString("NOTES4")
```





```
ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while ok1
App.Doc.ModifyData "CUSTORDNO",ST_DEGSKEN1
App.Doc.ModifyData "ORDER_FICHENO",ST_DEGSKEN1
App.Doc.ModifyData "ORDER_FICHEDATE",ST_DEGSKEN2
ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
ok1=App.Doc.NextChild
wend
rOk=App.Doc.NextRoot
wend
App.Doc.SaveData
end sub
```

12 - Tedarikçi Kodunun Fatura Satırına Getirilmesi

```
sub ProcessEvent(App)
rOk=App.Doc.FirstRoot
while rOk
okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while okay
OZELKOD =
App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_AUXIL_CODE")
App.Doc.ModifyData
"TRANSACTIONS.TRANSACTION.ITEM_CUSTOMERCODE", OZELKOD
okay=App.Doc.NextChild
wend
rOk=App.Doc.NextRoot
wend
App.Doc.SaveData ' Yapılan değişiklikler
kaydediliyor end sub
```

13 - İhracat faturalarında satır miktar toplamının açıklamaya getirilmesi

```
sub ProcessEvent(App)
Type1 = App.Doc.FieldAsString("EXIM_FICHE_TYPE")
If Type1 = 2 or 3 Then
b = 0.0
okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while okay
Type2 = App.Doc.FieldAsInteger("TRANSACTIONS.TRANSACTION.TYPE")
If Type2 = 0 Then
Quantity = App.Doc.FieldAsFloat("TRANSACTIONS.TRANSACTION.QUANTITY")
a = (Quantity) + b
```





```
b = a
end if
okay=App.Doc.NextChild
wend
App.Doc.WriteStartElement "ADDITIONALDOCUMENTS/ADDITIONALDOCUMENT"
App.Doc.WriteString "ID", b
App.Doc.WriteString "ISSUEDATE", Year(Date) & "-" & Right("0" & Month(Date), 2) & "-" & Right("0" & Day(Date), 2)
App.Doc.WriteString "DOCUMENTTYPE", "TOPLAMMIKTAR"
App.Doc.WriteEndElement
end if
App.Doc.SaveData
end sub
```

Tasarıma gelmesi için ek alan xslt kodu:

```
<xsl:for-each select="//n1:Invoice/cac:AdditionalDocumentReference">
<xsl:if test ="cbc:DocumentType = 'TOPLAMMIKTAR'">
  <xsl:value-of select="cbc:ID"/>
</xsl:if>
</xsl:for-each>
```

14 - Malzemenin ingilizce kod ve açıklamasının getirilmesi

İNGİLİZCE KOD VE AÇIKLAMA

```
sub ProcessEvent(App)
'Tip = App.Doc.FieldAsInteger("EXIM_FICHE_TYPE")
'if Tip = 5 then
strDBUser = "sa" ' SQL Server kullanıcı adı strDBPass =
"123456" ' SQL Server kullanıcısının şifresi strDBServer =
"SERRASARIKAYA" ' SQL Server strDBName = "TIGER" '
Veritabanı adı
' ADODB bağlantısı için standart yaratılması gereken nesne ve yazılması gereken cümleler
Set conn = CreateObject("ADODB.Connection")
conn.ConnectionString = "Provider=SQLOLEDB.1;Persist Security Info=False;User ID=" + strDBUser
+ ";Password=" + strDBPass + ";Initial Catalog=" + strDBName+ ";Data Source=" +
strDBServer conn.Open
Set rs = CreateObject("adodb.recordset")
'rs.ActiveConnection = conn
```





```
ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while ok1
    Ref =
App.Doc.FieldAsInteger("TRANSACTIONS.TRANSACTION.MASTER_ORGLOGICALREF")
    strRef = CStr(Ref) ' Referans Bilgisi Stringe çevriliyor
    strQuery = " SELECT A.FIELDCONT,B.FIELDCONT FROM LG_001_LNGEXCSETS
A,LG_001_LNGEXCSETS B WHERE A.DOCREF=B.DOCREF AND A.DOCID=1 AND A.LANGID=2 AND
A.FIELDID=10 AND B.FIELDID=1 AND A.DOCREF= " + strRef ' Sorgu oluşturuluyor
    rs.Open strQuery, conn ' Queryimizi çalıştırdığımız kısım
    strCode = rs.Fields(0).value ' ADODB bağlantısından sorgudan dönen değer okunuyor
    strAciklama = rs.Fields(1).value
    App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.MASTER_CODE", strCode
    App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.MASTER_DEF", strAciklama

    do while (rs.EOF = false)

        rs.MoveNext()
    loop
    rs.close
    App.Doc.SaveData
    ok1=App.Doc.NextChild
wend
'end if
' rs.close' RecordSet kapatılıyor
conn.close' ADODB bağlantısı kapatılıyor

end sub
```

15 - Hizmet kartı açıklama 2'nin açıklamaya getirilmesi

```
sub ProcessEvent(App)
okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while okay
    sType = App.Doc.FieldAsInteger("TRANSACTIONS.TRANSACTION.TYPE")
    If sType = 4 Then
        rOK=App.Doc.FirstChild("SALES_SERVICES.SERVICE")
        while rOk
            ACIKLAMA2 = App.Doc.FieldAsString("SALES_SERVICES.SERVICE.DESRIPTION2")
            rOk=App.Doc.NextChild
        wend
        App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.MASTER_DEF",
        ACIKLAMA2 end if
    okay=App.Doc.NextChild
end while
end sub
```





```
wend  
App.Doc.SaveData  
end sub
```

16 - Tanımlı alanların Başlık ile beraber açıklamalar alanına getirilmesi

```
sub ProcessEvent(App)  
    DETAY = App.Doc.FieldAsString("DEFNFLDSLST.DEFNFLD.TEXTFLDS1")  
    DETAY1 = App.Doc.FieldAsString("DEFNFLDSLST.DEFNFLD.TEXTFLDS2")  
    App.Doc.ModifyData "NOTES5", "Başlık: " + DETAY  
    App.Doc.ModifyData "NOTES6", "Başlık2: " + DETAY1  
    App.Doc.SaveData  
end sub
```

17 - Tesis devri kuralı

Özelkod: **TESISDEVRI**

