

Profile

A munkamódszerek listáját tartalmazza, illetve az alapértelmezett képernyőt! Tipikusan egy van belőle, és az XML csomag Profile könyvtárja alatt laknak. Példa:

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<Profile id="Proba">
  <DefaultView>MMFolyamatok/ScreenUgyletekEsemenyek</DefaultView>
  <MMk>
    <MM id="MMFolyamatok" icon="IconMMFolyamatok" color="56,86,131">Tudásbázis</MM>
    <MM id="MMTorzsadatok" icon="IconMMTorzsadatok" color="56,86,131">Törzsadatok</MM>
  </MMk>
</Profile>
```

Tagok

```
<Profile id="Proba">...</Profile>
```

Az id a fájlnev a Profile nélkül, azaz ennek a fájlnek ProfileProba.xml a neve.

```
<DefaultView>MMFolyamatok/ScreenUgyletekEsemenyek</DefaultView>
```

Ebben a tag-ben tároljuk el a kezdő nézetet. InnerText szintaxis: MM_xmlnév/Screen_xmlnév. Ez a tag dinamikusan kiértékelhető, ilyenkor is MM/Screen formában kell visszaadni az eredményt. Pl.:

```
<DefaultView type="SQL" default="MMPEP/ScreenPEP" return="string">
  SELECT CASE WHEN [##Session.UserID##] = 4
    THEN 'MMTorzsadatok/ScreenUsers'
    ELSE 'MMPEP/ScreenPEP'
  END
</DefaultView>
```

```
<MM id="MMFolyamatok" icon="IconMMFolyamatok" color="56,86,131">Tudásbázis</MM>
```

Munkamódszert kapcsol a Profile-hoz. Az id a munkamódszert leíró xml fájlneve kiterjesztés nélkül (ld. MM.doc). A tagok közé munkamódszer megjeleníteni kívánt neve kerül. Az icon és a color attribútumok már nem használatosak, de benne kell lenniük az xmlben.

```
<QuickLaunch>
  <Item>
    <Name>Jelszó módosítás</Name>
    <MM>MMTorzsadatok</MM>
    <Screen>ScreenNewUser</Screen>
    <Filters>
      <Filter>
        <Name>ID</Name>
        <Type>Out</Type>
        <Value type="SQL" return="int" default="0"><[[CDATA[SELECT TOP 1 ID FROM [User] WITH (NOLOCK) WHERE UserID = [##Session.UserID##] AND Deleted = 0 ORDER BY ID DESC]]></Value>
      </Filter>
    </Filters>
  </Item>
</QuickLaunch>
```

Ezzel lehet menüpontokat deiniálni a lenyíló listába. a Name lesz a menüpont felirata, az MM és Screen meghatározzák a betölteni kívánt képernyő nevét, illetve a Filters-ben kell definiálni azokat a filtereket és értékeiket, amelyek kellene a képernyő megnyitásához.

```
<MenuType>top</MenuType>
```

Desktop nézetben ezzel lehet állítani, hogy felső vagy oldalsó menü jelenjen meg.

```
<ExitButtonsVisible>>false</ExitButtonsVisible>
```

A side menüben a kilépés opció látható-e (default true)

```
<QuickLaunchBlockCollapsed>>false</QuickLaunchBlockCollapsed>
```

A QuickLaunch blokk side menüben automatikusan legyen összezárvá vagy sem (default true)

Példa

```
<?xml version="1.0" encoding="iso-8859-2"?>
<Profile id="Default">
  <DefaultView type="Constant" return="string" default="">MMIDE/ScreenIDE</DefaultView>
  <MMk>
    <MM id="MMIDE" icon="IconMMErtesites" color="56,86,131">IDE</MM>
    <MM id="MMGVH" icon="colors" color="56,86,131">GVH</MM>
    <MM id="MMCRT01" icon="colors" color="56,86,131">CRT01</MM>
  </MMk>
```

```
<SkinSwitchingIsEnabled>>false</SkinSwitchingIsEnabled>
<ESCIsEnabled>>false</ESCIsEnabled>
```

```
<QuickLaunch>
  <Item>
    <Name>Jelszó megváltoztatása</Name>
    <MM>MMTorzsadatok</MM>
    <Screen>ScreenChangePassword</Screen>
    <Filters>
      <Filter>
        <Name>ID</Name>
        <Type>Out</Type>
        <Value type="SQL" return="int" default="0">
          <[[CDATA[
            SELECT TOP 1 ID FROM [User] WITH (NOLOCK)
              WHERE UserID = [##Session.UserID##] AND Deleted = 0 ORDER BY ID DESC
          ]]]>
        </Value>
      </Filter>
    </Filters>
  </Item>
</QuickLaunch>
```

```
</Profile>
```

Node-ok

Tartalomjegyzék

Munkamódszer MM

A fájlok az XML csomagon belül egy MM nevű könyvtárban laknak. A fájlnev mindig MM-el kell, hogy kezdődjön, és a fő tag id attribútuma mindig a fájlnev a kiterjesztés nélkül. Minden munkamódszer minimum egy Screen-t (képernyőt) tartalmaz. A felületen a menü első sora a munkamódszerek, a második sora a munkamódszerhez tartozó képernyők (Screen).

Példa:

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<MM id="MMPEP">
  <Screen>
    <schema>ScreenPEP</schema>
    <icon>IconScreenEvent</icon>
  </Screen>
  <Rules>
    <Visible type="SQL" return="Boolean" default="false">
      SELECT CASE WHEN COUNT(*) > 0
      THEN 'True'
      ELSE 'False'
      END
      FROM PeopleGroupsPeople as pgp with (readpast)
      WHERE pgp.PeopleID = [##Session.UserID##]
      AND pgp.GroupID IN (50000, 50011, 3)
    </Visible>
  </Rules>
</MM>

<schema>...</schema>
```

A schema annak a resource állományban levő XML állománynak a neve, ami az adott Screen leírását tartalmazza (a kiterjesztés elhagyásával). Az <icon>...</icon> tag nem használt, de kötelező elemek, értékük jelenleg bármi lehet. Lehetőség van láthatósági szabály megadására a Rules node-ban. A Rules tag-en belül csak egy Visible RuleValueType lehet (Isd.: RuleValueType.doc)

Node-ok

Tartalomjegyzék

Screen

Az egy képernyőn található View-kat, riportokat, adatbeviteli képernyőket és egyéb nézeteket Screen-nek nevezzük. Egy képernyő egy vagy több komponensből (vagy megjelenítőtől) áll (Isd Component.doc). Egy képernyőt lehet vízszintesen és függőlegesen bontani.

Példák

Első példa:

```
<?xml version="1.0" encoding="iso-8859-2"?>
<Form id="ScreenEvent" name="Események részletekkel" MaxWidth="95%" MaxHeight="100%">
  <CssClass>EditFormProject</CssClass>
  <SplitContainers>
    <Split id="1" orientation="Vertical" splitterDistance="50%" splitterDistanceOnTablet="315">
      <Panel1 type="Component" id="1">ComponentEventEditForm</Panel1>
      <Panel2 type="Component" id="2">ComponentEvent03Hoz</Panel2>
    </Split>
  </SplitContainers>

  <Components>
    <Component id="1" isreversible="false" isswitchable="false">2</Component>
  </Components>

  <ActiveConnections>
    <ActiveConnection id="1">2</ActiveConnection>
  </ActiveConnections>
</Form>
```

Második példa:

```
<?xml version="1.0" encoding="iso-8859-2"?>
<Form id="ScreenNewUser" name="Felhasználók" MaxWidth="85%" MaxHeight="80%">
  <Panel type="Component" id="1">ComponentUserEdit</Panel>

  <Rules>
    <Visible type="SQL" return="Boolean" default="false">
      SELECT CASE WHEN COUNT(*) > 0 THEN 'True' ELSE 'False' END
      FROM PeopleGroupsPeople as pgp with (readpast)
      WHERE pgp.PeopleID = [##Session.UserID##]
      AND pgp.GroupID IN (50000, 50011, 3)
    </Visible>
  </Rules>
</Form>
```

Magyarázat

```
<Form id="ScreenEvent" name="Események részletekkel" MaxWidth="95%" MaxHeight="100%">...</Form>
```

Az id és a name attribútumok kötelezőek, az id egyedi azonosítója az adott Screen, ez legyen az XML fájlnek a neve .a kiterjesztés nélkül, a name pedig az a szöveg amit megszeretnénk jeleníteni a menüben. A felugró képernyők esetén a MaxWidth és a MaxHeight beállításal a képernyő maximális méretét lehet megadni (ne foglalja ki automatikusan a rendelkezésre álló helyet egy olyan képernyő, amelyben csak alig van pár mező). Ennek az értéke lehet % és pixel is, pixel esetén csak a számot kell megadni, % jellel jelölni kell külön, ha százalékot állítunk be.

Osztott képernyők

Az első példa vonatkozik ide. <SplitContainers>...</SplitContainers> tagek között kell megadni a képernyő felosztását.

Tetszőleges számú <Split id="1" orientation="Vertical" splitterDistance="50%">...</Split> rakhatunk a SplitContainer tagek közé. Mindegyik Split egy felosztást definiál. Az id attribútum a split azonosítója (egész szám).

Az orientation attribútum értelemszerűen a Split elválasztó irányát adja meg. Lehetséges értékei a Horizontal és a Vertical. A splitterDistance attribútum az osztás helyét adja meg. Az érték lehet százalékban megadni, aszerint, hogy az adott SplitContainer-ben mekkora százalékot foglalnak el. A százalék jelet utána kell írni. Ha nincs ott a % jel, akkor pixelben értendő az osztás helye.

A splitterDistanceOnTablet: hasonló értékeket vehet fel, mint a splitterDistance. Tábla nézetben határozza meg az osztás helyét.

A Split tagen belül a Panel1 és Panel2 tagek adják meg az osztás bal és jobb oldalán megjelenő komponenseket (vagy egy újabb osztást).

A type attribútum adja meg az adott panelre kerülő kontrol típusát, lehetséges értékei:

- **Component:** Kötelező id attribútumot is írni, amit később a komponensek összekötésekor használni kell. Az id-nak az értéke az egész XML-re nézve egyedinek kell lennie (mármint Panel1 és Panel2 tagek id attribútumáról van szó). A tag értéke ilyenkor egy Component XML fájlneve a kiterjesztés nélkül.

```
<Panel1 type="Component" id="1">ComponentEventEditForm</Panel1>
<Panel2 type="Component" id="2">ComponentEvent03Hoz</Panel2>
```

- **Split.** Ekkor az adott panel tovább lesz bontva. Az tag értékének a másik Split id attribútumának az értékét kell megadni.

```
<Split id="1" orientation="Horizontal" splitterDistance="60%">
  <Panel1 type="Split">2</Panel1>
  <Panel2 type="Component" id="1">ComponentProjectTermekek</Panel2>
</Split>
<Split id="2" orientation="Vertical" splitterDistance="38%">
  <Panel1 type="Component" id="2">ComponentProjectsEdit</Panel1>
  <Panel2 type="Component" id="3">ComponentProjectConnections</Panel2>
</Split>
```

Osztások közötti kapcsolat definiálása

Miután az összes Split és Komponens elhelyezkedést megadtunk, ezután meg kell adnunk, hogy a különböző id-ju Komponensek mivel kapcsolódhatnak. Ez alapján fogja a program felépíteni a szűrő gombokat.

```
<Components>
  <Component id="1" isreversible="true" isswitchable="true">2</Component>
</Components>
```

Ebben az esetben az 1 és a 2 id Component-nek engedélyeztünk a kapcsolatot. Az irreversible és isswitchable attributumok jelenleg nincs használatban értékük legyen mindig true. Ezekkel a beállításokkal lehetne beállítani, hogy a két panel szétkapcsolható-e, illetve a kapcsolat iránya állítható-e.

Osztások közötti kapcsolat aktiválása

Hasonló szintaxissal, mint ez engedélyezett kapcsolat láncoknál, definiáljuk azokat a kapcsolatokat, amik már a megjelenésnél aktívak. Figyelnünk kell arra, hogy nem engedélyezett láncot (pl.: kört) ne csináljunk.

```
<ActiveConnections>
  <ActiveConnection id="1">2</ActiveConnection>
</ActiveConnections>
```

Osztatlan képernyők

A második példa vonatkozik ide. Abban az esetben, ha csak egy komponens szeretnénk a képernyőre, akkor mindössze egy sorra van szükség:

```
<Panel type="Component" id="1">ComponentUserEdit</Panel>
```

A típus itt értelemszerűen csak komponens lehet, és nincs szükség: SplitContainers, Components, ActiveConnections tagekre.

Lehetőség van láthatósági szabály megadására a <Rules>...</Rules> tag között. Ez a munkamódszer menüben való megjelenítés szabályozza (jogosultságok).

Node-ok

Tartalomjegyzék

Component

Egy képernyő komponensekből áll. A komponensek megjelenítő egységeket tartalmaznak! A komponenseken lévő fülekkel lehet köztük váltani! Egy komponensen tetszőleges számú megjelenítő (vagy fül) lehet! A fülök sorrendje azonos lesz az XML-ben a Tab tagek sorrendjével.

Magyarázat

```
<Component id="EventNotes">...</Component>
```

Ebben soroljuk fel az egyes füleket. Az id attribútum a komponens azonosítója az XML fájl nevével kell megegyeznie (mínusz a Component szó).

```
<Tab id="resztvevoktab">
  <Name>Résztevők</Name>
  <Display type="dgvc">PeopleInEvent</Display>
  <Visible type="SQL" return="boolean" default="false">
    SELECT CASE
      WHEN [#Filter.Template_Event#] = 'Rendezveny' THEN 'false'
      ELSE 'true'
    END
  </Visible>
  <RefreshParentOnClose>true</RefreshParentOnClose>
</Tab>
```

A kiragadott részletben egy „resztvevoktab” azonosítójú fület definiálunk, amelynek a felirata „Résztevők” lesz. A fülön egy grid megjelenítő (dgvc) lesz, aminek az XML neve dgvcPeopleInEvent lesz majd (lásd DGVC.doc). A Visible RuleValueType (lsd. RuleValueType.doc) beállítással lehet szabályozni a fül megjelenését. A Visible beállítást nem kötelező megadni.

```
<RefreshParentOnClose>true</RefreshParentOnClose>
```

Ha bezárjuk a képernyőt és az a fül volt kijelölve az első panelen, akkor bezárás után az alatta levő képernyőre tol egy refresh-t.

```
<Display type="dgvc">EventWorkflowDocument</Display>
```

A type attribútum lehet:

- dgvc: A tag értéke a fájl neve mínusz dgvc és a kiterjesztés. Grid, CardList, CardGrid, Workflow, Calendar, SectionList megjelenítők valamelyike lehet. (lsd. DGVC.doc)
- ChartFormPanel: Diagram megjelenítő. Ilyenkor a tag értéke a fájl neve a kiterjesztés nélkül (lsd. ChartFormPanel.doc)
- EditFormPanel: Egy formot jelenít meg adatbevitelhez. Ilyenkor a tag értéke a fájl neve a kiterjesztés nélkül (lsd. EditFormPanel.doc)
- EmbeddedWebDisplay: beágyazott web megjelenítő. Ilyenkor a tag értéke a fájl neve a kiterjesztés nélkül (lsd. EmbeddedWebDisplay.doc)

Példa

```
<?xml version="1.0" encoding="iso-8859-2" ?>
<Component id="EventNotes">
  <Tab id="MeghivottakForSales">
    <Name>Részvételre jelöltek</Name>
    <Display type="dgvc">RendezvenyMeghivottakForSales</Display>
    <Visible type="SQL" return="boolean" default="false">
      SELECT CASE
```

```

WHEN [##Filter.Template_Event##] = 'Rendezveny' AND 'Marketing' IN (SELECT ppg.GroupName FROM PeopleGroupsPeople ppg WITH(NOLOCK) WHERE ppg.Deleted = 0 AND ppg.PeopleID = [##Session.UserID##]) THEN
WHEN [##Filter.Template_Event##] != 'Rendezveny' THEN 'false'
ELSE 'true'
END
</Visible>
</Tab>
<Tab id="ttfchart">
<Name>Terv, tény, forecast</Name>
<Display type="ChartFormPanel">ChartFormSales</Display>
<CssClass>ChartTab</CssClass>
</Tab>
<Tab id="folydoktab">
<Name>Folyamat dokumentumok</Name>
<Display type="dgv">EventWorkflowDocument</Display>
</Tab>
</Component>

```

Node-ok

Tartalomjegyzék

Szabályok

Az Effector rendszerben számos szabály, funkció paraméterezése RuleValueType sémára történik. Leggyakoribb szabályok / funkciók:

- Visible: láthatósági szabály
- Readonly: csak olvashatósági szabály
- Required: kötelezőségi szabály
- WorkflowRequired: ha folyamatnál leokézásakor lefutó kötelezőségi szabály
- DefaultValue: alapértelmezett érték
- SessionValue: kontrolban a „házikó” ikon megnyomásához érték
- ComputedValue: mező értékének kiszámolására szabály
- ValidateRule: a mező érvényességének vizsgálata

pl.:

```
<DefaultValue type="Constant" return="int" default="0">szabály vagy érték</DefaultValue>
```

A type attribútum

A kiértékelendő szabály típusát állítja be, értéke lehet:

- Constant: a szabályban található hivatkozásokat kicseréli és kész
- Simple: ugyanaz mint a Constant, csak JavaScript motoron kiértékeli (egyszerű feltételek megadása)
- SQL: ugyanaz, mint a Constant, csak ez érték kiszámításához a szerverhez fordul, amelyik lefuttatja az adott SQL mondatot, és annak az eredményével tér vissza

A return attribútum

A szabály kiértékelése után kapott értéket milyen típusra kell konvertálni. Értéke lehet:

- int
- boolean
- string
- datetime

A default attribútum

Ha a szabály kiértékelése közben valamilyen hiba lép fel, akkor az itt megadott értékre lesz az eredménye a szabálynak.