```
sub ProcessEvent(App)  
    OZELKOD=App.Doc.FieldAsString("AUXIL_CODE")  
    If OZELKOD= "TESISDEVRI" then  
        rOk=App.Doc.FirstRoot  
        while rOk  
            ToplamTutarA= App.Doc.FieldAsFloat("TOTAL_NET")  
            ToplamTutarB= App.Doc.FieldAsFloat("TC_NET")  
            Geneltutar=App.Doc.FieldAsFloat("EXPENSE_DISTRB")  
            Geneltutarnew=CDbl(geneltutar)  
            ToplamTutar1=CDbl(ToplamTutarA)  
            ToplamTutar2=CDbl(ToplamTutarB)  
            Toplambedel1=Geneltutarnew+ToplamTutarA  
            Toplambedel2=Geneltutarnew+ToplamTutarA  
            App.Doc.ModifyData "TOTAL_VAT",Replace(cstr(Geneltutarnew),",",".")  
            App.Doc.ModifyData "TOTAL_NET",Replace(cstr(toplambedel1),",",".")  
            App.Doc.ModifyData "TC_NET",Replace(cstr(toplambedel2),",",".")  
            ok1=App.Doc.FirstChild("DISPATCHES.DISPATCH")  
            while ok1  
                App.Doc.ModifyData "TOTAL_VAT",Replace(cstr(Geneltutarnew),",",".")  
                App.Doc.ModifyData "TOTAL_NET",Replace(cstr(toplambedel),",",".")  
                ok1=App.Doc.NextChild  
            wend  
        wend  
    end if  
end sub
```





```
ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while ok1
  App.Doc.ModifyData "VAT_AMOUNT",Replace(cstr(Geneltutarnew),",",".")
  ok1=App.Doc.NextChild
wend

  ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION.EINVOICE_EXTENSIONS.EXTENSIONS.TA
XSUBTOTALS.TAXSUBTOTAL")
  while ok1
    App.Doc.ModifyData "TAXAMOUNT",Replace(cstr(Geneltutarnew),",",".")
    ok1=App.Doc.NextChild
  wend
  ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION.EINVOICE_EXTENSIONS.EXTENSIONS.
TAXSUBTOTALS")
  while ok1
    App.Doc.ModifyData "TAXAMOUNT",Replace(cstr(Geneltutarnew),",",".")

ok1=App.Doc.NextChild
wend

  ok1=App.Doc.FirstChild("EINVOICE_EXTENSIONS.EXTENSIONS.TAXSUBTOTALS.TAXSUBTOTAL")
  while ok1
    App.Doc.ModifyData "TAXAMOUNT",Replace(cstr(Geneltutarnew),",",".")
    App.Doc.ModifyData "PERCENT",18
    ok1=App.Doc.NextChild
  wend
  ok1=App.Doc.FirstChild("EINVOICE_EXTENSIONS.EXTENSIONS.TAXSUBTOTALS"
) while ok1
    App.Doc.ModifyData "TAXAMOUNT",Replace(cstr(Geneltutarnew),",",".")
    ok1=App.Doc.NextChild
  wend
  rOk=App.Doc.NextRoot
wend
App.Doc.SaveData
end If
end sub
```

Dağıtılacak Toplam : KDV Tutarı





Döviz		Kuru	Toplam	Dağılacak Toplam
Raporlama	USD	2,3277	56,27	
İşlem	TL			KDV Oranı 18
			Lot No.	
Sevkiyat Hesabı				
Kodu			<input type="checkbox"/> Teminat Riskini Etkileyecek	
Unvanı			<input type="checkbox"/> Riski Etkileyecek	
Sevkiyat Adresi				
Kodu				
Açıklaması				
İrsaliye Bilgileri			Açıklama	
Türü				
No.				
Belge No.				
Doküman İzleme Numarası			Ödeme Tipi	
			İşlem Yapılmayacak	
Sevkiyat Bilgileri				
Teslim Şekli				
Taahhüt Kodu				
Paket/Koli No.				
Taahhüt Tipi				
Tax Free				
<input type="checkbox"/> Tax Free		Kampanya Kodu		
Pasaport No.				
Kredi Kartı No.				
KDV Muafiyet Sebebi				

18 - Ana birim kodu getirme

```
sub ProcessEvent(App)
    okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
    while okay
        Quantity = App.Doc.FieldAsFloat("TRANSACTIONS.TRANSACTION.QUANTITY")
        'msgbox(Quantity)
        GBKod = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.UNIT_GLOBAL_CODE")
        ok1=App.Doc.FirstChild("UNIT_SETS.UNIT_SET.UNITS.UNIT")
        while ok1
            GBKod2 = App.Doc.FieldAsString("GLOBAL_CODE")
            If GBKod = GBKod2 Then
                value = App.Doc.FieldAsFloat("CONV_FACT1")
                'msgbox(value)
            end if
            ok1=App.Doc.NextChild
        wend
        a = ((Quantity) * (value))
    App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.DESCRPTION",
        a okay=App.Doc.NextChild
    wend
```





```
App.Doc.SaveData  
end sub
```

19 - Ana birim çevrim kuralı

```
sub ProcessEvent(App)  
    b = 0.0  
    okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")  
    while okay  
        Quantity = App.Doc.FieldAsFloat("TRANSACTIONS.TRANSACTION.QUANTITY")  
        'msgbox(Quantity)  
        GBKod = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.UNIT_GLOBAL_CODE")  
        ok1=App.Doc.FirstChild("UNIT_SETS.UNIT_SET.UNITS.UNIT")  
        while ok1  
            GBKod2 = App.Doc.FieldAsString("GLOBAL_CODE")  
            If GBKod = GBKod2 Then  
                value = App.Doc.FieldAsFloat("CONV_FACT1")  
                'msgbox(value)  
            end if  
            ok1=App.Doc.NextChild  
        wend  
        a = ((Quantity) * (value))  
        'msgbox(a)  
        c = b + a  
        b = c  
        'msgbox(b)  
    okay=App.Doc.NextChild  
wend  
    App.Doc.WriteStartElement "ADDITIONALDOCUMENTS/ADDITIONALDOCUMENT"  
    App.Doc.WriteElementString "ID", b  
    App.Doc.WriteElementString "ISSUEDATE", Year(Date) & "-" & Right("0" & Month(Date), 2) & "-" & Right("0" & Day(Date), 2)  
    App.Doc.WriteElementString "DOCUMENTTYPE", "KILOTOPLAM"  
    App.Doc.WriteEndElement  
    App.Doc.SaveData  
end sub
```

20 - Gelen faturada satırları tek bir malzeme ile eşleme

```
sub ProcessEvent(App)  
    ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")  
    while ok1
```





```
Code = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_CODE")
App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.TYPE", "0"
App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.UNITSET_CODE", "05"
App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.UNIT_GLOBAL_CODE",
"C62" App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.UNIT_CODE", "ADET"
App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.MASTER_CODE", "TM-001"
ok1=App.Doc.NextChild
wend
App.Doc.SaveData
end sub
```

21 - Dovizli faturalarda Yalnız TL ibaresinin kaldırılması

```
sub ProcessEvent(App)
sTip=App.Doc.FieldAsInteger("CURR_INVOICE")
if sTip <> Empty Then
App.Doc.RemoveChild "TOTAL_NET_STR"
end if
App.Doc.SaveData
end sub
```

22 - İş yeri bazlı adres bilgilerinin e-faturaya getirilmesi

```
sub ProcessEvent(App)
strDBUser = "sa" ' SQL Server kullanıcı adı
strDBPass = "xxxxxx" ' SQL Server kullanıcısının şifresi
strDBServer = "xxxxxx" ' SQL Server
strDBName = "xxxxx" ' Veritabanı adı
' ADODB bağlantısı için standart yaratılması gereken nesne ve yazılması gereken cümleler
Set conn = CreateObject("ADODB.Connection")
conn.ConnectionString = "Provider=SQLOLEDB.1;Persist Security Info=False;User ID=" + strDBUser
+ ";Password=" + strDBPass + ";Initial Catalog=" + strDBName + ";Data Source=" +
strDBServer conn.Open
Set rs = CreateObject("adodb.recordset")
Ref = App.Doc.FieldAsString("NUMBER")
strRef = CStr(Ref) ' Referans Bilgisi Stringe çevriliyor
'strQuery = " SELECT A.FIELDCONT,B.FIELDCONT FROM LG_001_LNGEXCSETS
A,LG_001_LNGEXCSETS B WHERE A.DOCREF=B.DOCREF AND A.DOCID=1 AND A.LANGID=2 AND
A.FIELDID=10 AND B.FIELDID=1 AND A.DOCREF=" + strRef ' Sorgu oluşturuluyor
strQuery = "SELECT STREET,ROAD,DOORNR,DISTRICT,CITY FROM L_CAPIDIV WHERE
FIRMNR=1 AND NR IN (SELECT BRANCH FROM LG_001_03_INVOICE WHERE FICHENO =" + """" + strRef
+ """)"
```





```
rs.Open strQuery, conn ' Queryimizi çalıştırdığımız kısım
strStreet = rs.Fields(0).value ' ADODB bağlantısından sorgudan dönen değer
okunuyor

'msgbox(strStreet)
strRoad = rs.Fields(1).value
strDoorNr = rs.Fields(2).value
strDistrict = rs.Fields(3).value
strCity = rs.Fields(4).value
App.Doc.ModifyData "SND_STREET", strStreet
App.Doc.ModifyData "SND_ROAD", strRoad
App.Doc.ModifyData "SND_CITY", strCity
App.Doc.ModifyData "SND_DOORNR", strDoorNr
App.Doc.ModifyData "SND_DISTRICT", strDistrict

do while (rs.EOF = false)

    rs.MoveNext() ' birden fazla değer okunacaksa bir sonraki satıra
konumlanılıyor

loop
rs.close
App.Doc.SaveData

' rs.close' RecordSet kapatılıyor
conn.close' ADODB bağlantısı kapatılıyor

end sub
```

23 - Detaylı bilgi alanının “;” karakteri olmadan genel açıklamaya getirilmesi

KURAL (Her faturada çalışır)

```
sub ProcessEvent(App)
rOk=App.Doc.FirstRoot
while rOk
ST_DETAY=App.Doc.FieldAsString("ITEXT")
App.Doc.ModifyData "NOTES12",Replace((ST_DETAY),"|","\\YS_ST#")
rOk=App.Doc.NextRoot
wend
App.Doc.SaveData
end sub
```





XSLT İlk Adım (Genel Açıklamalar kısmı aranır ve silinir. Alttaki kod yapıştırılır.)

```
<xsl:if test="//n1:Invoice/cbc:Note">
<b>&#160;&#160;&#160;&#160;&#160; Genel
Açıklamalar</b> <br/>
<xsl:for-each select="//n1:Invoice/cbc:Note">
<b>&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;</b>
<xsl:call-template name="insertBreaks">
<xsl:with-param name="pText" select="." />
</xsl:call-template>
</xsl:for-each>
</xsl:if>
```

XSLT En Alt alana gelir. (</xsl:template> bundan sonraya eklenir.)

```
<xsl:template match="text()" name="insertBreaks">
<xsl:param name="pText" select="."/> <xsl:choose>

<xsl:when test="not(contains($pText, '\\YS ST#'))">
<xsl:copy-of select="$pText"/> </xsl:when>

<xsl:otherwise>
<xsl:value-of select="substring-before($pText, '\\YS ST#')"/>
<br /><b>&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;</b>
<xsl:call-template name="insertBreaks">
<xsl:with-param name="pText" select="substring-after($pText,
'\\YS ST#')"/> </xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

24 - Varyant kodu ve açıklamasının malzeme kodu ve açıklamasına getirilmesi

sub ProcessEvent(App)

ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")

while ok1

Var = App.Doc.FieldAsInteger("CANCONFIG")

if Var = 1 then

Code = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.VARIANTCODE")

Desc = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.VARIANTNAME")





```
App.Doc.ModifyData " TRANSACTIONS.TRANSACTION.MASTER_CODE",Code
App.Doc.ModifyData " TRANSACTIONS.TRANSACTION.MASTER_DEF",Desc
end if
ok1=App.Doc.NextChild
wend
App.Doc.SaveData
end sub
```

25 - Malzeme açıklaması ile malzeme kodunun değiştirilmesi (cari hesap kodu koşullu)

```
sub ProcessEvent(App)
TAX=App.Doc.FieldAsString("ARP_CODE")
If TAX= "120.01.002.0" then
rOk=App.Doc.FirstRoot
while rOk
ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while ok1
KOD = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_CODE")
DEF = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_DEF")
App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.MASTER_DEF", KOD
App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.MASTER_CODE", DEF
ok1=App.Doc.NextChild
wend
rOk=App.Doc.NextRoot
wend
App.Doc.SaveData
end If
end sub
```

26 - Malzeme kodu ile üretici kodunun, malzeme açıklaması ile açıklama2 nin yanyana gelmesi

```
sub ProcessEvent(App)
rOk=App.Doc.FirstRoot
while rOk
ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while ok1
ST_DEGISKEN1=App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_CODE")
ST_DEGISKEN2=App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.PROCEDUCER_CODE")

ST_DEGISKEN3=App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_DEF")
```





```
ST_DEGISKEN4=App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_DEF3")
```

```
App.Doc.ModifyData "MASTER_CODE", ST_DEGISKEN1+" / "+ ST_DEGISKEN2
App.Doc.ModifyData "MASTER_DEF", ST_DEGISKEN3+" / "+ ST_DEGISKEN4
ok1=App.Doc.NextChild
wend
rOk=App.Doc.NextRoot
wend
App.Doc.SaveData
end sub
```