A message attribútum

Leginkább ValidateRule szabályhoz használt. Mi legyen az üzenet nem valid érték esetén. Ha a szabály olyan eredményt hoz, amely nem értelmezhető boolean értéként, akkor az lesz az üzenet (ilyenkor az üres visszatérés azt jelenti, hogy valid az érték).

A szabályok szövegében használható hivatkozások

A hivatkozások [##.##] formátumban jelennek meg. A csoport lehet:

- Field: az üzleti objektum egyik mezőjének értéke
- Session: a bejelentkezett felhasználó adatai. Session.UserID, Session.UserName
- Special: aktuális dátum ilyesmi. Special.Today, Special.Date
- Filter: a panelek közötti kommunikációból (filterek) egy érték
- Triggered: csak folyamatlépéseknél használt. Ilyenkor a Field mindig annak a BO-nak az értéke ahonnan jön a folyamat, a Triggered pedig a létrehozandó BO mezője.

Példa

```

<DefaultValue type="Constant" return="datetime">[##Special.Today##]</DefaultValue>

<ComputedValue type="SQL" return="datetime" default="">
<![CDATA[
DECLARE
@RafordOra INT,
@RafordPerc INT

IF [##Field.RafordOra##] = " BEGIN
SET @RafordOra = 0
END ELSE BEGIN
SET @RafordOra = [##Field.RafordOra##]
END

IF [##Field.RafordPerc##] = " BEGIN
SET @RafordPerc = 0
END ELSE BEGIN
SET @RafordPerc = [##Field.RafordPerc##]
END

SELECT DATEADD(minute, @RafordPerc, DATEADD(hour, @RafordOra, CAST([##Field.StartDate##] AS DATETIME)))
]]>
</ComputedValue>

<Trigger businessObject="Event" template="Megrendeles03" type="Create" group="0">

```

```
<TriggerEvent>OKEvent</TriggerEvent>
<Condition type="Simple"><![CDATA[["##Field.Jovahagyva##"]=="43"]></Condition>
</Trigger>
```

Node-ok

Tartalomjegyzék

DGVC és PSearch típusok

A DGVC xml fájlok az XML csomag Display nevű könyvtárjában vannak. A fájlnev mindig dgvc szóval kezdődik. Egy ilyen xml a felületen Grid, CardList, CardGrid, TreeView, SectionList, Calendar kontrollként jelenhet meg.

- Grid: Táblázatos megjelenítő. Egyedül ebben a nézetben van összegző sor a táblázat alján, csak itt van ExcelExport, és csak itt van egyszerű szűrő (ami nem olyan jó...)
- CardList: Azonos méretű téglalap alakú divok-ról van szó, amelyben különböző helyen és méretben képeket, adatokat lehet megjeleníteni. Ebben a nézetben van csak lehetőség csoportosításra, illetve említést érdemel az iGraph is.
- CardGrid: Hasonlít a CardList kontrolhoz, ám itt egy sor egy divet jelenít meg, és az is jellemzően hosszúkas. Működik valahogy.
- TreeView: Ezzel fába lehet szervezni az adatokat. Leginkább folyamatábrák rajzolására használtuk eddig. Nem tökéletes.
- SectionList: A CardGrid megjelenítőre hasonlít a legjobban. Lényegében a rajta megjelenített elemek szorosan egymás után jelennek meg (nincs rés közöttük), illetve itt minden egyes elem eltérő méretű lehet (magasság). Jól használható dokumentumok dokumentum elemekből való összeállítására.
- Calendar: Egy naptárat rajzol ki, és a DDO elemeit elhelyezi benne.

Magyarázat

```
<Name>Users</Name>
```

Az XML fájl neve minusz a dgvc szó. Ez egyedileg azonosít egy dgvc-t.

```
<Caption>Felhasználók</Caption>
```

A dgvc címe, nincs használva a felületen, lehet üres is.

```
<DDO>DDBaseUsers</DDO>
```

Itt adjuk meg a DDO xml nevét (kiterjesztés nélkül). Kötelező (lsd. DDO.doc). Az itt hivatkozott DD fogja szolgáltatni az adatokat a táblázat alapú megjelenítőnek (Grid, CardList, CardGrid, TreeView, SectionList, Calendar)

```
<BusinessObject idcolumn="ID">BusinessObjectUser</BusinessObject>
```

A táblázathoz megadható egy üzleti objektum, amely üzleti objektum kezeli a kijelölt sorok törlését. Az üzleti objektum az idcolumn attribútumban megadott oszlopból szedi a szükséges egyedi azonosítót a törléshez. Ahhoz, hogy betöltéskor az alapértelmezetten kijelölt sor Out Filterei megfelelően ki legyenek küldve, meg kell adni ezt a taget.

```
<OrderColumn>PeopleName</OrderColumn>
```

Annak az oszlopnak az SQL neve, amely szerint alapértelmezetten rendezni szeretnénk a táblázat sorait. A tagnek lehet egy direction attribútuma, amely a rendezés irányát állítja be. Értéke lehet: Asc, Desc.

```
<Style>Card</Style>
```

Ezzel a beállítással a Grid megjelenítőt el lehet forgatni 90°-al (vagyis az oszlopokból lesznek a sorok). Ritkán használt, nem kiforrott.

```
<PSearchHandlingMethod>OnValueChange</PSearchHandlingMethod>
```

Ezzel be lehet állítani, hogy a táblázathoz tartozó szűrőmezők milyen esemény hatására indítsák el az adatlekérést a szervertől. Értéke lehet: OnValueChange, OnRefreshButton. OnRefreshButton esetén a felületen megjelenő szűrős gombot kell megnyomni (ilyenkor nem minden egyes szűrő változtatásakor történik meg az adatlekérés).

```
<OutFilterColumn>ID</OutFilterColumn>
```

Annak a DDObeli oszlopnak a neve, amely a filter kezelésben elsődlegesen részt vesz.

```
<DocumentDragAndDropPSearch>NewDocument</DocumentDragAndDropPSearch>
```

Ennek csak akkor van értelme, ha van a griden olyan PSearch, ami Dokumentum felvitelre szolgál. A tag-ek közé ennek a PSearch-nek az egyedi nevét kell írni. Ilyenkor kártya nézet esetén a dokumentumot az adott objektumhoz be lehet csatolni drag and drop módszerrel, rá kell húzni a fájlt a kártyára.

```
<AutoRefresh>1</AutoRefresh>
```

Bekapcsolja az automatikus frissülést. A Beírt érték percben értendő.

```
<RowsPerPage AllRowsMeans="1000">100</RowsPerPage>
```

Beállítja a lapméretet, és hogy mennyi sort hozzon le az Effector a számlálóra kattintásra. Ha értéke -1, akkor alaphoz hoz minden sort. Ha AllRowsMeans értéke 0, akkor ki van kapcsolva a funkció, -1 esetén az összes sort jelenti.

```
<SimpleFilter>false</SimpleFilter>
```

Ezzel ki lehet kapcsolni az egyszerű szűrő megjelenítését. Ha nincs meg ez a beállítás, a default az, hogy megjelenik.

Columns, Column

A táblázatban használt oszlopok (lehetnek rejtettek is) <Columns>...</Columns> tagek között kell felsorolni.

```
<Column name="PeopleName">
  <ColumnCaption>Felhasználó</ColumnCaption>
  <Visible>true</Visible>
</Column>
```

A Column tag name attribútuma egy DDO beli oszlop nevét tartalmazza (lsd. DDO.doc). A **disallowGroup** attribútum CardList esetén a csoportosítás funkcióban való részvételt szabályozza (lsd Visible tag). A **displayFormat** attribútum a megjelenő értéket lehet formázni (ezres csoportosítás # ### vagy # ###,## vagy # ###,00; stb...)

```
<ColumnCaption>Felhasználó</ColumnCaption>
```

Az oszlop felíratát tartalmazza. Ha az oszlop látható, de nincs címke beállítva, akkor a DDO beli oszlopnév jelenik meg a felületen.

```
<Visible>true</Visible>
```

Oszlop láthatósága. CardList esetén csoportosítani csak látható oszlopokra lehet, illetve olyan oszlopokra, amelyek nem láthatóak, de a Column tagnak van disallowGroup = false beállítása.

```
<ReadOnly>false</ReadOnly>
```

Ez egy RuleValueType, és azt adja meg, hogy az oszlop értéke szerkeszthető vagy sem az inplace editing funkcióval (jelenleg csak a táblázatos megjelenítésben működik).

```
<Type>DateTime</Type>
```

További lehetőség az oszlopban megjelenő értékek alakítására. Az értéke lehet: Date, DateTime. Ha az oszlop nem readonly, akkor ezzel lehet megmondani, hogy milyen kontrollal lehet szerkeszteni az értéket. Erről bővebben az inplace editing szakaszban.

```
<Width>220</Width>
```

Az oszlop szélessége állítható ezzel a taggel.

```
<MultiLine height="150">true</MultiLine>
```

Ha a <Style> Card, akkor ezzel lehet állítani azt, hogy több soros lehet az oszlopban tárolt érték, és a példában 150px magas legyen. Csak akkor él az opció, ha a Type nincs kitöltve, vagy TextBox az értéke.

PSearches, PSearch (szűrők)

```
<PSearches>
  <PSearch id="DescriptionText" displaytype="TextSearch" filtertype="simple">
    <Caption>Leírás</Caption>
    <ConnectedColumn>e.Description</ConnectedColumn>
    <Operator>like%</Operator>
  </PSearch>
</PSearches>
```

Szűrésre használt kontrolok, nyomógombok (új felvitel, törlés, export, stb). Szűrők definiálására a <PSearches>...</PSearches> tagek között felsorolt <PSearch>...</PSearch> tagekkel lehet.

Attribútumok

- id: Egyedileg azonosítani kell a vezérlőt. Nem lehet egy dgvc XML-ben két azonos nevű. displaytype: PSearch vezérlő típusát határozza meg. Lehetséges értékei:
 - ActionButton: egy nyomógomb jelenik meg, amely a következő funkciókkal bírhat: új objektum felvitel, objektum törlése, egy külső modul egy funkciójának meghívása, ProcessResultSet
 - TextSearch: Egysoros szövegszerkesztő vezérlő a szöveges kereséshez.
 - ComboSearch: Adott értéklistából történő szűrészt valósít meg egy ComboBox vezérlővel.
 - BoolSearch: Bool típusú adatoszlopra tudunk vele szűrni. Három állapotú jelölőnégyzet vezérlővel.
 - DateSearch: Dátumszűrészt tesz lehetővé, dátumkiválasztó vezérlővel.
 - DateRange: Dátum intervallumszűrészt valósít meg.
 - TagCloud: címkefelhő szűrés
 - QuerySearch: ComboBox-hoz hasonló szűrő, ám ilyenkor egy választó ablak ugrik fel (táblázatból lehet értéket választani)
- filtertype: A program jelenleg nem használja, értéke lehet simple és complex
- actiontype Nem nyomógomboknál nincs figyelembe véve.
 - Link :Nyomógombok esetén ezt az akciótípust kell megadni.
 - Outer: Külső dll-be történő hívás esetén ezt kell megadni
 - ProcessResultSet: ProcessResultSet funkció

Altagek

```
<ConnectedColumn>pep.PeopleID</ConnectedColumn>
```

A szűrés melyik oszlopra vonatkozzon. Itt egy érvényes SQL kifejezést kell megadni, amely a DDO SQL mondatára alapszik, abba illeszkedik.

```
<Operator>...</Operator>
```

Ez mondja meg, hogy milyen SQL operátor fog a szűrésben részt venni. A lehetséges értékei:

- %like%, like%, %like
- =, <, >, <=, >=, <>
- in: ilyenkor egy <Query>...</Query> altage is szükség van
- between: dátumintervallumok esetében használható

```
<Caption>...</Caption>
```

A szűrő feletti felirat, vagy a nyomógomb felirata.

```
<Query>...</Query>
```

Ez az in operátor esetében használható elem. Itt egy SQL where mondatrészt kell megadni, amely a ConnectedColumn-ban megadott oszlopra alapul. Azaz a táblázat where része a következő résszel fog kibővülni: ConnectedColumn in (elem tartalma). Ha Query értéke tartalmazza a #param# karakterláncot, akkor annak helyére lesz beillesztve a kontrol értéke.

```
<DefaultValue>...</DefaultValue>
```

Ez egy RuleValueType (Isd RuleValueType.doc), a szűrőben levő alapértelmezett értéket lehet vele beállítani

```
<ReadOnly>...</ReadOnly>
```

Ez egy RuleValueType (Isd RuleValueType.doc), a szerkeszthetőséget szabályozza.

```
<Visible>...</Visible>
```

Ez egy RuleValueType (Isd RuleValueType.doc), a láthatóságot szabályozza.

```
<SessionValue>...</SessionValue>
```

Ez egy RuleValueType (Isd RuleValueType.doc), a szűrőhöz tartozó házikó ikon értékét állítja be. Nem minden kontrolnak van házikó ikonja, illetve ha a kontrolhoz nincs beállítva SessionValue RVT, akkor meg sem jelenik a házikó ikon.

```
<Width>...</Width>
```

A kontrol szélességét szabályozza.

```
<ComputedValue>...</ComputedValue>
```

Ez egy RuleValueType (Isd RuleValueType.doc). Ezzel a beállítással a filter-ből érkező érték alapján lehet dinamikus kiszámolni egy szűrő értékét.

```
<Tooltip>...</Tooltip>
```

Ez egy RuleValueType (Isd RuleValueType.doc). Ezzel a beállítással lehet magyarázó szöveget (aka tooltip) paraméterezni a szűrőhöz.

```
<DisplayOver>...</DisplayOver>
```

A PSearch a tag értékéként megadott oszlop egyszer szűrője helyén jelenjen meg.

ComboSearch esetén

```
<ComboDescription>...</ComboDescription>
```

A ComboBox vezérlőt leíró XML állomány nevét határozza meg. Az állomány nevét a kiterjesztés nélkül kell megadni.

ActionButton esetén

```
<LinkMM>...</LinkMM>
```

Az új képernyő melyik munkamódszeren jelenjen meg.

```
<LinkScreen>...</LinkScreen>
```

Az új képernyő Screen-je

```
<LinkBusinessObject>...</LinkBusinessObject>
```

Az új képernyő BusinessObject-je

```
<LinkComponent>...</LinkComponent>
```

Az új képernyő Component-je

```
<Action>...</Action>
```

Nyomógomb esetén bír jelentéssel. Értéke lehet:

- NewObject: egy új felvétel indul el
- DeleteObject: törődik a kiválasztott sor
- ExcelExport: A táblázatban megjelenített adatokra meghívja a keretrendszer ExcelExport funkcióját (Excel táblázatot készít belőle)
- SimpleLink: egy szabadon beparaméterezett screen-t hoz fel, ahol tetszőleges további műveletek végezhetőek. Lehetséges attribútuma:
 - RefreshScreenAfterAction: true / false, a művelet végrehajtása után frissítse be a képernyőt
- ProcessResultSet: A táblázat adataira meghívja a keretrendszer ProcessResultSet funkcióját. Ebben az esetben az Action tagnek lehetséges attribútumai:
 - sp: A meghívandó tárolt neve.
 - CloseScreenAfterAction: true / false, bezárja a képernyőt a művelet befejezése után
 - RefreshScreenAfterAction: true / false, a művelet végrehajtása után frissítse be a komponenst. Elavult, használj helyette inkább a RefreshAfterAction-t.
 - RefreshAfterAction: screen / component / nothing. A RefreshScreenAfterAction hamisan állítja, hogy a képernyőt frissíti. A képernyő melyik része frissüljön a PRS meghívása után. Ha ez a beállítás nincs, de a RefreshScreenAfterAction van, akkor annak a beállítása lesz érvényes.
 - OnlySelectedRows: true / false, csak a kijelölt sorokat adja át a PRS funkciónak.
- Tetszőleges szöveg, amely egy külső DLL-be történő híváskor, mint paraméter fog átadódni. (Isd. Module tag)

```
<Module classname="orn_OTP.SpecProjectPDFExportButton"#{##LocalProgramPath##}OTP.dll</Module>
```

A meghívandó modul neve, teljes elérési útvonallal A .[##LocalProgramPath##] karakterláncit a rendszer kicseréli a saját útvonalára (C:\inetpub\wwwroot\Progen\bin). A classname attribútum, a modulban lévő osztályt határozza meg, amely osztálynak adott interfészt kell megvalósítania (IEVIDataGrid).