27 - Ürün menşei bilgisinin satır açıklamasına getirilmesi (ihracat fatura koşullu)

```
sub ProcessEvent(App)
```

```
    strDBUser = "xx"
    strDBPass = "xxxxx"
    strDBServer = "xx"
    strDBName = "xxxxx"

    Set conn = CreateObject("ADODB.Connection")
    conn.ConnectionString = "Provider=SQLOLEDB.1;Persist Security Info=False;User ID=" + strDBUser
+ ";Password=" + strDBPass + ";Initial Catalog=" + strDBName+ ";Data Source=" +
    strDBServer conn.Open
    Set rs = CreateObject("adodb.recordset")
    rs.ActiveConnection = conn

    okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
    while okay
    Type2 = App.Doc.FieldAsInteger("TRANSACTIONS.TRANSACTION.TYPE")
    If Type2 = 0 then
    intlItemRef = App.Doc.FieldAsInteger("TRANSACTIONS.TRANSACTION.MASTER_ORGLOGICALREF")
    strItemRef = CStr(intlItemRef)
    strQuery = "SELECT DEFINITION_ FROM LG_xxx_SPECODES WHERE"
    strQuery = strQuery + "(((CODETYPE = 8) AND (SPECODETYPE = 1)) OR ((CODETYPE = 8)
AND (SPECODETYPE > 1)))"
    strQuery = strQuery + "OR (((CODETYPE > 8))) AND (((CODETYPE = 8) AND (SPECODETYPE <= 1))
OR ((CODETYPE < 8)))"
    strQuery = strQuery + "AND SPECODE IN (SELECT PRODCOUNTRY FROM LG_xxxx_ITEMS
ITMSC WHERE LOGICALREF =" + """" + strItemRef + """" + ")"
    rs.Open strQuery, conn
    strPro = rs.Fields(0).value
    App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.DESCRPTION", strPro
```





```
do while (rs.EOF = false)
rs.MoveNext()
loop
rs.close
App.Doc.SaveData
end if
okay=App.Doc.NextChild
wend

conn.close
end sub
```

28 - Satır açıklaması dolu ise satır açıklamasının malzeme açıklamasına gelmesi

```
sub ProcessEvent(App)
okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while okay
ACIKLAMA = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.DESRIPTION")
if ACIKLAMA <> Empty then
App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.MASTER_DEF", ACIKLAMA
end if
okay=App.Doc.NextChild
wend
App.Doc.SaveData
end sub
```

29 - Satır açıklamasının malzeme koduna getirilmesi (cari hesap kodu koşullu)

```
sub ProcessEvent(App)
CariKod = App.Doc.FieldAsString("ARP_CODE")
if CariKod = "" Then
rOk=App.Doc.FirstRoot
while rOk
okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
while okay
ASIN = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.DESRIPTION")
App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.MASTER_CODE", ASIN
okay=App.Doc.NextChild
wend
rOk=App.Doc.NextRoot
wend
App.Doc.SaveData ' Yapılan değişiklikler kaydediliyor
End if
```





end sub

30 - Perakende satış faturasında kdvli birim fiyatı satıra getirme

```
sub ProcessEvent(App)
    ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
    while ok1
        name = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.PRICE")
        App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.ITEM_SUPPLIERCODE", name
    ok1=App.Doc.NextChild
    wend
    App.Doc.SaveData
end sub
```

31 - İrsaliye numarası ve tarihinin sipariş numarası ve tarihine getirilmesi

```
sub ProcessEvent(App)
    ARPCODE=App.Doc.FieldAsString("ARP_CODE")
    If ARPCODE = "XXX" then
        rOk=App.Doc.FirstRoot
        while rOk
            MusteriSiparisNo = App.Doc.FieldAsString("DISPATCHES.DISPATCH_NUMBER")
            IrsaliyeTarihi=App.Doc.FieldAsString("DISPATCHES.DISPATCH.DATE")
            ok1=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
            while ok1
                App.Doc.ModifyData "CUSTORDNO", MusteriSiparisNo
                App.Doc.ModifyData "ORDER_FICHEDE",
                    IrsaliyeTarihi ok1=App.Doc.NextChild
            wend
            rOk=App.Doc.NextRoot
        wend
        App.Doc.SaveData
    end If
end sub
```

32 - Doküman izleme numarasının "BAYINO" alanına getirilmesi (cari hesap kodu koşullu)

```
sub ProcessEvent(App)
    ARPCODE=App.Doc.FieldAsString("ARP_CODE")
    If ARPCODE = "XXXX" then
        DETAY = App.Doc.FieldAsString("DOC_TRACK_NR")
    end If
end sub
```





```
App.Doc.ModifyData "SHIPLOC_CODE",DETAY
App.Doc.SaveData
End if
end sub
```

33 - Malzeme özel kod1 değerinin satır açıklamasına getirilmesi (ihracat fatura koşullu)

```
sub ProcessEvent(App)
Type1 = App.Doc.FieldAsString("EXIM_FICHE_TYPE")
If Type1 = 2 or 3 Then
    strDBUser = "xx"
    strDBPass = "xxxxx"
    strDBServer = "xxxxxx"
    strDBName = "xxxxxx"

    Set conn = CreateObject("ADODB.Connection")
    conn.ConnectionString = "Provider=SQLOLEDB.1;Persist Security Info=False;User ID=" + strDBUser
+ ";Password=" + strDBPass + ";Initial Catalog=" + strDBName+ ";Data Source=" +
    strDBServer conn.Open
    Set rs = CreateObject("adodb.recordset")
    rs.ActiveConnection = conn

    okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
    while okay
    Type2 = App.Doc.FieldAsInteger("TRANSACTIONS.TRANSACTION.TYPE")
    If Type2 = 0 then
        intlItemRef = App.Doc.FieldAsInteger("TRANSACTIONS.TRANSACTION.MASTER_ORGLOGICALREF")
        strItemRef = CStr(intlItemRef)
        strQuery = "SELECT DEFINITION_ FROM LG_001_SPECODES WHERE"
        strQuery = strQuery + "(((CODETYPE = 1) AND (SPECODETYPE = 1)) OR ((CODETYPE = 1)
AND (SPECODETYPE > 1)) OR ((CODETYPE > 1)))"
        strQuery = strQuery + "AND (((CODETYPE = 1) AND (SPECODETYPE <= 1)) OR ((CODETYPE <
1)))" strQuery = strQuery + " AND SPECODE IN (SELECT SPECODE FROM LG_001_ITEMS WHERE
LOGICALREF =" + "" + strItemRef + "" + ")"
        rs.Open strQuery, conn
        strPro = rs.Fields(0).value
        App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.DESCRPTION",
            strPro do while (rs.EOF = false)
            rs.MoveNext()
        loop
    end if
end while
end sub
```





```
rs.close
App.Doc.SaveData
end if
okay=App.Doc.NextChild
wend
end if
conn.close
end sub
```

34 - Satıcı kod ve satıcı adının genel açıklamalara getirilmesi

```
sub ProcessEvent(App)
    DETAY = App.Doc.FieldAsString("SALESMAN_CODE")
    DETAY1 = App.Doc.FieldAsString("SALESMAN_DEF")
    App.Doc.ModifyData "NOTES5", "Satıcı Kodu:" + DETAY
    App.Doc.ModifyData "NOTES6", "Satıcı Adı:" + DETAY1
    App.Doc.SaveData
end sub
```

35 - Malzeme barkodunun satıra getirilmesi

```
sub ProcessEvent(App)
    rOk=App.Doc.FirstRoot
    while rOk
        okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
        while okay
            OZELKOD = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.BARCODE")
            App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.PROCEDUCER_CODE", OZELKOD
            okay=App.Doc.NextChild
        wend
        rOk=App.Doc.NextRoot
    wend
    App.Doc.SaveData ' Yapılan değişiklikler kaydediliyor
end sub
```





36 - E-arşiv faturada malzeme barkodunun satıra getirilmesi

```
sub ProcessEvent(App)
    strDBUser = " XXXX "
    strDBPass = " XXXX "
    strDBServer = "XXXXX"
    strDBName = "XXXX"
    Set conn = CreateObject("ADODB.Connection")
    conn.ConnectionString = "Provider=SQLOLEDB.1;Persist Security Info=False;User ID=" +
strDBUser + ";Password=" + strDBPass + ";Initial Catalog=" + strDBName+ ";Data Source="
+ strDBServer
    conn.Open

    Set rs = CreateObject("adodb.recordset")
    rs.ActiveConnection = conn

    okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION") ' Faturanın ilk satırına konumlanılıyor.

    while okay
        sType = App.Doc.FieldAsInteger("TRANSACTIONS.TRANSACTION.TYPE") ' Satır tipi okunuyor

        StrOrflineRef =
App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_CODE")
        strQuery = "SELECT BRK.BARCODE FROM LG_001_UNITBARCODE BRK " + _
"LEFT OUTER JOIN LG_001_ITMUNITA ITA ON ITA.LOGICALREF=BRK.ITMUNITAREF " + _
"LEFT OUTER JOIN LG_001_ITEMS ITM ON ITM.LOGICALREF=ITA.ITEMREF " + _
"LEFT OUTER JOIN LG_001_01_STLINE STL ON STL.STOCKREF=ITM.LOGICALREF " + _
"LEFT OUTER JOIN LG_001_01_INVOICE INV ON STL.INVOICEREF=INV.LOGICALREF " + _
"WHERE ITM.CODE= '" + StrOrflineRef + "'"

        rs.Open strQuery, conn
        do while (rs.EOF = false)
            StrOrflineNo =rs.Fields(0).value
            rs.MoveNext()
        loop
        rs.close
        StrDescription = StrOrflineNo
        App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.PROCEDUCER_CODE", StrDescription

        okay=App.Doc.NextChild
    wend
    conn.close
    App.Doc.SaveData
end sub
```





37 - GTIP kodunun satıra getirilmesi

```
sub ProcessEvent(App)
  rOk=App.Doc.FirstRoot
  while rOk
    okay=App.Doc.FirstChild("TRANSACTIONS.TRANSACTION")
  while okay
    GTIP = App.Doc.FieldAsString("TRANSACTIONS.TRANSACTION.MASTER_GTIPCODE")
    App.Doc.ModifyData "TRANSACTIONS.TRANSACTION.ITEM_SUPPLIERCODE", GTIP
    okay=App.Doc.NextChild
  wend
  rOk=App.Doc.NextRoot
  wend
  App.Doc.SaveData ' Yapılan değişiklikler kaydediliyor
end sub
```

38 - UQUERY_GETFLD Cari Özel Kod 5 Alanının Cari Açıklama Alanına Yazılması

```
sub ProcessEvent(App)
  CICode=App.Doc.FieldAsString("ARP_CODE")
  If CICode ="AS-001" Then
    rOk=App.Doc.FirstRoot
    While rOk
      CIOzelCode5 = App.UQUERY_GETFLD(15,"SPECODE5", 16, 52,
FALSE,"CODE='"+CICode+"'")
      App.Doc.ModifyData "ARP_DEF", CIOzelCode5
      rOk=App.Doc.NextRoot
    Wend
  end if
  App.Doc.SaveData
end sub
```

39 - UQUERY_GETFLD Sevkiyat Adresi Adres Bilgilerinin Ek Alana Eklenmesi





```
sub ProcessEvent(App)
    strShipLoc_Code = App.Doc.FieldAsString("SHIPLOC_CODE")

    sevkiyatLine1 = App.UQUERY_GETFLD(58, "ADDR1", 200, 52, FALSE, "CODE
= '"+strShipLoc_Code+"'")

    App.Doc.WriteStartElement "ADDITIONALDOCUMENTS/ADDITIONALDOCUMENT"
    App.Doc.WriteString "ID", sevkiyatLine1

    App.Doc.WriteString "ISSUEDATE", Year(Date()) & "-" & Right("0" & Month(Date), 2) & "-"
& Day(Date())

    App.Doc.WriteString "DOCUMENTTYPE", "ADRES"
    App.Doc.WriteEndElement
    App.Doc.SaveData
end sub
```

40 - UQUERY_GETFLD Metodu

UQuery_GetFld(portNr, fldName, fldSz, fldType, isDate, whrCls)

portNr: Her tablo için numara listesi şöyle;

```
1 ; { Stok kartlari portu      }
2 ; { Hizmet kartlari portu    }
3 ; { Hizmet toplamlari portu  }
53; { Hizmet ambar rakamlar}
5 ; { Ambar toplamlari portu   }
57; { Stok ambar toplamlari    }
59; { Genel Stok ambar toplamlari }
7 ; { Karma koli satirlari     }
8 ; { Stok fi'Yleri portu      }
9 ; { Stok hareket portu      }
10; { Promosyon karti portu    }
11; { Indirim masraf portu     }
12; { Siparis fisi portu       }
13; { Siparis satiri portu     }
14; { Fatura Portu            }
42; { Mustahsil makbuzu portu  }
15; { Cari Hesap kart portu    }
58; { Sevkiyat Bilgileri      }
56; { Cari hesap fisleri portu }
16; { Cari Hesap hareket portu }
17; { Cari Hesap istihbarat portu }
```





- 18; { Cari Hesap risk rakamlari portu }
- 19; { Cari Hesap toplamlar portu }
- 60; { Cari Hesap genel topl. portu }
- 20; { Odeme Plani portu }
- 21; { Odeme Plani satiri portu }
- 22; { Odeme hareketi portu }
- 23; { Banka karti portu }
- 24; { Banka hesap portu }
- 54; { Banka fiYleri portu }
- 25; { Banka hareket portu }
- 26; { Banka toplamlar portu }
- 61; { Banka genel toplamlar portu }
- 27; { Muhasebe kart portu }
- 28; { Muhasebe fis portu }
- 29; { Muhasebe hareket portu }
- 30; { Muhasebe toplamlar portu }
- 32; { Masraf Merkezi portu }
- 33; { Kasa kart portu }
- 34; { Kasa hareket portu }
- 41; { Kasa toplamlari portu }
- 45; { Calisma donemine bagli bilgiler }
- 62; { Kasa genel toplamlari portu }
- 35; { Cek / senet portu }
- 36; { Cek / senet bordro portu }
- 37; { Cek / senet hareket portu }
- 38; { Fiyat Listesi portu }
- 39; { Entegrasyon baglanti kodlari }
- 40; { Ozel kod tanimlari }
- 43; { Alis-Satis kosullari portu }
- 44; { Bilanco tablolar sat r portu }
- 46; { Log portu }
- 47; { Sabit kymet kaytlar }
- 48; { Sabit kymet ylbilgileri }
- 163; { SK yillara yaygin maliyet ve ROFM bilgileri }
- 49; { Sistem Log portu }
- 50; { Cost PermFile Portu }
- 55; { Kart muhasebe referans portu }
- 70; { Satıs elemanları portu }
- 72; { Rota tanımları portu }
- 73; { Rota sıraları portu }
- 74; { Satıcı hedef tanımları portu }
- 75; { Watermark doküman folder portu }
- 76; { Endüstri/sektörler }
- 77; { Müşteri-satıcı }
- 78; { Müşteri-sektör atama }
- 71; { Müşteri-satıcı atama }
- 79; { Müşteri CH atama }
- 80; { Müşterideki ilgili kişiler }
- 81; { Satıs fırsatları }
- 82; { Satıs faaliyetleri }
- 83; { Teklifler }
- 84; { Teklif alternatifleri }