DateRange esetén

```
<ComboDescription>...</ComboDescription>
```

A választható értékek listáját egy ComboDescription xmlfájl nevét tartalmazza. Az állomány nevét a kiterjesztés nélkül kell megadni. Az értékészlet kulcsa dátumtól|dátumig formátumú kell, hogy legyen (2011-01-01|2011-02-01).

```
<ConnectedColumn>...</ConnectedColumn>
```

DateRange esetén a ConnectedColumnnak tartalmaznia kell a {0} és {1} karakter láncokat. Az elsőt a dátumtól, a másodikat a dátumig értékkel helyettesíti be a rendszer.

```
<Operator>...</Operator>
```

Az operátor csat between lehet.

TagCloud esetén

```
<PSearch id="tagCloud1" displaytype="TagCloud" filtertype="simple">
  <Caption>Címke felhő</Caption>
  <DDO>DDTagCloudProba</DDO>
  <TagGroupColumn>tagGroup</TagGroupColumn>
  <TagCaptionColumn>tag</TagCaptionColumn>
  <TagValueColumn>count</TagValueColumn>
  <TagIDColumn>tagValue</TagIDColumn>
  <TagPSearchDefinitionColumn>psearchDef</TagPSearchDefinitionColumn>
  <Width>430</Width>
  <DisplayGroupInControl>false</DisplayGroupInControl>
</PSearch>
```

DDO: annak a DDO-nak (Isd. DDO.doc) a neve, amely a címkefelhő adatait adja vissza. Nagyon fontos, hogy ez a DDO fogadni tudja a DGVC-n definiált szűrőket! (alias-ok SQL-ben ilyesmi)

- TagGroupColumn: A DDO azon oszlopa, amely a csoport nevét tartalmazza.
- TagCaptionColumn: A DDO azon oszlopa, amely a címkék feliratát tartalmazza.
- TagValueColumn: A DDO azon oszlopa, amely a címke mellett megjelenített számláló értékét tartalmazza.
- TagIDColumn: A DDO azon oszlopa, amely a címkehez tartozó (csoporton belüli) egyedi azonosítót tartalmazza.
- TagPSearchDefinitionColumn: A DDO azon oszlopa, amely a szűréshez definiál egy PSearch XML elemet.
- DisplayGroupInControl: Értéke lehet true vagy false. Ha false az értéke, akkor a kontrolon nem jelenik meg, hogy mely csoportokból lett választva az aktív címkék.

QuerySearch esetén

```
<PSearch id="MarcomListQuerySearch" displaytype="QuerySearch" filtertype="complex">
  <QueryDescription LinkScreen="ScreenChoiseRendezvenyLista" QueryDescription="DDChoiseRendezvenyLista" ValueMember="U_PRS_ID" DisplayMember="DisplayValue" />
  <Caption>Listák</Caption>
  <ConnectedColumn>pe.peopleid</ConnectedColumn>
  <Operator>in</Operator>
  <Query>
    (SELECT p.peopleid
     FROM orn_PeopleList m with (nolock)
     LEFT OUTER JOIN people p with (nolock) ON m.PeopleID = p.peopleid and p.deleted = 0
     WHERE m.Deleted=0 AND m.ProcessResultSetID = #param#)
  </Query>
</PSearch>

<QueryDescription>...</QueryDescription>
```

Ezzel a taggal lehet megadni a felugró képernyő adatait.. Attribútumok:

- LinkScreen: Annak a képernyőnek a neve, amely meg kell, hogy nyíljon a ... gomb megnyomásakor.
- QueryDescription: Annak a DDO-nek a neve, amely az ID-hoz visszakeresi a megjelenítendő szöveget.
- ValueMember: Annak az oszlopnak a neve, amely az értéket (vagyis az ID-t) tartalmazza.
- DisplayMember: Annak az oszlopnak a neve, amely az értékhez tartozó megjeleníteni kívánt értéket tartalmazza.

BoolSearch esetén

```
<TriState>false</TriState>
```

Ezzel lehet átkapcsolni három állású szűrőből két állásúba.

ChainListNode

```

<ChainListNodes>
  <ChainListNode>
    <ClickedColumn>CompanyName</ClickedColumn>
    <KeyColumn type="Company_ID">CompanyID</KeyColumn>
    <LinkMM>MMFolyamatok</LinkMM>
    <LinkScreen>ScreenCompanyEdit</LinkScreen>
    <LinkComponent>ComponentCompanyEdit</LinkComponent>
  </ChainListNode>
</ChainListNodes>

```

```
<ChainListNodes>...</ChainListNodes>
```

A táblázatban linkeket hoz létre, amelyek segítségével egy új képernyőre tudunk navigálni. Ez az elem ChainListNode elemeket fog össze. Ezekben az elemekben lehet a linkeket definiálni.

```
<ChainListNode>...</ChainListNode>
```

Egy nem kötelező attribútuma van: type. Értéke, ha New, akkor a felugró képernyő új adat felvitelnek fog számítani (nem szerkesztésnek). Ha az értéke NewOrExists, akkor a keretrendszer ellenőrzi hogy létezik –e a hivatkozott objektum. Ha igen akkor felhossa szerkesztésre, ha nem, akkor újat készít. Ha a type attribútum értéke ProcessResultSet, akkor a linke kattintva egy PRS hívás történik.

- ClickedColumn: Melyik oszlopnál jelenjen meg a link (Column tag name attribútuma)
- KeyColumn: Melyik oszlop értéke kerüljön a szűrőbe. A type attribútum a kimenő szűrő neve (nem használt?)
- LinkMM: Melyik munkamódszeren jelenjen meg az új ablak.
- LinkScreen :Melyik képernyőn jelenjen meg az új ablak.
- LinkComponent: Melyik komponens legyen aktív az új ablakon.

Summaries, Summary

Grid megjelenítő esetén lehetőség van egy összegző sor megjelenítésre a panel alján. Ezt a <Summaries>...</Summaries> tagek között felsorolt <Summary SummaryType="Sum" Column="EUR_MSRRP_grid" /> tagekkel lehet. A SummaryType lehet: Sum, Average, Count, Maximum, Minimum. A Column attribútum tartalmazza annak az oszlopnak a nevét, amelyre az összegzés kell.

```

<Summaries>
  <Summary SummaryType="Sum" Column="EUR_MSRRP_grid"></Summary>
  <Summary SummaryType="Sum" Column="EUR_MSRRP_grid_sulyozott"></Summary>
  <Summary SummaryType="Sum" Column="SO_grid"></Summary>
  <Summary SummaryType="Sum" Column="SO_sulyozott_grid"></Summary>
  <Summary SummaryType="Sum" Column="UM_grid"></Summary>
  <Summary SummaryType="Sum" Column="UM_sulyozott_grid"></Summary>
</Summaries>

```

Inplace editing

A táblázatos nézetben lehetőség van cellák értékének szerkesztésére. Alapvető szerkesztési lehetőségekről van szó, csodát ne várjon senki. A Type taggal lehet megadni, hogy milyen beviteli mezővel lehet szerkeszteni az adott oszlopot. Jelenleg TextBox és ComboBox típusok vannak lekezelve.

Az alapvető működés a következő: dgvc oszlop neve alapján kikeresi a definíciót a DD-ben. ott IdColumn tagben megadott oszlopot veszi alapul a mentéshez. Például, ha a számlázandó oszlopot szeretném szerkeszteni, akkor:

dgvc:

```

<Column name="Szamlazando" displayFormat="# ###">
  <Type>TextBox</Type>
  <FieldProperties type="System.Int32" />
  <ColumnCaption>Számházandó</ColumnCaption>
  <ReadOnly>false</ReadOnly>
</Column>

```

DD:

```

<Column name="EventID">
  <BusinessObject>BusinessObjectEvent</BusinessObject>
  <SqlName>e.EventID</SqlName>
  <IdColumn>EventID</IdColumn>
  <OutFilter type="Out">
    <Alias>Event</Alias>
  </OutFilter>
</Column>
<Column name="Szamlazando">
  <BusinessObject>BusinessObjectEvent</BusinessObject>
  <SqlName>e.Szamlazando</SqlName>
  <IdColumn>EventID</IdColumn>
</Column>

```

A felhasználó a számlázandó oszlopra kattintva kap egy szövegbeviteli mezőt, ami csak egész számot fogad el. A mentés pedig a BusinessObjectEvent-el történik, az id az EventID oszlopban van eltárolva (Szamlazando oszlop IdColumn tagje mondja meg ezt), és a Szamlazando mezőt fogjuk befrissíteni.

Fontos: Ha a ReadOnly szabály értéke (ez kiértékelhető) false, csak akkor jelenik meg a szerkesztő mező!

TextBox paraméterezése

```
<MultiLine height="150">true</MultiLine>
```

Többsoros (bevétel, egyenlőre csak megjelenítés)

```
<FieldProperties type="System.Int32" />
```

A beviteli mező által elfogadható értékeket szabályozza. A type attribútum a típust adja meg. Jelenleg System.String, System.Int16, System.Int32, System.Int64, System.Byte, System.Double lehet az értéke.

A maxLength és allowNull attribútumok csak System.String esetén vannak értelmezve, az első a maximálisan elfogadható hoszt szabályozza

```
<Mask>99999999-9-99</Mask>
```

ComboBox paraméterezése

```
<Type source="SQL" ComboDescription="ComboCompany" keyColumn="CompanyID">ComboBox</Type>
```

A korábban definiáltaktól kissé eltérő módon működik. Itt a felhasználó arra az oszlopra kattint, ami az értéket szövegesen mutatja, de a keyColumn attribútumban megadott oszlopra fogja menteni a kiválasztott elem azonosítóját, a BusinessObject és az id értéke itt is kiszámolódik, de ebben az esetben a keyColumn-ban megadott oszlop lesz a kályha.

CardList, CardGrid

```
<CardViewHTML>...</CardViewHTML>
```

Ha ez a tag szerepel a DGVC xmlben, akkor a keretrendszer átkapcsolja kártyanézetre a megjelenítést. Ennek a tagnek az innerText részében a kártya tartalmát leíró HTML utasításokat kell írni. A keretrendszer a [##mező##] karakterláncot a mező értékére cseréli. (Columns)

Attribútumok:

- sort: A rendezés combobox megjelenjen-e a szűrőmezők között. Értéke lehet true / false. Alapértelmezett értéke true.
- group: A csoportosítás combobox megjelenjen-e a szűrőmezők között. Értéke lehet true / false. Alapértelmezett értéke true.
- graph: A grafikon combobox megjelenjen-e a szűrőmezők között. Értéke lehet true / false.
- InsertBeforeControl: Értéke annak a PSearch-nek a neve, amely elé kell beszúrnai a csoportosítás, rendezés, grafikon comboboxokat. Ha ez a beállítás nincs, akkor ezek lesznek az utolsók.
- CardViewGroupColors: A csoportok eltérő stílusúak lehetnek. Értéke vesszővel elválasztott css osztálynevek. Jelenleg alter1, alter2, alter3, alter4, alter5, alter6, alter7, alter8, alter9, alter10 lehet. Ezek közül tetszőleges számú lehet. Ha több csoport van, mint az itt beállított css osztályok nevei, akkor az elsőől folytatja.
- DefaultGroupByColumn: Opcionális, értéke egy DDO column neve, amely alapján alapértelmezetten csoportosítani szeretnénk a kártyákat.
- DefaultGroupByDirection: Opcionális, értéke Asc, Desc, és az előző beállítással együtt kell / lehet használni.. Azt szabályozza, hogy a csoportok nevei milyen sorrendben jelenjen meg.
- CardViewGroupLayout: Opcionális, értéke Horizontal, vagy Vertical lehet. A csoportok vízszintesen vagy függőlegesen jelenjenek meg.
- Graph: Opcionális. Azoknak az oszlopoknak a nevei vesszővel elválasztva, amelyre alkalmazható a Graph megjelenítés.
- CardWidth="100%": kártya szélességének állítása, ha kell. Ha a CardWidth és a CardHeight egyszerre be van állítva, akkor a megjelenítés átkapcsol CardGrid megjelenítőre.
- CardHeight="45px": kártya magasságának állítása, ha kell. Ha a CardWidth és a CardHeight egyszerre be van állítva, akkor a megjelenítés átkapcsol CardGrid megjelenítőre. Különböző magasságú kártyák esetén a CardHeight attribútum értékének „FromCardViewHTML”-nek kell lennie, majd azon belül az első div stílusában kell megadni a height:”px (ahol a csillag lehet egy szám vagy egy mező neve szokásos módon megadva: [##MezoNev##]) pl.:

```
<CardViewHTML InsertBeforeControl="Interval" CardHeight="FromCardViewHTML" CardWidth="310px">
  <![CDATA[
    <div style="height:[##height##]px; display:[##Display##]">
      <div>...</div>
    </div>
  ]>
</CardViewHTML>
```

- CardCssFloat="none": kártyák css float beállításának felülbírálása
- <CardViews AlternateCardColumn="EventID">
- DefaultDisplayType = „grid”: ezzel lehetőség van megadni, hogy alapértelmezetten a kártya nézet a kártya megjelenítés helyett a grid megjelenítést használja.
- prsPsearchId vagy ProcessResultSetPsearch: Annak a PSearch-nek az azonosítója, amely a ProcessResultSet hívást elvégzi a kártyára kattintás után. Ennek a PSearch-nek nem kell, hogy látható legyen.
- DisableAppearanceToggle: letiltja a megjelenítés váltó gombot
- CardColumnCountColumn, CardWidthColumn, CardHeightColumn: a kártyákat egymás mellé teszi mátrixszerű elrendezésben. A CardColumnCountColumn annak az oszlopnak a nevét takarja, amelyben le van írva, hogy mennyi „oszlop” legyen a mátrixnak. A CardWidthColumn és a CardHeightColumn azoknak az oszlopoknak a nevét takarja, amely megmondja, hogy az adott kártya mekkora, ilyen formán lehetőség van különböző méretű kártyákat kirkanni. A méretet pixelben kell megadni.
- CardViewShowGroupElementCount: ha false az értéke, akkor nem jelenik meg a számláló a csoport neve mellett.

TreeView

```
<TreeViewHTML IDColumn="EventID" ParentIDColumn="ParentEventID" ArrowText="Description">
  <![CDATA[
    <div class="treeview_item_content_01" style="background:url(gfx/fabra_feladat.png); background-repeat:no-repeat; overflow:hidden; height: 100%;">
      <div class="listview_item_img" style="margin-top:6px;margin-left:6px;"></div>
      <div style="float:left;top:32px;left:7px;position:absolute;"></div>
      <div style="float:left;top:32px;left:7px;position:absolute;width:20px;text-align:center;"><font style="font-weight:bold;width:10px;text-align:center;">[##Megjegyzések##]</font></div>
      <div style="margin-top:5px;float:left;width:143px;word-wrap:break-word;">
        <div class="chainListNode" style="height:30px;overflow:hidden;font-size:11px;margin-top:4px;">[##Description##]</div>
        <div style="bottom:5px;left:33px;position:absolute;font-size:12px;width:110px;height:15px;overflow:hidden;">[##Felelos##]</div>
        <div style="height:25px;position:absolute;right:5px;bottom:5px;">
          </div>
        </div>
      </div>
    ]>
  </TreeViewHTML>
```

Ha ez a tag szerepel a DGVC xmlben, akkor a keretrendszer átvált folyamatábra a megjelenítésre. Ennek a tagnek az innerText részében a folyamatlépés táglalap tartalmát leíró HTML utasításokat kell írni. A keretrendszer a [##mező##] karakterláncot a mező értékére cseréli. (Columns) Attribútumok:

- IDColumn: Annak az DDO oszlopnak a neve, amely egy folyamatlépést (téglalap) egyértelműen azonosít.
- ParentIDColumn: Az adott téglalap szülőjének az azonosítóját tartalmazó DDO oszlop neve.
- ArrowText: Az összekötő nyílra írandó szöveget tartalmazó oszlop neve. Jelenleg még nem használt.