- 85; { Teklif satırları }
- 86; { Satıcı-İlgili kişi atama }
- 87; { Aktivite-İlgili kişiler }
- 88; { Aktivite-Dosyalar }
- 91; { Satıcı-C.Hesap bağlantısı portu }
- 89; { Faaliyetlerim Ağacı Maddeleri }
- 90; { Faal. Ağacı Maddl. Filtreleri }
- 92; { İhracat kredi kartları portu }
- 93; { İhracat kredi satırları portu }
- 253; { Satış fırsatları tarihçe portu }
- 254; { İlgili kişiler özel günler portu }

- 100; { MPS veya MRP Başlık Kaydı }
- 101; { MPS veya MRP Satır Kaydı }
- 102; { MPS veya MRP Bağlantı Kaydı }
- 103; { MPS veya MRP Öneri Kaydı }
- 249; { MRP-Item bağlantı Kaydı }
- 252; { MRP-Item bağlantı Kaydı }

- 106; { Çalışan tanımları }
- 105; { Çalışan grupları }
- 107; { Çalışan ve grubu bağlantısı }

- 51; { Seri / Lot hareketi }
- 110; { Ambar yerleşimleri }
- 111; { Seri / Lot Numaraları }
- 112; { Malzeme - ambar parametreleri }
- 113; { Malzeme Karakteristik kodları }
- 114; { Malzeme Karakteristik değerleri }
- 115; { Malzeme - sınıf ilişkisi }
- 117; { Malzeme tedarikçiler }
- 118; { Malzeme - Üretim reçetesi ilişkisi }
- 119; { Alternatif malzemeler }
- 123; { Malzeme Karakteristik Atama }
- 125; { Malzeme karakteristikleri seçim listesi }
- 127; { Malzeme Sınıfı Dağıtım Şablonu }
- 128; { Dağıtım Şablonu satırı }

- 130; { Birim Setleri Tanımları }
- 131; { Birim Setleri Satırları }
- 132; { Malzeme Birim Set Satırı Atamaları }
- 133; { Birim Seti Çevrim Bilgileri }
- 134; { Hizmet Birim Set Satırı Atamaları }
- 135; { Kalite Kontrol Karakteristik setleri }
- 136; { Kalite Kontrol Karakteristik seti satırları }
- 137; { Kalite Kontrol Karakteristik seti satır değerleri }
- 138; { Kalite Kontrol Karakteristik set atamaları }

- 139; { Seri lot kalite kontrol değeri girişi portu }
- 140; { Dönem kapama maliyet kayıtları }

- 141; { Trigger&sp yazımında gerekli olan alanlar }





- 143; { Operasyon kartı }
- 145; { Operasyon İhtiyaçları }
- 146; { Çalışan İhtiyaçları }
- 147; { Araç İhtiyaçları }
- 161; { Operasyona iş istasyonu özellik bağlantısı }
- 189; { Malzeme - fabrika parametreleri }
- 192; { Önceki operasyon atama }
- 193; { Üretim Rotaları }
- 197; { Üretim Rota Satırı }

- 176; { Malzeme, iş istasyonu toplamaları }
- 177; { Malzeme, iş istasyonu tanımları }

- 182; { İş istasyonları }
- 184; { İş istasyon grupları }
- 185; { İş istasyonu ve grubu bağlantısı }

- 198; { İş istasyonu özellik tanımları }
- 199; { İş istasyonu özellik değerleri }
- 194; { İş istasyonu özellik atama }
- 195; { İş istasyonu özellik değeri atama }

- 108; { Firma üretim emri parametreleri }
- 109; { Reçete Parametreleri }
- 155; { Parametre değerleri }
- 167; { Ortalama Döviz Kurları }
- 183; { Standart "Malzeme/İş istasyonu/Çalışan" Maliyetleri }
- 186; { Standart Maliyet Periyodları }
- 187; { Standart Reçete Maliyetleri }
- 188; { Üretim Reçeteleri }
- 190; { Üretim Reçeteleri Revizyon }
- 191; { Üretim Reçeteleri Satırları }
- 175; { Yan ürün BOM bağlantısı }
- 144; { BOM satırı operasyon ilişkisi }

- 150; { Standart Maliyetler }

- 156; { Mühendislik değişikliği satırı }

- 157; { Üretim Emri }
- 154; { Üretim emri satırı }
- 159; { İş emri satırı }
- 160; { İş emri sıralaması }
- 158; { İşlem bağlantıları }
- 165; { Kaynak işgal kaydı }
- 149; { Malzeme temin kaydı }

- 196; { ÜG Fişi WS-Acc referans portu }
- 168; { Tanimli Alanlar Deger Portu (Kartlar için)}
- 169; { Tanimli Alanlar Deger Portu (Fişler için)}
- 171; { İş emri durma nedeni }
- 172; { Durma nedeni - İş istasyonu ataması }





- 173; { İş emri durma işlemi }
- 200; { Dağıtım Araçları }
- 201; { Dağıtım Rotaları }
- 202; { Dağıtım Rota Satırları }
- 203; { Dağıtım Emirleri }
- 204; { Dağıtım Emirleri Satırları }
- 205; { Kampanya Kartları }
- 206; { Kampanya satırları }
- 207; { Genel Gider Kartları }
- 208; { Genel Gider Muhasebe Hesapları }
- 209; { Maliyet Yerleri }
- 210; { Maliyet Yerleri satırları }
- 211; { Maliyet Yerleri Hizmet Dağıtımı }
- 212; { İş istasyonları Maliyet Yerleri Atamaları }
- 213; { Operasyon ihtiyaç Aktiviteleri }
- 214; { Fatura İstihbarat portu }
- 215; { Genel Gider Hareketleri }
- 216; { Yansıtma şablonu }
- 217; { Şablon satırları }
- 218; { Filleri Genel Gider Yükleme }
- 221; { Gerçek Aktivite Miktarları }
- 222; { Reflect Assign Table }
- 223; { üretim emri fark muhasebe kodları }
- 224; { Stok Satırları Enflasyon Katsayıları }
- 225; { Muhasebe Satırları Enflasyon Katsayıları }
- 226; { Talep Fişi }
- 227; { Talep Fişi Satırları }
- 229; { Projeler Portu }
- 230; { Talep Fişi Karşılama Bağlantı Kaydı }
- 231; { Muhasebe Hesap Kartı Dağıtım Şablonu }
- 232; { Muhasebe Hesap Kartı Dağıtım Şablonu Satırları }
- 233; { Muhasebe Hareketleri Detay Satırları }
- 234; { Hızlı Üretim Fişi }
- 235; { Muhasebe Kartları Karşı Hesaplar }
- 236; { Paketleme Fişleri }
- 237; { Paketleme Hareketleri }
- 238; { Paket Atama }
- 239; { Kredi kartı geri ödeme planı }
- 240; { Kasa Muhasebe Hareketleri Detay Satırları }
- 241; { İndirimli Ödeme Satırları }
- 242; { İndirimli Ödeme Hareketleri }
- 243; { Muhasebe Bağlantıları }
- 244; { Sipariş Bağlantı Portu }
- 245; { İftar Tarihçesi }
- 246; { Geri Ödeme Planı (Yeni) }
- 247; { Geri Ödeme Plan Satırları }
- 248; { Logo Data Exchange History }
- 329; { Logo Data Exchange }
- 250; { Export - import Type } 5/2004
- 251; { Marka Kartları Portu }





- 255; { Bilanco tabloları başlık portu }
- 95; { DIIB portu }
- 96; { DIIB satırları portu }
- 97; { DIIB reçete bilgileri portu }
- 256; { İthalat // ihracat dosyası GD,Exim }
- 257; { İthalat / Malzeme Dolasım Fisi/Millilestirme Fisi }
- 258; { İthalat / Malzeme Dolasım Fisi/Millilestirme Lines }
- 259; { İthalatta Kullanılacak Hizmet Kartları İlişkileri }
- 260; { İthalat // ihracat dosyası satırları, Her stLine için bir satır }
- 261; { İthalat / Dağıtım Fişleri }
- 262; { İthalat / Dağıtım Fiş Satırları }
- 263; { İthalat / Dağıtım Fiş Satırları Bağlantı Kayıtları }
- 264; { İthalat/İhracat Tarihçesi }
- 265; { Almanya Mevzuatı Genel Amaçlı Tanımlar }
- 266; { Almanya Mevzuatı İrsaliye Ek Bilgileri }
- 267; { Taksit Kartı }
- 268; { Fatura ve siparis fisi Kefil Bilgileri }
- 273; { Gösterge parametreleri portu }
- 274; { Erişim ağacı parametreleri portu }
- 277; { SMM mahsup fişi ve Tecil Terkin Mahsup Fişi- fatura bağlantıları portu }
- 120; { Vardiya tanımları }
- 121; { Vardiya saatleri }
- 122; { Vardiya atamaları }
- 104; { İstisnai durum atamaları }
- 178; { Çalışılmayan günler }
- 162; { Ürün hattı }
- 170; { Tic.Sistem Par. Ambarlar }
- 219; { Özel Tüketim Vergisi Kartı }
- 220; { Özel Tüketim Vergisi Satırları }
- 181; { İş istasyonu parçaları }
- 174; {MBS Crm kart bağlantıları }
- 180; {MBS Crm hareket bağlantıları }
- 278; { Analitik Bütçe Dönemleri }
- 279; { Analitik Bütçe }
- 280; { Analitik Bütçe Satırları }
- 281; { Analitik Bütçe Periyod Satırları }
- 282; { Analitik Bütçe Tahsis Fişleri }
- 283; { Analitik Bütçe Tahsis Fişi Satırları }
- 284; { Analitik Bütçe Tahsis Fişi Periyod Satırları }
- 285; { Muhasebe Talep Fişleri }
- 286; { Muhasebe Talep Fiş Satırları }
- 287; { Muhasebe Talep Fiş detay satırları }
- 288; { Analitik Bütçe Revizyon Fişleri }
- 289; { Analitik Bütçe Revizyon Fişi Satırları }
- 290; { Analitik Bütçe Revizyon Fişi Periyod Satırları }
- 291; { Geçici Kabul Bağlantıları }
- 293; { E-Beyanname Kayıtları }
- 294; { E-Beyanname Satırları }
- 295; { Maliyet Dağıtım Fişleri }
- 296; { Maliyet Dağıtım Fiş Satırları }
- 297; { Maliyet Dağıtım Fiş Satırları Bağlantı Kayıtları }
- 298; { Cari Hesap Teminat Mektubu Risk Bilgileri de }
- 299; Teminat kartları





300; { Teminat bordroları }
301; { Teminat bordro hareketleri }
302; { Satınalma teklif yönetimi Fişleri }
303; { Satınalma teklif yönetimi Fiş satırları }
304; { Generic Module Port }

305; { Malzeme Özellik Setleri }
306; { Malzeme Özellik Setleri - Özellik Bağlantıları }
307; { Malzeme Varyantları }
308; { Malzeme Varyant - Özellik Bağlantıları }
309; { Varyant Ambar toplamları portu }
310; { Varyant ambar toplamları } 311; { Genel Varyant ambar toplamları }
312; { Üretim Reçete Formül/Koşul Satırları }
313; { Varyant İstisna ve Fiyat Tanımları }
314; { Otomatik malzeme/varyant kart oluşturma şablon portu }

315; { Malzeme giriş/çıkış hareketi ilişkisi portu }

316; { Ürün Katalog Tanımları }
317; { Ürün Katalog Satırları }
318; { İş emri operasyon tamamlama bağlantı satır }
267; { personel tanımları portu }
330; { Banka Kredileri Kartları }
331; { Banka Kredi Taksitleri }
332; { Birim-barkod bağlantı tablosu }
333; { Malzeme market bilgileri tablosu }
334; { Malzeme market bilgileri tablosu }
335; { Malzeme market bilgileri tablosu }
336; { İşyeri bazında onaylama bilgileri }
347; { İşyeri bazında yevmiye madde numaralama }
348; { Teminat komisyonu ödemeleri }
349; { Sabit kıymet işlemleri }
350; { Garanti Kayıtları Portu }
351; { Bakım Kayıtları Portu }
353; { Muhasebe Kodları üzerinden dağıtım şablonu detayları portu }
354; { Fiyat Kartları İşyeri Bilgisi Portu }
355; //Satınalma Siparişi Şablon Kartı
356; //Satınalma Siparişi Şablon Satırı

fldName : query'den dönecek olan alanın ADI

fldSz : query'den dönecek olan alanın BOYUTU

fldType : query'den dönecek olan alanın TİPİ.