Calendar

```
<Calendar StartDateColumn="StartDate" EndDateColumn="EndDate" DisplayType="Daily">
  <![CDATA[
    <div class="calendar_item_content_01" style="overflow: hidden; height: 100%;">
      <div class="listview_item_img" style="margin:2px;"></div>
      <div class="chainListNode" style="overflow:hidden;font-size:10px;position:absolute;bottom:5px;left:20px;">[##Description##]</div>
      <div style="top:5px;left:20px;position:absolute;overflow:hidden;">[##ProjectName##]</div>
    </div>
  ]>
</Calendar>
```

Ha ez a tag szerepel a DGVC xmlben, akkor a keretrendszer átvált naptár a megjelenítésre. Ennek a tagnek az innerText részében a megjelenítő elem tartalmát leíró HTML utasításokat kell írni. A keretrendszer a [##mező##] karakterláncot a mező értékére cseréli. (Columns) Attribútumok:

- StartDateColumn: Annak a DDO-ban szereplő oszlopnak a neve, amely a naptárban megjeleníteni kívánt elem kezdő dátumát tartalmazza.
- EndDateColumn: Annak a DDO-ban szereplő oszlopnak a neve, amely a naptárban megjeleníteni kívánt elem befejező dátumát tartalmazza.
- DisplayType: nem használt, mindig Daily.

Ha a hozzá tartozó DD-ben felvesszünk egy Readonly nevű mezőt, akkor annak értékével megadhatjuk, hogy az adott esemény szerkeszthető vagy sem a naptár nézetben (Ha a Readonly mező értéke true vagy 1 akkor a mezőt nem szerkeszthetjük).

SectionList

```
<SectionListHTML CardWidth="auto" CardHeight="auto">
  <![CDATA[
    <div class="listview_item_content_01">
      <div style="margin:[##Margin##];text-align:[##Align##];font-size:12px;">
        <div>[##Hivatkozas_Tipus##] [##Type##] [##Name##] [##EditTitle##] [##EditMegallapitas##] [##EditMegallapitasForNew##] [##Edit_ProjectName##]</div>
      </div>
    ]>
  </SectionListHTML>
```

Ha ez a tag szerepel a DGVC xmlben, akkor a keretrendszer átvált szekciólista megjelenítésre. Ennek a tagnek az innerText részében a megjelenítő elem (1 sor a DDO-ból) tartalmát leíró HTML utasításokat kell írni. A keretrendszer a [##mező##] karakterláncot a mező értékére cseréli. (Columns) Attribútumok:

- CardWidth: kártya szélességének állítása, célszerű, hogy auto maradjon az értéke.
- CardHeight: kártya magasságának állítása, célszerű, hogy auto maradjon az értéke.
- CardCssFloat: nem használt

Példa

```
<?xml version="1.0" encoding="iso-8859-2"?>
<DataGridViewClass>
  <Name>Users</Name>
  <Caption>Felhasználók</Caption>
  <OutFilterColumn>ID</OutFilterColumn>
  <DDO>DDBaseUsers</DDO>
  <BusinessObject idcolumn="ID">BusinessObjectUser</BusinessObject>
  <OrderColumn>PeopleName</OrderColumn>
  <DocumentDragAndDropPSearch>NewDocument</DocumentDragAndDropPSearch>

  <Columns>
    <Column name="PeopleName">
      <ColumnCaption>Felhasználó</ColumnCaption>
      <Visible>true</Visible>
    </Column>
    <Column name="ID">
      <Visible>false</Visible>
    </Column>
    <Column name="UserID">
      <Visible>false</Visible>
    </Column>
    <Column name="Email">
      <Visible>true</Visible>
    </Column>
    <Column name="UserName">
      <ColumnCaption>Felhaszn. név</ColumnCaption>
      <Visible>true</Visible>
    </Column>
    <Column name="Department">
      <Visible>false</Visible>
    </Column>
  </Columns>

  <PSearches>
    <PSearch id="NewUser" displaytype="ActionButton" actiontype="Link">
      <LinkMM>MMPEP</LinkMM>
      <LinkScreen>ScreenNewUser</LinkScreen>
      <LinkBusinessObject>BusinessObjectPeople</LinkBusinessObject>
      <LinkComponent>ComponentUserEdit</LinkComponent>
      <Caption>Új felhasználó</Caption>
      <Action>NewObject</Action>
    </PSearch>
  </PSearches>
</DataGridViewClass>
```

Node-ok

Tartalomjegyzék

DDO és a filterek kezelése

Az Effector rendszerben a DataDefinitionObject-ek az üzleti objektumok és a megjelenítő komponensek között teremtik meg a kapcsolatot. A DDO a megjelenítő komponensek adatforrásaként funkcionál. A fájlneve mindig DD karakterláncsal kell kezdődnie és az XML csomag DisplayDefinition könyvtárjában találhatóak meg.

Magyarázat

```
<Name>DDCompany</Name>
```

A DDO egyedi azonosítója. Értéke mindig a fájlnev leszámítva a kiterjesztést.

```
<SqlSelect>
  from Company as c with (nolock)
  where Deleted = 0 and 1=1
</SqlSelect>
```

Ez a tag, egy SQL mondatot tartalmaz, az SQL mondat oszlopokat kiválasztó része nélkül. Az oszlopok definiálása majd a Columns elemben lesznek megadva. Ez az SQL mondat határozza meg, hogy az objektum milyen adatokat olvas fel az adatbázisból. Ezek az adatok fogják valamilyen megjelenítő komponensnek az adatforrásául szolgálni.

Lehetőség van tárolt eljárás megadására is, ekkor az SqlSelect tag-nek kell adni egy type attribútumot, aminek az értéke „StoredProcedure”. Ilyenkor az SqlSelect tag értéke a tárolt eljárás neve.

```
<Columns>...</Columns>
```

Ez a tag fogja össze az oszlopokat definiáló <Column>...</Column> tag-eket.

```
<Column name="CompanyID">
  <BusinessObject>BusinessObjectCompany</BusinessObject>
  <SqlName>c.CompanyID</SqlName>
  <IdColumn>CompanyID</IdColumn>
  <OutFilter type="Out">
    <Alias>Company_ID</Alias>
  </OutFilter>
</Column>
```

A Column tag name attribútuma lesz a DDO oszlopnév (amit majd DGVC-ben használni kell). Ennek meg kell egyeznie az oszlop SQL nevével (vagy alias-al).

```
<BusinessObject>BusinessObjectCompany</BusinessObject>
```

Az oszlop milyen üzleti objektumot határoz meg. Egy külső kulcs esetén, megadható a külső kulcs által meghatározott adattáblára vonatkozó üzleti objektum. Az oszlophoz tartozó üzleti objektumban megadott láthatósági szabályok befolyásolhatják a megjelenítendő sorokat és oszlopokat.

```
<SqlName>c.CompanyID</SqlName>
```

Az oszlop SQL definícióját tartalmazza. Ennek ugyanolyan szintaktikai szabályoknak kell megfelelnie, mint egy SQL-es oszlop kifejezésnek.

```
<IdColumn>CompanyID</IdColumn>
```

Ha az oszlop aktív a dgvc objektumon, akkor az itt megadott oszlop fog Out típusként szerepelni az Out-ban, mint kimenő szűrési paraméter. A többi oszlop IdColumn-ja is szerepelni fog az Out-ban, de mint csak paraméter

```
<OutFilter type="Out">
  <Alias>Company_ID</Alias>
</OutFilter>
```

A type attribútum az Out-ban történő megjelenés típusát határozza meg. Lehetséges értékei: Out és a Parameter. Az Alias tag az adott oszlopot ezen a néven helyezi be az Out-ba.

```
<InFilters>
  <InFilter alias="Company_ID">
    <MyField>CompanyID</MyField>
  </InFilter>
  <InFilter alias="Created_By_ID">
    <MyField>CreatedbyID</MyField>
  </InFilter>
</InFilters>
```

Az itt felsorolt bejövő szűrők fogják a DDO SQL mondata által meghatározott eredmény sorokat leszűrni. Itt általában egy tábla egyedi azonosítóját várjuk. Az alias attribútum a bejövő szűrő nevével határozza meg. Ilyen nevű Out típusú paramétert keres a bejövő szűrőben. A forward attribútum lehetséges értékei a „true” és „false” szövegek. True érték esetén a bejövő szűrőben szereplő adott alias-szal rendelkező paraméter a DDO Out-jában, kimenő szűrőjében, Out típusként, megjelenik.

A MyField tag teremt kapcsolatot a bejövő szűrők és a DDO oszlopai között. Itt az SqlName –et kell megadni. A bejövő Out típusú paraméter értéke lesz az SqlName-ban megadott oszlopnál az SQL mondat where részében összetársítva, megvalósítva így a szűrés műveletét.

```
<ForwardedFilters>
  <InOutFilter type="Parameter">Company_ID</InOutFilter>
  <InOutFilter type="Parameter">Project_ID</InOutFilter>
</ForwardedFilters>
```

Az itt felsorolt, a bejövő szűrőben szereplő paraméterek továbbítva lesznek a kimenő szűrőbe.

```
<InOutFilter>...</InOutFilter>
```

Az elemtartalom meghatározza, hogy melyik bejövő paraméter, a type attribútum (Out vagy Parameter) pedig meghatározza, hogy milyen típusúval kerüljenek be a kimenő szűrőbe.

Filterek

A filterek jelen esetünkben nem azokat a szűrőmezőket jelentik, amik a képernyőn jelennek meg, hanem a panelek és képernyők közötti kommunikációt megvalósító struktúrát, mechanizmust jelenti.

Paraméterezés szempontjából kulcsfontosságú szerepet kapnak a DDO xml fájlok. Ezekben az XML állományokban van leírva, hogy az adott megjelenítő (amihez a DDO tartozik milyen filter értékeket ad ki, illetve hogy ő milyen filter értékeket képes fogadni. Vannak Forwarded filterek, ez mindössze annyit jelent, hogy ami a bemenetén megjelenik az a kimeneten is látszani fog.

A filternek egyik fontos tulajdonsága az Alias neve. Ezen a néven jelenik meg a panelek és képernyők közötti kommunikációban az érték. Másik fontos tulajdonsága a típusa. A filterek típusa lehet Out és Parameter.

Out típusú filterből mindig csak egy lehet (persze a DDO-t lehet úgy paraméterezni, hogy az több különböző Out-ot adhatson ki, de ezek közül mindig csak egy lesz ténylegesen Out) Azt, hogy melyik Out filter megy ki ténylegesen Out-ként azt az IdColumn tag határozza meg. Ilyenkor a többi Out-ként paraméterezett filter Parameter-ként megy ki.

Parameter típusú filterek gyakran vesznek részt az editform-on egy kontroll alapértelmezett értékeként, vagy valamilyen szabályban. Ugyanakkor a Template_ Alias névvel kezdődő filtereknek speciális jelentése van, erről bővebben nemsokára.

DGVC-ben paraméterezett OutFilterColumn jelentőségéről itt is szót ejtek. Tegyük fel, hogy egy képernyő ketté van bontva. A bal oldalon az ügyletek vannak kilistázva, a jobb oldalon az ügylet-hez tartozó események. Ha az ügyleteket tartalmazó képernyőn a cég nevére kattintok, akkor felugrik a cégszerkesztő képernyő. Ugyanakkor ilyenkor az ügylet-esemény képernyőn az ügylet-hez tartozó eseményekre kell szűrni. Ilyenkor annak az oszlopnak a nevét kell beállítani az OutFilterColumn tag-ben, amely a jobb oldali események leszűréséhez szükséges beállítást tartalmazza a DDO-ban, jelen esetben ez a ProjectID.

Példa

```
<?xml version="1.0" encoding="ISO-8859-2" ?>
<DisplayDefinition>
  <Name>DDCompany</Name>
  <SqlSelect>
    from Company as c with (nolock)
    where Deleted = 0 and 1=1
  </SqlSelect>

  <Columns>
    <Column name="CompanyID">
      <BusinessObject>BusinessObjectCompany</BusinessObject>
      <SqlName>c.CompanyID</SqlName>
      <IdColumn>CompanyID</IdColumn>
      <OutFilter type="Out">
        <Alias>Company_ID</Alias>
      </OutFilter>
    </Column>
    <Column name="Name">
      <BusinessObject>BusinessObjectCompany</BusinessObject>
      <SqlName>c.Name</SqlName>
      <IdColumn>CompanyID</IdColumn>
    </Column>
    <Column name="StreetNumber">
      <BusinessObject>BusinessObjectCompany</BusinessObject>
      <SqlName>c.StreetNumber</SqlName>
      <IdColumn>CompanyID</IdColumn>
    </Column>
    <Column name="Street">
      <BusinessObject>BusinessObjectCompany</BusinessObject>
      <SqlName>c.Street</SqlName>
      <IdColumn>CompanyID</IdColumn>
    </Column>
    <Column name="AreaType">
      <BusinessObject>BusinessObjectCompany</BusinessObject>
      <SqlName>c.AreaType</SqlName>
      <IdColumn>CompanyID</IdColumn>
    </Column>
    <Column name="City">
      <BusinessObject>BusinessObjectCompany</BusinessObject>
      <SqlName>c.City</SqlName>
      <IdColumn>CompanyID</IdColumn>
    </Column>
    <Column name="Zip">
      <BusinessObject>BusinessObjectCompany</BusinessObject>
      <SqlName>c.Zip</SqlName>
      <IdColumn>CompanyID</IdColumn>
    </Column>
    <Column name="Fax">
      <BusinessObject>BusinessObjectCompany</BusinessObject>
      <SqlName>c.Fax</SqlName>
      <IdColumn>CompanyID</IdColumn>
    </Column>
    <Column name="Phone">
```

```
<BusinessObject>BusinessObjectCompany</BusinessObject>
<SqlName>c.Phone</SqlName>
<IdColumn>CompanyID</IdColumn>
</Columns>
</Columns>

<InFilters>
<InFilter alias="Company_ID">
<MyField>CompanyID</MyField>
</InFilter>
<InFilter alias="Created_By_ID">
<MyField>CreatedbyID</MyField>
</InFilter>
</InFilters>
</DisplayDefinition>
```

Node-ok

Tartalomjegyzék

ComboDescription

Egy lenyíló lista definícióját tartalmazó xml.

Az id attribútumnak meg kell egyezni az xml nevével (kiterjesztés nélkül).

Altagek:

- **ConnectionType**: kapcsolat típusa, jelenleg csak **SQLDbConnection**
- **SelectionString**: Egy select utasítás, ami a combo értékhalmozát kérdezi le. A where feltételbe tegyük bele az 1=1-et, hogy később tudjuk szűkíteni a combo tartalmát!
- **ValueMember**: Az a mező, amihez kapcsolódik a ComboBox
- **DisplayMember**: Amit meg szeretnénk jeleníteni.

A ValueMembernek és a DisplayMembernek szerepelni kell a SelectionString-ben is (mezőfelsorolásban).

Példa

```
<ComboDescription Name="ComboSectionTitle">
<ConnectionType>SQLDbConnection</ConnectionType>
<SelectionString>
<![CDATA[
select ProjectSectionID, SectionName
from or_n_ProjectSection with (nolock)
where deleted = 0 AND 1=1
order by Sort
]]>
</SelectionString>
<ValueMember>ProjectSectionID</ValueMember>
<DisplayMember>SectionName</DisplayMember>
</ComboDescription>
```

Node-ok

Tartalomjegyzék

EditForm

Az EditForm típus egy szerkesztő képernyő egy rekord celláinak felvitelére, módosítására.

Felépítés

Az EditForm az alábbi komponensekből tevődik össze:

- tartozik hozzá egy DisplayDefiniton, egy BusinessObject, egy egyedi Name, és egy Caption
- ControlGroup-ok, melyek tartalmazzák az egyes megjelenő Control-okat. Minden Controlnak az EditForm-on belül egyedi neve kell, hogy legyen, minden ControlGroup-nak egyéni order attribútum értéke kell, hogy legyen. Ez utóbbi határozza meg a Kontrolrok („függőleges”) sorrendjét.

Működés

- csak a lényegi műveletekről esik itt szó, mint lekérdezés, sor beszúrás, és módosítás. A törlés művelete a módosítás alá esik.
- lekérdezés: egy létező rekord esetén annak lekérése a BO-ban felsorolt mezők alapján.
- módosítás: a controlokban lévő értékek alapján udpate-eli a teljes sort, ami szerepel a BO-ban. Minden változásról készül egy bejegyzés az FSYS_Log táblába automatikusan. Speciális eset a törlés funkció, ez egy update –re fog ráfutni, ahol a Deleted mező értéke változik 1-re.
- beszúrás: két lépcsős művelet. Ahhoz hogy az esetlegesen kapcsolt komponensek (tipikusan az EditForm mellett jobbra egy grid, ami pl. az esemény résztvevőit mutatja) meg tudják kapni az újonnan felvett rekord azonosítóját – így teljes értékűen tudjanak működni – meg kell kapniuk az EditForm mögött lévő adatsor egyedi azonosítóját. Ami ilyenkor történik az, hogy az új gomb megnyomás után beszűrődik egy üres sor, majd ezt jeleníti meg az EditForm és a Save gomb megnyomása után egy sima módosítás fog lefutni. A definiált DefaultValue-k (BO-ban, vagy EditForm-on) értelemszerűen a beszúrással egy időben kerülnek be a sorba. Ilyenkor a Deleted oszlop értéke előbb 2 lesz, majd a mentés gomb után 0-ra változik az értéke.

Filterek

Létezik egy speciális filter csak EditForm esetén, amit a keretrendszer automatikusan kezel. Ezek:

- **JumpType** alias, az értéke New, vagy NoAction. Új felvitel esetén mindig van ilyen Filter, az értéke New. Egyéb LinkScreen esetén vagy nincs ilyen Filter, vagy az értéke NoAction. (note: a LinkScreen nem feltétlenül EditForm)

Példa:

```
<?xml version="1.0" encoding="iso-8859-2"?>
<Form>
<Name>EditFormEvent</Name>
<Caption>Esemény módosítása</Caption>

<DDO>DDEventEdit</DDO>
<BusinessObject>BusinessObjectEvent</BusinessObject>

<!-- Itt jönnek a ControlGroup -ok -->
</Form>
```

Magyarázat

Ezek itt az alap adatai egy-egy EditForm-nak. Az összes kötelező, és mindegyikből csak egy adható meg.

- A leíró gyökéreleme a <Form>
- A <Name> tag tartalmazza az EditForm egyedi nevét, ami EditForm a bo neve (mínusz Business Object szó)
- A <Caption> tag tartalmazza az EditForm címkéjét, mely jelenleg nem jelenik meg sehol.
- <DDO> tag-ben adjuk meg az EditForm-hoz tartozó DDO-t (filterek kezelésében játszik szerepet)
- <BusinessObject> tag-ben adjuk meg az EditForm-hoz tartozó BO-t
- <NearControlErrorDisplay> Ha a tag értéke false, akkor a kontrollokhöz tartozó hibaüzenet nem a kontrol mellett jelenik meg, hanem a a képernyő közepén, mint ahogy egy hagyományos üzenet megjelenik.

ControlGroup példa (alap Control)

```
<ControlGroup order="10">
<Control>
  <Type>Label</Type>
  <Name>Label_Name</Name>
  <ControlCaption>Név</ControlCaption>
  <Width>150</Width>
  <Left>10</Left>
</Control>
<Control>
  <Type>TextBox</Type>
  <Name>TextBox_Name</Name>
  <Width>200</Width>
  <Left>160</Left>
  <BindingName>Name</BindingName>
</Control>
</ControlGroup>
```

Magyarázat

- Egyes <ControlGroup> tag-ek között soroljuk fel az egymás mellett megjelenő Controlokat. Kötelező attribútum: order – a controlgroupok sorrendje. Érdemes a későbbi könnyebb bővíthetőség érdekében nem egyesével növelni! A ControlGroup tagek között soroljuk fel az egymás mellett megjelenítendő kontrollokat (tetszőleges számú lehet) pl.: Label – Textbox
- A tag-ek között írjuk le egy –egy control összes tulajdonságát. Ezek közül, ami mindenhol kötelező, és ugyanaz
 - Type: a Control típusa, pl. Label, vagy TextBox, részletesebben ld. lent
 - ControlCaption: a Label értéke, csak Label típus után van értelme
 - Name: a Control egyedi neve
 - Width: a control szélessége pixelben
 - Left: a control bal margótól vett távolsága pixelben
 - BindingName: minden Controlnak amihez tartozik adatbázis (BO!!) mező, azt itt kell megadni, így van összekapcsolva a DataSet-tel.

Control típusok

- TextBox
 - Egyszerű Textbox típus szabad szöveges bevitelre
 - Opcionális tag: <Multiline>true</Multiline> Többsoros Textbox
 - Opcionális tag: <Multiline UseHTMLEditor="true">true</Multiline> HTML szerkesztő Opcionális HTMLEditroToolbars attribútummal egy eszköztár (toolbar) készlet nevét lehet megadni, amit definiálni a FNetOnline2\uis\default\extras\lckeditor\config.js fájlban kell definiálni, pl a nevezett fájlban a toolbar_Basic beállítás. Ilyenkor az attribútum értéke Basic kell, hogy legyen
 - Opcionális tag: <Password>true</Password> nem jelenik meg a beírt karakter

- TouchValueList: Lehetőség van arra, hogy egy dobozban pár értéket megadva egyszerűsítsük az elem kiválasztását. Ezt a TouchValueList attributum megadásával tehetjük meg melyben egy SQL lekérdezéssel adjuk meg az elemeket. A helyes működéshez szükséges a type="Constant" valamint az oszlopneveknek Key-nek és Value-nak kell lenniük. Itt is használhatóak a szokásos [##Field.##], [##Filter.##],... értékek.

```
<TouchValueList type="Constant">
  <[[CDATA[SELECT TOP 10 CompanyID AS 'Key', Name AS Value FROM Company WITH(NOLOCK) WHERE deleted = 0]]>
</TouchValueList>
```

- Extra: maszkolási lehetőség, pl. <Mask>99999999-9-99</Mask> a maszkban a „9” jelenti a számot, „a” jelenti a betűt, * jelent vagy számot, vagy betűt a köztük lévő spec karakterek megjelennek és el is tárolódnak az adatbázis táblában

- Label

- QueryTextBox

```
<ConnectedControls>
  <ConnectedControl column="Product_Type">TextBox_Product_Type</ConnectedControl>
  <ConnectedControl column="Part_Number">TextBox_Part_Number</ConnectedControl>
  <ConnectedControl column="EUR_MSRRP">TextBox_Listaar</ConnectedControl>
</ConnectedControls>
```

A fenti beállítással lehet a kiválasztott sor valamelyik oszlopának értékét átadni az editform egy kontrolnak

- összetett listaválasztó

- TouchValueList: Lehetőség van arra, hogy egy dobozban pár értéket megadva egyszerűsítsük az elem kiválasztását. Ezt a TouchValueList attributum megadásával tehetjük meg melyben egy SQL lekérdezéssel adjuk meg az elemeket. A helyes működéshez szükséges a type="Constant" valamint az oszlopneveknek Key-nek és Value-nak kell lenniük. Itt is használhatóak a szokásos [##Field.##], [##Filter.##],... értékek, pl.:

```
<TouchValueList type="Constant">
  <[[CDATA[SELECT TOP 10 CompanyID AS 'Key', Name AS Value FROM Company WITH(NOLOCK) WHERE deleted = 0]]>
</TouchValueList>
```

- példa:

```
<Type LinkScreen="ScreenChoisePeople" QueryDescription="DDPeopleComplex" ValueMember="PeopleID" DisplayMember="PeopleName">QueryTextBox</Type>
```

itt az attribútumokban megadott paraméterek alapján fel tud hozni egy gridet, melyről egy sort lehet választani. Az érték a sor egyedi azonosítója (ValueMember), ez kerül be az adatbázisba. A megjelenített érték pedig DisplayMember attribútumban megadott oszlopnévhez tartozó érték a sorból.

- DateTime: DateTime típusú kontroll
- Date: Date típusú kontroll
- ComboBox: egy lenyíló listában szereplő key-value párokból lehet egyet választani ComboBox

a „ComboCegPotencial” ComboDescription-ben definiált lista jelenik meg, ami közül lehet választani értéke. A ComboDescription –ben van definiálva hogy mi a ValueMember – ez elmentődik, és a DisplayMember – ami megjelenik.

- TouchValueList: Lehetőség van arra, hogy egy dobozban pár értéket megadva egyszerűsítsük az elem kiválasztását. Ezt a TouchValueList attributum megadásával tehetjük meg melyben egy SQL lekérdezéssel adjuk meg az elemeket. A helyes működéshez szükséges a type="Constant" valamint az oszlopneveknek Key-nek és Value-nak kell lenniük. Itt is használhatóak a szokásos [##Field.##], [##Filter.##],... értékek.

```
<TouchValueList type="Constant">
  <[[CDATA[SELECT TOP 10 CompanyID AS 'Key', Name AS Value FROM Company WITH(NOLOCK) WHERE deleted = 0]]>
</TouchValueList>
```

- o Egy gyakori felhasználási eset, ha a lookup táblából szedjük az adatot, plusz 2 attribútumra is szükségünk lesz. Így nem kell mindegy egy lookup Group-ra külön leíró készíteni.
 - lookupfiltername, ez megmondja, hogy melyik filter helyére helyettesítsük be
 - lookupfiltervalue, ezt fogja behelyettesíteni

- Példa:

```
<Type source="SQL" ComboDescription="ComboFixedValuesLookup" lookupfiltername="LookupID" lookupfiltervalue="1">ComboBox</Type>
```

- CheckBox: egyszerű CheckBox típus, a mögötte lévő sql mező BIT típusú
- DocumentTextBox

```
<Type LinkScreen="ScreenChoiseDocument" QueryDescription="DDChoiseDocument" ValueMember="DocumentID" DisplayMember="OriginalFilename">DocumentTextBox</Type>
```

- o ugyanazok az attribútumok, mint a QueryTextBox esetén, erre azért van szükség, hogy el lehessen érni a már meglévő dokumentumok választó képernyőjét. Ez akkor lehetséges, ha nem DocumentBusinessObject az EditForm, amin rajta van a Control, hanem bármi más.
- o Ha az editform alapja nem DocumentBO, akkor szükség van egy DocumentObjectType beállításra is, amely megmondja, hogy milyen típusú dokumentumról van szó. (és a BO-ban is be kell állítani)
- o Méret korlátozáshoz FileSizeLimit tag használható, értéke megabájtban értendő
- o A típus korlátozásra is lehetőség van, space-el elválasztott MIME típusokat kell felsorolni az AllowedMIMETypes tag-gel.
- HorizontalRule
 - o Design elem, nincs hozzá adat. Egy egyszerű vízszintes vonalat rajzol ki, amivel lehet pl. tagolni az űrlapot.
- GoogleMap: elavult
- FormCardList: ez egy olyan control, ami kitesz egy cardlistet az editformra: megjegyzésekre, fórumszerű ablakokra jó. Egy-egy bejegyzés egy kártya, és ez az egész alatt egy Control szerepel, amivel lehet írni új megjegyzést.
 - o paraméterezése: bindingName nem kell, és meg kell adni a Control-on belül egy <CardScreenName>ScreenCommentsGrid</CardScreenName> tag-et, melyben megadunk egy screen-t, ami eljut a dgvc-ig, amit itt kell megjeleníteni
- InnerComponent
 - o lehetőség van a formon egy másik komponenst megjeleníteni (mondjuk egy olyant, amely a megrendeléshez tartozó tételek rögzítésére és megjelenítésre szolgáló táblázat van)
 - o paraméterezése:


```
<Type Component="ComponentEventNotes">InnerComponent</Type>
```
- Button: Egy gomb rendszerint DLL hívásokhoz
- SaveButton: Mentés gomb
- CancelButton: Mégsem gomb
- DeleteButton: Törlés gomb

Egyéb opciók, szabályok, kiértékelések

Lehetőség van megadni egyes controlonként az alábbiakat. Minden esetben egy RuleValueType-ről van szó, teljesen hasonló paraméterezéssel, lásd a felsorolás után.

Ezeknek a szabályoknak a paraméterezést lásd a RuleValue –k paraméterezéséről szóló fejezetben!

Szabályok, kiértékelések

- Required
 - o Addig nem engedi a felhasználót elmenteni a program, ameddig van olyan Control, ahol van „True”-ra kiértékelt Required szabály. Vizuális megjelenítését segíti az, ha a Label-re is meg van adva a szabály. Ilyen esetben True esetén a Label színe pirosra vált, jelezve ezzel, hogy oda muszáj valamit írni, vagy választani.
- WorkflowRequired
 - o Ua., mint a Required, annyi különbséggel hogy mindez csak akkor lép életbe, ha a `[[Field.Done]] == true`
- Readonly
 - o szerkeszthetőségi szabály
- DefaultValue
 - o Csak új felvitelnél(!) kiértékelt default érték
- SessionValue
 - o Megadható egy konstans vagy számított érték, és ha ez megvan, akkor kontrol jobb oldalán megjelenik egy extra gomb, melyet megnyomva mindig ez az érték kerül be a control-ba. Tipikusan a bejelentkezett felhasználó maga, az ő cége, stb.
- Visible
 - o láthatósági szabály
- Tooltip
 - o Tooltip címke, egy string RVT, ha nem üres, akkor megjelenik a tooltip ikon. Ha az ikon felé viszi a kurzort a felhasználó, akkor megjelenik a szöveg.
- Warning
 - o Egy kis piros felkiáltójeles kör jelenik meg a hatására, a tooltip-nek lehet adni szöveget. alapvetően string RVT, és ha nem üres, akkor jelenik meg.
- ValidateRule
 - o validálási szabály, addig nem engedi elmenteni az űrlapot, míg az itt definiált feltétel true-t nem ad eredményül, illetve nem boolean visszatérési érték esetén az üres üzenet jelenti, hogy valid a kontrolba írt érték
- ComputedValue
 - o számított mező érték, mely függ folyamatosan a többi mező állapotától.
 - o az érték újraszámolódásnál azt veszi figyelembe, hogy a kiértékelésben mely mezőkre hivatkozunk (`[[Field.Valami]]`) Ezeknek a változására fog valami történni, további beállításokat lásd: ReComputing
 - o az újraszámolások egyéb szabályokra is hatással vannak, mint pl. Visible, Readonly, stb.
- ComputedValueList
 - o számított lista érték Combo típus esetén, mely függ folyamatosan a többi mező állapotától
 - o az érték újraszámolódásnál azt veszi figyelembe, hogy a kiértékelésben mely mezőkre hivatkozunk (`[[Field.Valami]]`) Ezeknek a változására fog valami történni, további beállításokat lásd: ReComputing
 - o az újraszámolások egyéb szabályokra is hatással vannak, mint pl. Visible, Readonly, stb.

Egyéb:

- ReComputing: ez adja meg az egyes Controloknál, hogy az esetleges számított mezők milyen eseményekre számolódjanak újra.

TriggerOnRefresh

- o Értékei:
 - All: Mindig, az ablak megjelenésekor is. Nem használt.
 - OnChange: Csak érték változáskor. Alapértelmezett.
 - TriggerOnRefresh: A form frissítésekor vegye úgy mintha változott volna a kontroll értéke, így az összes szabály és ComputedValue lefut, ami hivatkozik erre a mezőre – ahol ez be van állítva.

ControlGroup példa (nyomógombok)

```
<ControlGroup order="500">
<Control>
<Type>DeleteButton</Type>
<Name>Del</Name>
<ControlCaption>Törlés</ControlCaption>
<Width>80</Width>
<Left>160</Left>
</Control>
```

```

<Control>
  <Type>SaveButton</Type>
  <Name>Save</Name>
  <ControlCaption>Mentés</ControlCaption>
  <Question type="OkCancel">
    <Condition type="SQL" return="boolean" default="false">select case when not exists(select documentconnectionid from documentconnection where boid = [##Field.ProjectID##] and botype = 'project' and deleted = 0) then 't'
    <Message type="Constant" return="string" default="">Az ügylethez jelenleg nincs csatolva dokumentum. Biztosan így szeretné elmenteni? Ha igen, kattintson az "ok", ha nem akkor a "mégsem", vagy "cancel" gombra!</Me
  </Question>
  <CloseWindow>false</CloseWindow>
  <Width>80</Width>
  <Left>250</Left>
</Control>
<Control>
  <Type>CancelButton</Type>
  <Name>Cancel</Name>
  <ControlCaption>Mégsem</ControlCaption>
  <Width>80</Width>
  <Left>340</Left>
</Control>
</ControlGroup>

```

Magyarázat

Annyiban speciálisak az ilyen Controlok, hogy nincs Label, illetve értelemszerűen sosincs bindingname tag. A fentebb felsorolt RuleValueType-ok nagy része ugyanúgy használható rájuk, mint sima Controlok esetében. Jellemzően a Visible szabály.