BYTE= 1;

SMALLINT = 2;

INT = 3;

REAL = 5;

DOUBLE = 6;

EXTENDED = 7;





STRING = 51;
CSTRING = 52;
WSTRING = 53;

isDate : Dönecek olan alan TARİH tipinde mi ?

whrCls : Where koşulu

41 - Operatörler

Kuralın çalışacağı koşulları tanımlanırken bir takım alan ve bilgileri karşılaştırıp, kıyaslayıp, kural tanımının bu karşılaştırmaların sonuçlarına göre işlemesi sağlanabilir. Bu durum için operatörlerden yararlanılır.

- **Matematiksel Operatörler** : Alanların değerleri üzerinde çalışırken kullanılabilir.

(+) Toplama


(-) Çıkarma

(*) Çarpma


(/) Bölme

- **Karşılaştırma Operatörleri** : Karşılaştırma operatörleri değişkenin belirlediğimiz kriterle arasındaki ilişkiyi belirlerken kullanılabilir. Eşit Operatörü : Belirlenen alanın değeri ile belirlenen kriterin eşit olma durumu. Eşit Değil Operatörü : Belirlenen alanın değeri ile belirlenen kriterin eşit olmama durumu.


 Büyük Operatörü : Belirlenen alanın değeri ile belirlenen kriterin büyük olma durumu.

 Büyük Eşit Operatörü : Belirlenen alanın değeri ile belirlenen kriterin büyük eşit olma durumu.

 Küçük Operatörü : Belirlenen alanın değeri ile belirlenen kriterin küçük olma durumu.

 Küçük Eşit Operatörü : Belirlenen alanın değeri ile belirlenen kriterin küçük eşit olma durumu.

- **Mantıksal Operatörler** : Bazı durumlarda yukarıda açıklanan karşılaştırma operatörleri ile tanımlanan koşullar yeterli olmayabilir. Birden fazla şart'ı bir arada kullanmak için mantıksal operatörlere ihtiyaç duyulabilir.


 Ve Operatörü : Birden çok şart'ın doğru olmasını isteyen bir operatördür. Birleştirdiği her bir karşılaştırma şartının doğru olmasını gerekir.

ÖRN : DOĞRU AND DOĞRU = DOĞRU






YANLIŞ AND DOĞRU = YANLIŞ


 VEYA Operatörü : Birden çok şart'ın içerisinde sadece bir şart'ın doğru olmasının yeterli olduğu operatördür. Birleştirdiği tüm karşılaştırma şartlarından sadece birisi doğru olması yeterlidir.

ÖRN : DOĞRU OR YANLIŞ = DOĞRU

YANLIŞ OR YANLIŞ = YANLIŞ



 DEĞİL Operatörü : Belirlenen sonucun tersine işlem yapar.

ÖRN : NOT DOĞRU = YANLIŞ

 EXCLUSIVE OR (Özel VEYA) Operatörü : Koşullardan sadece seçilen bir tanesinin sağlanması durumunda kullanılır. Sonucunun olumlu olabilmesi için bileşenlerden sadece bir tanesinin olumlu olması gerekir, her iki birleşende olumlu ise sonuç olumsuz olur.

ÖRN : DOĞRU XOR DOĞRU = YANLIŞ

YANLIŞ XOR DOĞRU = DOĞRU

 ,  ve Yardımcı Operatörler : Mantıksal operatörler kullandığımızda mutlaka ifadeler parantez ile düzenlenmelidir.

- **Koşul Operatörleri** : Kuralın belirlenen koşulun sağlanması halinde çalışmasını sağlayan operatörlerdir. IFTHEN ifadesi : Bu operatör ile tanımlanmış bir şart ifadesinin sonucunun doğru ya da yanlış olmasına göre kuralın nasıl çalışacağı belirlenebilir. ÖRN : IFTHEN(Koşul? Doğruysa;Yanlışsa) : Koşul doğruysa 1. yanlışsa 2. değeri verir.

NUMBER = IFTHEN(LEN(NUMBER)=7?'X'+NUMBER;'Y'+NUMBER)

Yukarıdaki örnekte kural tanımında seçilmiş olan ilgili XML şablonunda bulunan fiş numarası 7 karakter ise başına 'X' koymakta, 7 karakterden küçük veya büyük ise fiş numarasının başına 'Y' karakteri eklemektedir.

- SWITCH ifadesi : IFTHEN ifadesinden farklı birden fazla değer tanımlanarak kuralın nasıl çalışacağı belirlenebilir.

ÖRN : SWITCH(ifade?Değer1=Sonuç1;DeğerN=SonuçN;...) : ifade sonucu değer1 ise Sonuç1, ifade sonucu değerN ise SonuçN verir.

NUMBER = SWITCH(DOC_NUMBER?'01'='A';'02'='B';'03'='D')

Yukarıdaki örnekte fiş numarasına fişin doküman izleme numarasındaki değer '01' ise 'A', '02' ise 'B', '03' ise 'D' değerleri atanmaktadır.

- **Metin Operatörleri** : Metin alanlar içinden istenilen bilginin alınması veya işlenmesinde kullanılır. Kural tanımlarında metin tipindeki ifadeler 'XYZ' de olduğu gibi tek tırnak içerisinde gösterilirler.

- MID Operatörü : Bir metin alanın istenilen karakterinden başlayıp istenilen kadarını almak için kullanılır.

ÖRN : MID(String,Başlangıç,Bitiş) : Verilen String'in Başlangıç'tan itibaren Bitiş'e kadar olan karakterlerini verir.

MID(CODE,1,5)='KALEM'





Yukarıdaki koşul tanımında ilgili XML şablonundaki CODE alanının taşıdığı metin bilginin ilk beş karakteri alınır ve "KALEM" ifadesi ile karşılaştırılır. Eğer ilk beş karakter "KALEM" ise koşul olumlu, değil ise koşul olumsuz sonuç üretir.

- LEN Operatörü : Bir metin ifadenin karakter sayısını öğrenmek için kullanılır.

ÖRN : LEN(String) : Parametre olarak aldığı metin ifadenin kaç karakter olduğunu sonuç olarak döner.

NUMBER = MID(DOC_NUMBER,1,LEN(DOC_NUMBER)-1)

Yukarıdaki kural tanımında seçilen XML şablonundaki doküman numarası tag'inde (DOC_NUMBER) taşınan bilginin son karakteri hariç fiş numarası (NUMBER) alanına atanır.

- % Operatörü : Bir metin ifadesinin bulunduğu koşullarda kullanıldığında sonraki karakterler ne olursa olsun anlamını taşır.

ÖRN : NUMBER ='A%' : İlgili fişin numarası 'A' ile başlıyorsa sonuç olumlu döner.

- **Özel Operatörler** : Farklı amaçlar tanımlanmış operatörlerdir.

- TODAY Operatörü : Üzerinde çalışılan terminal'in sistem tarihinden okuyarak günün tarih'ini metin olarak getirir. Oluşturulan metnin işletim sisteminin bölgesel ayarlarından etkilendiği unutulmamalıdır.