Két extra RuleValueType-ot meg lehet adni a gomboknál, melyek összetartoznak, egy Question tag fogja össze őket, lásd példa.

```
<Question type="OkCancel">
```

- A type attribútum jelenleg OkCancel, Ok, Cancel lehet. OkCancel: Azt jelenti, hogy egy olyan ablak fog feljönni a gombnyomásra, mely tartalmaz egy „OK”, és egy „Cancel” gombot. Az OK gomb megnyomására a művelet (mentés/törés/mégse) folytatódik, a Mégsem gomb megnyomására nem fog történni semmi, folytatódik a szerkesztés. Ok: tulajdonképpen egy tájékoztató ablakot lehet megjeleníteni mentés előtt. Cancel: szintén tájékoztatást jelenít meg, de meg is akadályozza a mentésre.
- A Question tag-en belül két RuleValueType-nak kell szerepelni, az egyik neve Condition, a másik pedig Message. A Condition-t adjon vissza true-t vagy false-t, és true esetén fog feljönni a párbeszédablak, melyben a Message értéke fog látszódni. Mindkét RuleValue-ban használhatóak a Field.Valami értékek.
- CloseWindow: Ha ennek a beállításnak az értéke false, akkor az editform a mentés után nem záródik be. Értéke lehet true és false, az alapértelmezett a true

Node-ok

Tartalomjegyzék

FragmentType

Első példa

```

<?xml version="1.0" encoding="iso-8859-2"?>
<FragmentType>
  <Name>TypeProjectRendszertervezes</Name>
  <Caption>Ügylet</Caption>
  <DDO>DDProjectComplex</DDO>
  <BusinessObject>BusinessObjectProject.Rendszertervezes</BusinessObject>

  <ControlGroup order="200">
    <Control>
      <Type>Label</Type>
      <Name>Label_Raforditas</Name>
      <ControlCaption>Becsült ráfordítás</ControlCaption>
      <Width>150</Width>
      <Left>10</Left>
    </Control>
    <Control>
      <Type>TextBox</Type>
      <Name>TextBox_Raforditas</Name>
      <Width>200</Width>
      <Left>160</Left>
      <BindingName>Raforditas</BindingName>
    </Control>
  </ControlGroup>
</FragmentType>

```

Második példa:

```

<?xml version="1.0" encoding="iso-8859-2"?>
<FragmentType>
  <Name>TypeEventTelefon</Name>
  <Caption>Esemény</Caption>
  <DDO>DDEventEdit</DDO>
  <BusinessObject>BusinessObjectEvent.Telefon</BusinessObject>
</FragmentType>

```

Magyarázat

A BusinessObject-nél szó volt arról, hogy lehet származtatott BusinessObject –et létrehozni. A Fragment leegyszerűsítve nem más, mint ehhez a származtatott BusinessObject-hez tartozó EditForm részlet. Az XML csomag FragmentTypes könyvtárjában laknak az XML állományok.

Itt elegendő csak a származtatott BO kiegészítő mezőjéhez definiálni az EditForm-on megjelenő kontrolokat. Ha nincs kiegészítő mezője a származtatott BusinessObject-nek, akkor is szükség van Fragment XML fájlra, ilyenkor csak a minimális szükséges beállításokat tartalmazza az XML (második példa).

Ha a Fragment-ben egy olyan id-val rendelkező ControlGroup definiál, ami a szülő BusinessObject-hez tartozó EditFormon is van, akkor az felülbírálja a szülő BO-hoz tartozó editform beállításait.

```
<Name>TypeProjectRendszertervezes</Name>
```

A fájl neve mínusz a kiterjesztés. Erre a következő szabály érvényes TypeAltípus lsd első példa.

```
<Caption>Ügylet</Caption>
```

Nincs jelenleg használva, de nem hagyható ki a tag.

```
<DDO>DDProjectComplex</DDO>
```

A szülőBO-hoz tartozó DDO. A filterek kezeléséhez van rá szükség.

```
<BusinessObject>BusinessObjectProject.Rendszertervezes</BusinessObject>
```

Fontos különbség az EditForm-hoz képest, hogy itt a BusinessObject megadása mainBO.típus formában történik (ahogy az első példában lehet látni).

A kiválasztható Fragmentek definiálása

A FragmentTypes könyvtárban a keretrendszer a szülő BO-hoz keres egy olyan XML-t, amely leírja, hogy milyen lehetséges típusok vannak. Ha main BO a BusinessObjectProject, akkor a keretrendszer egy ProjectTypes.xml fájlt keres.

Első példa:

```
<?xml version="1.0" encoding="iso-8859-2" ?>
<FragmentTypes>
  <Name>Dokumentum típusok</Name>
  <BusinessObject>BusinessObjectDocument</BusinessObject>
  <ListingType>XML</ListingType>
  <Fragments>
    <Fragment id="10" type="Dokumentum">
      <FragmentCaption>új dokumentum</FragmentCaption>
    </Fragment>
  </Fragments>
</FragmentTypes>
```

Második példa:

```
<?xml version="1.0" encoding="iso-8859-2" ?>
<FragmentTypes>
  <Name>Project típusok</Name>
  <BusinessObject>BusinessObjectProject</BusinessObject>
  <ListingType>SQL</ListingType>
  <StoredProcedure>osp_getProjectTypes</StoredProcedure>
  <Fragments />
</FragmentTypes>
```

Az első példában XML-ben van felsorolva a típusok, a második példában egy tárolt adja a lehetséges típusokat.

Node-ok

Tartalomjegyzék

Workflow

Előszó

A dokumentum a folyamatok paraméterezésével foglalkozik. Egyes elemeket más dokumentumokból kell hozzáolvasni.

A rendszer képes verziózásra is.

A rendszerben található folyamatok listáját, illetve az egyes folyamatlépések listáját a Workflow könyvtárban található XML állományok írják le.

WorkflowList.xml

Ebben a fájlban van felsorolva, hogy mely folyamatok léteznek a rendszerben.

```
<?xml version="1.0" encoding="iso-8859-2" ?>
<WorkFlows>
  <Workflow name="Igeny">Igény</Workflow>
  <Workflow name="Altalanos">Általános feladat</Workflow>
  <Workflow name="Rendszertervezes">Rendszertervezés (feladat)</Workflow>
  <Workflow name="Programozas">Programozás</Workflow>
</WorkFlows>
```

A fájlban léteznie kell akkor is, ha nincs folyamat az ügyfélnél. Ebben az esetben nincs benne egy Workflow tag sem. A példában négy folyamat van definiálva a name attribútum lesz a típusa, az érték pedig a folyamat beszédes neve. Ez az információ nincs megjelenítve sehol a felületen.

Minden workflow taghez keresni fog a keretszrendszer egy XML fájlt, mégpedig Workflow..xml nevű fájlt, ahol a a Workflow tag name attribútumának értékét jelenti, a pedig egy sorszám, amely 1-től indul.

A folyamat indításakor az adatbázisban a Workflow nevű táblában keletkezik egy sor. Az ObjectType oszlopban lesz a Workflow tag name attribútumának értéke. A keletkezett WorkflowID értéket bejegyzni a rendszer a folyamatban részt vevő objektumok WorkflowID oszlopába. Ezek az objektumok alapvetően BusinessObjectEvent (Event tábla) és BusinessObjectProject (Project tábla) BO-k.

WorkflowRendszertervezes.1.xml

```
<?xml version="1.0" encoding="ISO-8859-2" ?>
<Workflow name="Rendszertervezes" version="1">
  <WorkflowStep name="BusinessObjectProjectRendszertervezes">Project rögzítése</WorkflowStep>
  <WorkflowStep name="BusinessObjectEventRendszertervezes01Elokszites">01. Előkészítés alatt</WorkflowStep>
  <WorkflowStep name="BusinessObjectEventRendszertervezes02Megvalosithato">02. Megvalósítható</WorkflowStep>
  <WorkflowStep name="BusinessObjectEventRendszertervezes03Megoldva">03. Megoldva</WorkflowStep>
</Workflow>
```

A Workflow tag name attribútuma megegyezik a WorkflowList.xml fájlban paraméterezettel. A version attribútum értéke meg kell egyeznie a fájlnevben levő számmal (.xml előtti szám).

A WorkflowStep tagek sorolják fel a folyamatban résztvevő objektumok listáját. A name attribútum jelzi a lépéshez tartozó BusinessObject nevét, az érték pedig a folyamat lépés beszédes neve. Ez a felületen nem jelenik meg sehol.

A folyamatlépések egymásutánosságát az egyes BusinessObject XML fájlok írják le a Triggers taggel (Isd. BusinessObject.doc).

Létre kell hozni egy xml állományt, melyet a Workflow könyvtárba kell tenni, a neve pedig WorkflowPelda.xml. Tartalma:

```
<?xml version="1.0" encoding="ISO-8859-2" ?>
<Workflow name="Pelda" version="1">
  <WorkflowStep name="BusinessObjectProjectPelda">Project rögzítése</WorkflowStep>
  <WorkflowStep name="BusinessObjectEventPeldaLepes1">01. Lépés</WorkflowStep>
  <WorkflowStep name="BusinessObjectEventPeldaLepes2">02. Lépés</WorkflowStep>
  <WorkflowStep name="BusinessObjectEventPeldaLepes2a">02.a Lépés</WorkflowStep>
  <WorkflowStep name="BusinessObjectEventPeldaLepes3">03. Lépés</WorkflowStep>
</Workflow>
```

A fenti xml-t be kell jegyezni a Workflow mappában található WorkflowList.xml-be:

```
<?xml version="1.0" encoding="iso-8859-2" ?>
<WorkFlows>
```



```
<Workflow name="Pelda">Új példa folyamat</Workflow>
</WorkFlows>
```

A folyamat 1. lépése (BusinessObjectEventPeldaLepes1.xml)

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<BusinessObject mainbo="Event" Name="PeldaLepes1">
  <UniqueID>U_Event_ID</UniqueID>

  <Fields>
    <Field name="StartDate">
      <DefaultValue type="Constant" return="datetime">[##Special.Today##]</DefaultValue>
    </Field>
    <Field name="EndDate">
      <DefaultValue type="Constant" return="datetime">[##Special.Today##]</DefaultValue>
    </Field>
    <Field name="CreatedByID">
      <DefaultValue type="Constant" return="int">[##Session.UserID##]</DefaultValue>
    </Field>
    <Field name="Description">
      <DefaultValue type="Constant">01. Lépés</DefaultValue>
    </Field>
    <Field name="Task">
      <DefaultValue type="Constant">true</DefaultValue>
    </Field>
    <Field name="Done">
      <DefaultValue type="Constant">false</DefaultValue>
    </Field>
    <Field name="Felelos">
      <DefaultValue type="Constant">14564</DefaultValue>
    </Field>
  </Fields>

  <Triggers>
    <Trigger businessObject="Project" template="Pelda" type="Create" group="0">
      <TriggerEvent>Created</TriggerEvent>
    </Trigger>
  </Triggers>
</BusinessObject>
```

BusinessObjectEventPeldaLepes2.xml Triggers részre

```
<Triggers>
  <Trigger businessObject="Event" template="BusinessObjectEventPeldaLepes1" type="Create" group="0">
    <TriggerEvent>OKEvent</TriggerEvent>
    <Condition type="Simple">[##Field.Mezo1##] = 'Igen'</Condition>
  </Trigger>
</Triggers>
```

BusinessObjectEventPeldaLepes2a.xml Triggers részre

```
<Triggers>
  <Trigger businessObject="Event" template="BusinessObjectEventPeldaLepes1" type="Create" group="0">
    <TriggerEvent>OKEvent</TriggerEvent>
    <Condition type="Simple">[##Field.Mezo1##] = 'Nem'</Condition>
  </Trigger>
</Triggers>
```

BusinessObjectEventPeldaLepes3.xml Triggers részre

```
<Triggers>
  <Trigger businessObject="Event" template="BusinessObjectEventPeldaLepes2" type="Create" group="0">
    <TriggerEvent>OKEvent</TriggerEvent>
  </Trigger>
</Triggers>
```

Node-ok

Tartalomjegyzék

Dokumentum-kezelés

Működési mechanizmus

A felhasználó a felületen kiválaszt egy fájlt becsatolásra. A kiválasztás után az Effector feltölti a fájlt a szerverre egy átmeneti könyvtárba. Attól függően, hogy korábban mi volt a beviteli mező értéke feljön egy ablak, amelyben megkérdezi a felhasználót, hogy a most feltöltött dokumentum az előző új verziója lesz-e. A Mentés gomb megnyomása után a keretrendszer átnevezi, és végleges helyére másolja a fájlt.

A tökéletes működéshez szükséges beállítások

Az Effector könyvtárjában található _temp\fileuploads\ könyvtárra meg kell adni a jogot, hogy a keretrendszer írni tudja. Ide kerülnek az átmeneti fájlok. A végleges helyre másolás után itt is marad egy példány, így ezt a könyvtárat időről időre ki kell pucolni. Továbbá az Effectorhoz rendelt AppPoolnak olyan felhasználóval kell futnia, amelynek majd a végleges dokumentum könyvtárra van joga írni.

Document tábla

A Document táblában egy sor reprezentál egy dokumentum verziót. A ParentDocID oszlop jelzi, hogy ő melyik dokumentum egyik verziója. A Version oszlopban van, hogy ez ősdokumentum hányadik verziója.

DocumentConnection tábla

A DocumentConnection táblában van rögzítve, hogy egy Document sor mely objektumhoz (Event, Project, ...) van rendelve. A Document táblán található trigger ebbe a táblába is beszűr egy sort. Ilyenkor a beszűrőhöz a Document táblában található BoType és BoID oszlopokat használja fel. A felületen a Document tábla BoType és BoID oszlopait használni a join-okhoz nem javasolt!

DocumentTextBox control

Ezen a kontrolon keresztül lehet becsatolni fájlokat a rendszerben. Az adatbázisból csak akkor lehet dokumentumot választani, ha az XML-ben be van paraméterezve egy LinkScreen, és az EditForm-nak a mainBo-ja nem Document BO!

Dokumentumbeccsatolás paraméterezése

Az XML csomag gyökér könyvtárjában egy DocumentParameters.xml fájlban kell megadni a beállításokat: hova másolja a fájlt, és milyen néven.

```
<DocumentBusinessObjectType>...</DocumentBusinessObjectType>
```

Minden dokumentum típusra kell egy ilyen tag csoport.

```
<Type>Dokumentum</Type>
```

A dokumentum alítípusának a neve.

```
<AttachType>Copy</AttachType>
```

Ez csak Copy lehet.

```
<CopyPathTemplate type="SQL" return="string" default="">...</CopyPathTemplate>
```

Ez egy RuleValueType. Az itt megfogalmazott szabállyal lehet megfoglamazni a becsatolt dokumentum helyét (hova legyen másolva a fájl. Az elérési út mindig a szerveren van értelmezve!)

```
<FileNameTemplate type="SQL" return="string" default="">...</FileNameTemplate>
```

Ez egy RuleValueType. Nem kötelező. Egy szabályban lehet előállítani a becsatolt dokumentum fájlnevét.

Példa

```
<?xml version="1.0" encoding="iso-8859-2"?>
<DocumentParameters>
<DocumentBusinessObjectType>
<Type>Dokumentum</Type>
<AttachType>Copy</AttachType>
<CopyPathTemplate type="SQL" return="string">
<![CDATA[
DECLARE
@base_path VARCHAR(2000),
@company_name VARCHAR(2000),
@project_name VARCHAR(2000),
@objectType VARCHAR(2000),
@ProjectID INT

SET @base_path = 'p:\teszt\

SELECT TOP 1
@objectType = ISNULL('' dbo.ofn_RemoveSpecialCharacters_COLLATE(otype.DisplayLabel,'0-9a-zA-Z ') '', ''),
@project_name = '' dbo.ofn_RemoveSpecialCharacters_COLLATE(p.Name,'0-9a-zA-Z ') '',
@company_name = (SELECT TOP 1 dbo.ofn_RemoveSpecialCharacters_COLLATE(c.Name,'0-9a-zA-Z ') FROM Company AS c WITH(NOLOCK) WHERE CompanyID = [##Field.x_CompanyID##])
FROM Project AS p WITH(NOLOCK)
LEFT JOIN FSYS_ObjectTypesLookup AS otype WITH(NOLOCK) ON otype.ObjectType=p.ObjectType AND otype.ParentObject = 'Project' AND otype.Deleted = 0
WHERE p.ProjectID = [##Field.x_ProjectID##]

SELECT @base_path @company_name @project_name @objectType
]]>
</CopyPathTemplate>
<!-- <FileNameTemplate type="SQL" return="string">
SELECT dbo.ofn_CutAccent(dbo.ofn_RemoveSpecialCharacters_COLLATE([##Field.OriginalFileName##]','0-9a-zA-Z'))
</FileNameTemplate-->
</DocumentBusinessObjectType>
</DocumentParameters>
```

Node-ok

Tartalomjegyzék

Általánosan beparaméterezett funkciók megismerése

Node-ok

Tartalomjegyzék

Diagram

Diagramot lehet vele megjeleníteni. Az XML csomag ChartForm könyvtárjában található a fájlok. A fájl neve minden esetben ChartForm karakterláncsal kell kezdődnie.

Magyarázat

```
<Name>ChartFormProjectActivity</Name>
```

Mindig a fájl neve mínusz a kiterjesztés.

```
<DDO>DDChartProjectActivity</DDO>
```

A diagram adatait visszaadó DDO xml (lsd DDO.doc) neve.

```
<ChartType>areaspline</ChartType>
```

Az alapértelmezett diagramtípus

```
<Legend position="Bottom">true</Legend>
```

Az értéke lehet true / false, és azt szabályozza, hogy látható-e a jelmagyarázat. A position attribútum a jelmagyarázat helyét adja meg.

```
<TopText>...</TopText>
```

Lehetőség van a diagram fülé egy statikus szöveget írni.

```
<BottomText>...</BottomText>
```

Lehetőség van a diagram alá egy statikus szöveget írni.

```
<SeriesIDColumn>SeriesID</SeriesIDColumn>
```

A keretrendszer képes egyné több sorozat megjelenítésre (bal oldalt két személyt jelölnek ki, és job oldalon a diagramban a két személy mint két sorozat lesz megjelenítve a diagramban). A

sorozat azonosítója. (pl PeopleID)

```
<SeriesColumn>Series</SeriesColumn>
```

A sorozathoz megjelenő név (PeopleName).

```
<CategoriesColumn>Category</CategoriesColumn>
```

Az X tengely osztásai (pl. a hét száma)

```
<DataColumns>
  <DataColumn ChartType="areaspline" Caption=" - Valószínűség">Data2</DataColumn>
  <DataColumn ChartType="spline" Caption=" - Aktivitás">Data</DataColumn>
</DataColumns>
```

Az adatokat tartalmazó DDO beli oszlop. Több ilyen is lehet, ebben az esetben egy sorozathoz több diagram is tartozhat. A ChartType attribútum adja meg a diagram típusát, a Caption attribútummal pedig megkülönböztető végződést lehet adni. Ezek az attribútumok nem kötelezőek.

A ChartType lehet:

- areaspline
- spline
- line
- area
- bar
- pie
- column

```
<AutoRefresh>1</AutoRefresh>
```

Bekapcsolja az automatikus frissülést. A Beírt érték percben értendő.

```
<xAxis>
  <LabelRotation>-45</LabelRotation>
</xAxis>
```

Lehetőség van a tengelyen megjelenő szövegek elforgatására. Az yAxis tag az y tengely opcióit tartalmazza.

ChainListNode

Id. DGVC

Példa

```
<?xml version="1.0" encoding="iso-8859-2" ?>
<ChartForm>
  <Name>ChartFormProjectActivity</Name>
  <Caption>Ügylet aktivitás</Caption>

  <DDO>DDChartProjectActivity</DDO>

  <ChartType>areaspline</ChartType>
  <Legend position="Bottom">true</Legend>
  <TopText />
  <BottomText />
  <xAxis>
    <LabelRotation>-45</LabelRotation>
  </xAxis>
  <SeriesIDColumn>SeriesID</SeriesIDColumn>
  <SeriesColumn>Series</SeriesColumn>
  <CategoriesColumn>Category</CategoriesColumn>
  <DataColumns>
    <DataColumn ChartType="areaspline" Caption=" - Valószínűség">Data2</DataColumn>
    <DataColumn ChartType="spline" Caption=" - Aktivitás">Data</DataColumn>
  </DataColumns>
  <ChainListNodes>
    <ChainListNode>
      <ClickedColumn>Value</ClickedColumn>
      <KeyColumn type="Project_ID">SeriesID</KeyColumn>
      <LinkMM>MMFolyamatok</LinkMM>
      <LinkScreen>ScreenProject</LinkScreen>
      <LinkComponent>ComponentProjectsEdit</LinkComponent>
    </ChainListNode>
  </ChainListNodes>
</ChartForm>
```

Node-ok

Tartalomjegyzék

Extra funkciók

PRS

Az elnevezés a ProcessResultSet funkció rövidítése. Röviden a lényege: a Griden látható adatokkal, a kompsone bejövő, és onnan kimenő filterekkel, a szűrő PSearch-ök értékeivel, illetve az alap UserSession adatokkal végzett tetszőleges művelet. A műveleteket egy SQL tárolt eljárásban tudjuk definiálni. A PRS funkció folyamata röviden a következő:

1. a paraméterezésnek megfelelően a grid szükséges tartalmából egy XML formátumot generál, melyet eltárol egy adatbázis táblában. Ugyanide kerülnek be a filterek is, a szűrő és usersession értékek is. A beszűrés után a tábla egyedi azonosítóját megjegyzi, használja
2. az előzőekben megjegyzett egyedi azonosítóval hívja meg a paraméterezésben megadott tárolt eljárást
3. a tárolt eljáráson belül az előzőleg kötött XML formátumban elmentett adatokkal tetszőleges műveletek végrehajthatóak (pl. módosítás, listák létrehozása, tömeges feladatok, kampányok létrehozása, körlevelek adatainak előkészítése, stb.)
4. a tárolt eljárás a hibamentes futás végén opcionálisan visszaad (SELECT utasítással) egy tetszőleges visszatérési értéket, melyet a rendszer megjelenít. (pl. „A lista sikeresen kibővítve 42 darab új elemmel!”)

Egy példán keresztül bemutatva egy lehetséges működése a PRS-nek. Ettől természetesen funkcionálisan és a keretrendszer által megengedett működési szintekben is el lehet térni. Ez egy kötött feladat, nem feltétlenül iránymutató – lévén ez az egyik legszabadabb illetve sokoldalú funkciója az effector rendszernek.

Adott egy grid, mely mutatja a bal oldali objektumhoz (rendezvény) kapcsolt meghívott személyek listáját.

Ezen a griden van egy PSearch, az alábbi paraméterezéssel:

```
<PSearch id="AddToList" displaytype="ActionButton" actiontype="Link">
```

```

<LinkMM>MMPEP</LinkMM>
<LinkScreen>ScreenMarcomRendezvények</LinkScreen>
<LinkComponent></LinkComponent>
<Caption>Személy hozzáadása</Caption>
<Action RefreshScreenAfterAction="true">SimpleLink</Action>
</PSearch>

```

Ez itt önmagában még nem egy PRS akció, azonban szorosan hozzá tartozik ebben a paraméterezésben. Ahogyan látható is, itt az Action = "SimpleLink", ami annyit tesz, hogy a LinkScreen-ben megadott Screen-t fogjuk látni a kattintás után. Látható továbbá az is, hogy a RefreshScreenAfterAction attribútum értéke true, ami annyit jelent, hogy miután a felhözött Screen-en paraméterezett PRS akció lefutott, az után az a grid, ahonnan a SimpleLink hívás volt, be fog frissülni. Nyilvánvalóan ez itt azért kell, mert a PRS akció ennek a gridnek az adattábláján fog módosítást végrehajtani, amit azonnal látni is szeretnénk (megívtak hozzáadása a listához).

A felhasználó miután kattintott erre a PSearch-re, látni fogja a Screen-t, melyen van egy Grid, amin van egy PSearch, amelynek a paraméterezése az alábbi:

```

<PSearch id="PRS_1" displaytype="ActionButton" actiontype="ProcessResultSet">
<LinkMM>MMPEP</LinkMM>
<LinkScreen />
<LinkBusinessObject />
<LinkComponent />
<Caption>Listához hozzáad</Caption>
<Action sp="osp_prs_sp1 " CloseScreenAfterAction="true" RefreshScreenAfterAction="true" OnlySelectedRows="true">ProcessResultSet</Action>
</PSearch>

```

Nézzük mi itt az érdekes? actiontype = "ProcessResultSet" = Action, illetve az Action tag attribútumai között szerepel az összes jelenleg beállítható érték:

- sp: az sql tárolj eljárás neve, ami meg lesz hívva
- CloseScreenAfterAction: miután a teljes PRS folyamat lefutott, zárja –e be a screen-t, vagy ne. Miután a jelenlegi Screen amin dolgozunk egy SimpleLink hívással került eléln, itt ezt érdemes bezárni.
- RefreshScreenAfterAction: a neve beszédes, a PRS folyamat lefutása utáni grid frissítésre utal. Ez itt valójában egy felesleges lépés, mivel mindenképpen be lesz zárva a Screen az előző attribútum miatt.
- OnlySelectedRows: mely adatok kerüljenek át a PRS –hez feldolgozásra? Csak a kijelöltek, vagy a teljes Grid tartalma, amit pillanatnyilag látunk? Itt most csak a kijelöltekre van szükségünk, tehát az érték true.

Jön a lényeg: a felhasználó megnyomja a „PRS_1” ID-val megjelölt ActionButton-t. Innentől a fenti felsorolás alapján írom le a működést, konkrét adatokkal, illetve az itteni eljárásokból kivett részletekkel.

XML formátum elkészítése, tárolása.

A kattintás után a ProcessResultSet nevű rendszertáblában készül egy új rekord. Ennek a táblának van egy „Data” nevű, XML típusú oszlopa, ide fog bekerülni a legenerált XML. A ProcessResultSetID mező tartalma (mely itt az egyedi azonosító) kerül megjegyzésre. Nézzük, mi kerül be a Data cellába ebben az új rekordban?

```

<Grid>
<Rows>
<OneRow>
<CompanyName>Company_1899</CompanyName>
<PeopleName>People_1932</PeopleName>
<Street>Street_1932</Street>
<Zip>1134</Zip>
<City>Budapest</City>
<Email>Email_1932</Email>
<Jobtitle>gazdasági főigazgató-helyettes</Jobtitle>
</OneRow>
</Rows>
<ornFilters>
<ornInFilter>
<ornAlias>Event</ornAlias>
<ornType>Parameter</ornType>
<ornValue>24310</ornValue>
</ornInFilter>
<ornOutFilter>
<ornAlias>Company_ID</ornAlias>
<ornType>Param</ornType>
<ornValue>1899</ornValue>
</ornOutFilter>
</ornFilters>
<ornPSearches>
<ornPSearch>
<ornID>CompanyText</ornID>
<ornValue />
</ornPSearch>
</ornPSearches>
<UserSession>
<UserID>11</UserID>
<UserName>admin</UserName>
</UserSession>
</Grid>

```

A séma elég beszédes, nem foglalkozunk vele itt sokat. Látszódik hogy egy darab rekord volt kiválasztva, egy bejövő és egy kimenő filter van, egy szűrő aminek a neve arról árulkodik hogy cégnévre van kötve, érték nincs benne, illetve az alap UserSession adatok.

2-3-4. A megjegyzett ProcessResultSetID –val a paraméterezésben megadott tárolt eljárás meghívása. Az eljárásban ezzel a bejövő paraméterrel kapásból elérjük az eltárolt XML adatokat. Érdemes azzal kezdeni, hogy átalakítjuk az XML struktúrát táblává, így a továbbiakban sokkal könnyebb lesz kezelni az adatokat. Ehhez egy segéd függvényt csatoltam a fájlhoz (a neve: dbo.XMLTable), melynek a kommentjeiben van magyarázat. Ennél a függvénynél figyelni kell rá, hogy ez túl általános, és elég nehézkes is a használata. Fontos, hogy ez lassú is tud lenni nagyobb mennyiségű adatra, ezzel csak 10-es nagyságrendű rekordszámot érdemes kezelni. Ennél robusztusabb műveletekhez érdemes készíteni egy saját XML feldolgozó függvényt, erre a T-SQL sok lehetőséget ad.

Vázlat egy tárolt eljárásból, kommentekkel:

```

CREATE PROCEDURE [dbo].[osp_prs_sp1]
    @id INT
AS BEGIN
    SET NOCOUNT ON
    SET ANSI_WARNINGS OFF

    DECLARE @xml XML

    --lekérjük egy változóba az XML-ek tartalmát
    SET @xml = (SELECT TOP 1 data FROM ProcessResultSet WITH(NOLOCK) WHERE ProcessResultSetID = @id)

    --elkészítünk egy temp táblát, melynek a sémája az XML feldolgozó függvényünk visszatérési sémájával azonos
    CREATE TABLE #tmp_xml (
        -- MEZŐ DEFINÍCIÓK--
    )

    -- az elkészített temp táblába betesszük a feldolgozott XML adatokat
    -- itt a 4 tárolt közül a megfelelőt kell lefuttatni, amire szükségünk van.
    -- Értelemszerűen hozzák psearch, filter, vagy grid adatokat, illetve az „all”

```

```

-- végződésű ömlesztve mindent
INSERT INTO #tmp_xml
execute osp_getPrsDataFromXml_Psearch @id
execute osp_getPrsDataFromXml_Filters @id
execute osp_getPrsDataFromXml_Rows @id
execute osp_getPrsDataFromXml_All @id
-- ha volt használható adatunk, akkor végrehajtjuk rajtuk a szükséges műveleteket, majd visszatérünk egy szabadon választott üzenettel
-- egyébként kiírjuk a szintén szabadon választott üzenetünket.
IF((SELECT Count(1) FROM #tmp_xml) <> 0)
BEGIN
-- MŰVELETEK VÉGREHAJTÁSA--
SELECT 'A művelete eredménye ....'
END
ELSE
SELECT 'Nem jelölt sort!'

DROP TABLE #tmp_psearch
END

```

A PRS-nek van egy további funkciója. Előfordulhat olyan felhasználói eset is, amikor muszáj „fej adatokat” megadni egy-egy PRS által elvégzett művelethez. Egy példa: a PRS-sel pl. listákat hozunk létre, arról hogy kiknek szeretnénk karácsonyi ajándékot küldeni az ügyfeleink közül, cégekre lebontva. Ilyenkor adja magát, hogy egy-egy listához legyenek a tartalmán kívül ilyen extra adatok, mint: cég, ki készítette és mikor a listát, stb. Erre lehetőségünk van, ha abban a PSearch-ben megadunk egy Screen-t is, ahonnan a PRS tárolt eljárást meghívjuk. Ez a Screen mutasson egy olyan EditForm-ra, ami egy PRS típust (akár altípussal együtt) hoz létre. Ez konkrétan a ProcessResultSet táblát használja ilyenkor a BusinessObject! Az fog történni, hogy a ProcessResultSet táblába beszúrás a BusinessObject fogja elvégezni az EditForm-os mentés után, majd az így megkapott azonosítóval dolgozik tovább a rendszer, innentől ugyanaz a dolgok menete.

PRS ChainListNode-ban

```

<PSearches>
<PSearch id="mentes" displaytype="ActionButton" actiontype="Link">
<LinkMM></LinkMM>
<LinkScreen></LinkScreen>
<LinkBusinessObject></LinkBusinessObject>
<LinkComponent></LinkComponent>
<Caption>Mentés</Caption>
<Action sp="osp_wrk_ShowAdvocacyQuestionAnswers" CloseScreenAfterAction="false" RefreshAfterAction="screen">ProcessResultSet</Action>
<Visible>false</Visible>
</PSearch>
</PSearches>

<ChainListNodes>
<ChainListNode type="ProcessResultSet">
<ClickedColumn>Checked</ClickedColumn>
<KeyColumn type="id">AnswerID</KeyColumn>
<LinkMM></LinkMM>
<LinkScreen></LinkScreen>
<LinkBusinessObject></LinkBusinessObject>
<LinkComponent></LinkComponent>
<ProcessResultSetPSearch>mentes</ProcessResultSetPSearch>
</ChainListNode>
</ChainListNodes>

```

A ProcessResultSetPSearch-ben adott nevet használjuk a Psearch id-jaként.

A felületen üzenetet is meg lehet jeleníteni. Ez effector az első select első sorának első oszlopába írt értéket jeleníti meg, ha nincs Message nevű oszlop (ugyanis akkor annak a tartalmát mutatja meg az Effector). Ha a tárolt nem ad eredményül semmit, vagy a megadott oszlopban null van, akkor nem jelenít meg semmilyen üzenetet az Effector,

Lehetőség van PRS hívás után megnyitni egy képernyőt. Ilyenkor a tárolt eljárás a végén abban a Select mondatban több oszlop értéket ad vissza. A funkcióhoz használható oszlop nevek.

LinkMM: a megnyitni kívánt képernyőhöz MM név (MMPEP)

LinkScreen: a megnyitni kívánt képernyő neve (ScreenEvent)

FilterValues: a filter értékek json formátumban ([{"Alias": "Event", "Type": "Out", "Value": 108276 }, { "Alias": "Template_Event", "Type": "Param", "Value": "ISO2011_070FejFejITeszt"}])

EmbeddedWebDisplay

Egy panelen egy valamilyen szabály alapján generált url által hivatkozott tartalmat lehet megjeleníteni benne. Az xml fájlok a az XML csomag EmbeddedWebDisplay könyvtárában találhatóak meg. A fájlnev minden esetben EWD-vel kell kezdődnie.

Példa

```

<?xml version="1.0" encoding="ISO-8859-2"?>
<EmbeddedWebDisplay>
<Name>EWDMap</Name>
<Caption />
<URL type="SQL" return="string" default="">
<CDATA[
SELECT 'http://maps.google.com/maps?q=' + (SELECT top 1 isnull(city,') + ', ' + isnull(street,') + ' ' + isnull(StreetNumber,') from company with (nolock) where CompanyID = {##Filter.Company_ID##})
]]>
</URL>
</EmbeddedWebDisplay>

```

Magyarázat

<Name>...</Name>

Mindig a fájl neve mínusz a kiterjesztés

<URL type="SQL" return="string" default="">...</URL>

Ez egy RuleValueType. Megjeleníteni kívánt oldal URLjét lehet vele dinamikusan előállítani.

Fájl feltöltéshez vírus írtás

A rendszer a clamav-t használja, web.config fájlban kapcsolható és állítható be a funkció Ha clamav beállítás true, akkor be van kapcsolva a funkció.

Node-ok

Tartalomjegyzék

Többnyelvűség beállítása

Minden elem, ami a felületen megjelenik adatbázisból paraméterezhető. A tábla neve: FSYS_LabelLookup.

Fontosabb oszlopok: TableName, FieldName, Lang, Label, Deleted

A Lang oszlopba az adott nyelv két jegyű azonosítója kerül pl. HU, EN, RU, DE....

A TableName és a FieldName oszlopok. Az XML fájlban egy oszlop címkéje ilyen formában adható meg [##TableName.FieldName##]. Ilyenkor a rendszer a bejelentkező felhasználó által kiválasztott nyelvet használva visszakeresi az FSYS_LabelLookup táblából a kért címkét úgy, hogy a . mentén ketté bontja az XML-ben található hivatkozást TableName és FieldName elemekre.

A képernyőn megjelenő gombok (nem paraméterezett elemek, hanem amit a keretrendszer kitesz) és a hibaüzenetek (nem a szabályok hibaüzenetei, hanem a rendszer üzenetek) a táblában úgy vannak letárolva, hogy a TableName értéke \$SYSTEM\$, a FieldName egy a rendszerben használt név. Nincs baj, ha nincs meg a címke, mert a kódban van rá alapértelmezett érték (és magyar nyelvű).

Szótár szerű fordítás is megoldható (leginkább a kártyán vagy táblázatban megjelenő elemekre vonatkozik, de combo is így többnyelvűsíthető). Ilyenkor a TableName oszlopba \$PHRASE\$ értéket kell írni, a ovw_FSYS_Phrase nézetet lehet felhasználni.

Pl. van egy combobox amelyben igen és nem értékeket akarunk többnyelvűen megjeleníteni, akkor a lekérdezés így néz ki:

```
select ll.LookupListID, ISNULL(v.Value, ll.LookupValue) AS LookupValue
FROM fsys_lookuplist ll
left join ovw_FSYS_Phrase v on v.[Key] = ll.LookupValue AND v.Lang = 'EN'
where ll.groupnum = 5
```

A nyelvre a szabályokban és egyéb helyeken a [##Session.Language##] értékkel lehet hivatkozni.

DBPatch fájlok:

- 20130711_CREATE_ovw_FSYS_Phrase.sql
- 20130712_MultiLanguage_changes.sql
- 20130730_MultiLanguage_data.sql

A választható nyelvek listáját a Connection fájlban lehet megadni (Connection tagon belül, amelyek az első az a default)

```
<Languages>
<Language code="en-US">english</Language>
<Language code="hu-HU">magyar</Language>
</Languages>
```

Node-ok

Tartalomjegyzék

Üzleti objektum

Előszó

Az üzleti objektumok a rendszerben egy adattáblát, vagy egy adatbázis nézet kezelését valósítják meg, részt vesznek a folyamatok definiálásban, illetve további adatbázis műveleteket írhatnak le. A fájlokat a BusinessObject könyvtárba kell menteni. A fájlnevének tükröznie kell az objektum nevét, és altípusát. A tábláknak tartalmaznia kell Deleted TINYINT mezőt, ami a rendszer tranzakció-kezeléséhez kell. Ha BO-hoz altípust is szeretnénk definiálni, akkor kell egy ObjectType VARHCAR(60) oszlop is.

A főtag

```
<BusinessObject mainbo="" Name="People">...</BusinessObject>
```

Az üzleti objektum egyedi nevét a Name attribútum tartalmazza. Ebben az esetben a fájl nevének BusinessObjectPeople.xml kell, hogy legyen.

```
<BusinessObject mainbo="People" Name="Kontakt">...</BusinessObject>
```

A mainbo attribútum a származtatást jelöli. Ha ez a tag ki van töltve, akkor ez azt jelenti, hogy az üzleti objektum egy másik üzleti objektumból származik, kibővíti annak jelentését, működését. A fájl nevének is tükröznie kell ezt a beállítást, ebben az esetben BusinessObjectPeopleKontakt.xml kell, hogy legyen. Itt a BusinessObjectPeople bo van kibővíve, megjelölve. Nem lehet két különböző BusinessObject ugyazzal a névvel.

Altagek

```
<DefaultTable>People</DefaultTable>
```

Az üzleti objektum adatbázisban lévő tábláját határozza meg. Ez a beállítás gyakran elmarad származtatott objektumok esetén (nincs kiegészítő adat).

```
<UniqueID>PeopleID</UniqueID>
```

A DefaultTable beállításban jelzet tábla egyik oszlopa, amely a sor egyedi azonosításhoz kell. Származtatott üzleti objektumok esetén ez jellemzően U_ szokott lenni. A származtatott BO-nak a UniqueID oszlop neve nem lehet azonos a fő üzleti objektum UniqueID oszlop nevével. Ha a származtatott üzleti objektumnak nincs DefaultTable beállítása, akkor ez a beállítás is elhagyandó. A fő üzleti objektumok esetén ez a beállítás kötelező, és jellemzően autoincrementes int típusú a mező. Az egyedi azonosító nem lehet összetett kulcs.

RoleConnection

```
<RoleConnections>...</RoleConnections>
```

Egy üzleti objektum kapcsolatait írja le a tag. Ezek a kapcsolatok a mentési, törlési, adott esetben lekérdezési folyamatokat befolyásolják. Ez a tag egy gyűjtője a kapcsolatoknak, amelyek további tag-ekben vannak leírva. Egy kapcsolat a tag-ben van leírva.

```
<RoleConnection>...</RoleConnection>
```

Attribútumai:

- type: értéke lehet
 - Relation: A kapcsolat egy adatbázisban megszokott kapcsolatra utal. A kapcsolat egy gyerek-szülő viszonyt határoz meg.
 - Descent: A kapcsolat egy származtatást jelöl. A kapcsolat egy olyan üzleti objektumra mutat, amely az aktuális üzleti objektumot bővíti ki, a kapcsolat révén, további mezőkkel.
- direction: A kapcsolat irányát jelzi. A kapcsolat lehet Parent vagy Child. Ha Parent, akkor az üzleti objektumban a kapcsolat által meghatározott tábla egyedi azonosítója szerepel, mint külső kulcs. Ha Child, akkor az üzleti objektumhoz kapcsolódó táblában szerepel az adott üzleti objektum egyedi kulcsa, mint külső kulcs.
- roleName: A kapcsolat elnevezése. A kapcsolatnak egyedi névvel kell rendelkeznie, a RoleConnections tag-en belül.
- deletingmode: Az objektum törlési módját határozza meg. Lehetséges értékei AllowIfEmpty vagy Cascade. Ha az érték AllowIfEmpty, akkor az objektum csak akkor törölhető, ha nincs az adott kapcsolatban leírt táblában gyerek rekordja. Ha az értéke Cascade, akkor az objektum törlésekor a gyerek táblák azon sorai is törölődnek, amelyek erre az üzleti objektumokra mutatnak. (Ez lehet, hogy már nem működik jól....)

```
<ToMinimum>...</ToMinimum>
```

A kapcsolat számosságát határozza meg. A lehetséges értékek halmaza a következő: 0, 1, N. A kapcsolt táblában levő adatok számosságának minimumát jelöli.

```
<ToMaximum>...</ToMaximum>
```

A kapcsolat számosságát határozza meg. A lehetséges értékek halmaza a következő: 0, 1, N. A kapcsolt táblában levő adatok számosságának maximumát jelöli.

```
<FromMinimum>...</FromMinimum>
```

A kapcsolat számosságát határozza meg. A lehetséges értékek halmaza a következő: 0, 1, N. A táblában levő adatok számosságának minimumát jelöli.

```
<FromMaximum>...</FromMaximum>
```

A kapcsolat számosságát határozza meg. A lehetséges értékek halmaza a következő: 0, 1, N. A kapcsolt táblában levő adatok számosságának maximumát jelöli.

```
<ConnectionIDField>EventID</ConnectionIDField>
```

A kapcsolatot meghatározó mező neve. Az aktuális üzleti objektum az egyedi azonosítójával fog az itt megadott oszlophoz kapcsolódni.

```
<BusinessObject>...</BusinessObject>
```

A kapcsolat üzleti objektuma. Ezen keresztül határozódik meg a kapcsolatban szereplő tábla.

```
<ConnectionTableName>...</ConnectionTableName>
```

Ha a kapcsolat egy kapcsolótábla, akkor itt a kapcsolótábla nevét kell megadni.

```
<ConnectionTableID name="EventID">EventID</ConnectionTableID>
```

A két táblát összekötő mezőket tartalmazza. Ha több oszlop is szerepel a kapcsolatban, akkor több ConnectionTableID tag-nek kell szerepelnie. A tag értéke, azaz oszlop, amely ebben a BusinessObjectban a kapcsolatban részt vesz, míg a name attribútum a kapcsolótáblában levő oszlop neve.

```
<Alias Values>  
<Alias Value name="EventRoleID" type="SQL" default="1" return="int">1</Alias Value>  
</Alias Values>
```

Lehetőség van virtuális oszlopokat definiálni a BusinessObject-hez, Isd Virtuális mező szakasz.

Mezők listája (Fields, Field)

```
<Fields>...</Fields>
```

Az üzleti objektum oszlopainak gyűjtő tag-je. Az oszlopok a tag-ben vannak definiálva.

```
<Field>...</Field>
```

Az oszlop adatbázisbeli neve a name attribútum tartalmazza, ugyanakkor a name attribútum olyan oszlop nevet is definiálhat, amely fizikailag nincs az üzleti objektum táblájában, hanem egy másik táblában szerepel, mint kapcsolóadat. Ilyenkor a két tábla között az 1:1 kapcsolat követelmény, és alkalmazni kell két másik attribútumot is aliascolumn, aliasrolename. Az aliasrolename annak a RoleConnection neve, amely ezt a kapcsolatot leírja, az aliascolumn pedig a kapcsolt táblában található oszlop neve, amely az értéket tartalmazza. Elemei:

- DefaultValue: új létrehozáskor a mező alapértékét határozza meg. Ez egy RuleValueType szabály (Isd. RuleValueType.doc)

Folyamattal kapcsolatos tag-ek (Triggers, Trigger)

Adhoc eseményeknél ezek nem futnak le. Ha SQLt akrasz futtatni mondjuk mentés vagy leokézás után, akkor használd a Methods AfterSave lehetőséget

```
<Triggers>...</Triggers>
```

Az üzleti objektumok triggerjeinek gyűjtő tag-je.

```
<Trigger>...</Trigger>
```

A triggerek a folyamatlépéseket vezérlik, meghatározzák, hogy egy üzleti objektumban történt esemény hatására mi történjen, mi jöjjön létre. A triggerek meghatározhatják, hogy az adott üzleti objektum milyen más üzleti objektumokban történt változásokra készüljön el. Például, ha készen állítunk egy eseményt, az kiválthatja egy újabb folyamatlépés létrejöttét, vagy egy másik folyamatlépés állapotának beállítását, vagy egy tetszőleges SQL utasítás végrehajtását is.

Attribútumai:

- businessObject: Az az üzleti objektum, amely a triggert aktiválni fogja. (lehet Event, Project)
- template: Az üzleti objektum template-je. A BusinessObject és a template határozza meg pontosan a kiváltó objektumot. (PI: businessobject: Event, template: MegrendelesBekotes, akkor a kiváltó objektum a BusinessObjectEventMegrendelesBekotes, amely a BusinessObjectEventMegrendelesBekotes.xml állományban van leírva.
- type, meghatározza a kiváltott esemény típusát, ezek a típusok a következő értékeket vehetik fel
 - Create: Létrejön az új üzleti objektum
 - Modify: Az objektum módosítva lesz
 - Delete: Az objektum törölve lesz (sohasem használtuk, talán nem is működik)
 - RunSQL: Egy SQL utasítás lesz futtatva
- Group: Ha 0-tól eltérő az értéke, akkor a triggereket csoportba fogja ez az attribútum. Így egy esemény létrejöttét, több kiváltó eseményhez köthetjük. Az azonos group számmal rendelkező triggerek alkotnak egy csoportot. Ennek az értéke jelenleg mindig 0.
- jei:
- TriggerEvent: Az az esemény, amely kiváltja a trigger aktiválását
- SQL: A RunSQL trigger típus esetén futtatandó SQL utasítás. Az elem rendelkezik egy type attribútummal, amely jelenleg nincs felhasználva.
- Condition: A trigger aktiválódása esetén a condition feltétel teljesülése esetén hajtja végre a Type-ban megadott akciót.
- SetPropertyies: Ez a tag a Modify típusú trigger attribútum esetén határozza meg, hogy az üzleti objektum mely tulajdonságai, mire változzanak. Ez a tag, egy összetett elem, amelynek a felépítése a következő:

```
<SetProperties>  
<Property name="oszlop neve">  
  <Value type="típus">érték</Value>  
</Property>  
</SetProperties>
```

Az új érték nem lehet kifejezés. A type-ban meghatározott típusnak kell lennie, amely szöveges adatnak közvetlenül konvertálhatónak kell lennie a megadott C# adattípusra.

Validálás, extra műveletek paraméterezése

```
<Methods>  
<AfterSave>  
  <ClassName>X</ClassName>  
  <Module>Y</Module>  
  <Action>as</Action>  
</AfterSave>  
<BeforeSave>  
  <RunSQL><![CDATA[  
IF ISNULL(['##Filter.Event##'],)=""  
SELECT "  
ELSE BEGIN  
INSERT INTO orn_Event_Hivatkozas (EventID, ProjectID, Created, CreatedByID, Deleted)  
VALUES (['##Filter.Event##'], ['##Field.ProjectID##'], GETDATE(), ['##Session.UserID##'], 0)  
END  
]]>  
</RunSQL>  
</BeforeSave>  
<AfterDelete>  
  <RunSQL>SELECT ['##Field.ProjectID##']</RunSQL>  
</AfterDelete>
```

```
<OnValidate type="SQL" return="string" default="">SELECT [##Field.ProjectID##]</OnValidate>
</Methods>
```

AfterSave

Mentés után fusson le valami. Ez lehet SQL, ebben az esetben a RunSQL tag-ek közé tett SQL mondat lesz lefuttatva. Lehet dll hívás is, ebben az esetben a ClassName, Module, Action tag-eket kell használni. Module: dll fájl, ClassName az osztály neve, az Action pedig kijelöli, hogy melyik részfunkciót akarjuk meghívni (egy osztályban több funkció megírható, ezzel lehet kiválasztani, hogy melyiket akarjuk meghívni)

BeforeSave

Mentés előtt fusson le valami. Nincs lehetőség beavatkozni a mentési folyamatba (vagyis nincs „mégsem” lehetőség). Ez lehet SQL, ebben az esetben a RunSQL tag-ek közé tett SQL mondat lesz lefuttatva. Lehet dll hívás is, ebben az esetben a ClassName, Module, Action tag-eket kell használni. Module: dll fájl, ClassName az osztály neve, az Action pedig kijelöli, hogy melyik részfunkciót akarjuk meghívni (egy osztályban több funkció megírható, ezzel lehet kiválasztani, hogy melyiket akarjuk meghívni)

AfterDelete

Törlés után fusson le valami. Ez lehet SQL, ebben az esetben a RunSQL tag-ek közé tett SQL mondat lesz lefuttatva. Lehet dll hívás is, ebben az esetben a ClassName, Module, Action tag-eket kell használni. Module: dll fájl, ClassName az osztály neve, az Action pedig kijelöli, hogy melyik részfunkciót akarjuk meghívni (egy osztályban több funkció megírható, ezzel lehet kiválasztani, hogy melyiket akarjuk meghívni)

OnValidate

Ez egy RVT. Ide is lehet érvényesség szabályt definiálni, mentés előtt a rendszer elvégzi ennek a kiértékelését. Ha a szabály üres szöveget ad vissza, az azt jelenti, valid és a mentés megtörténhet. Amúgy meg a figyelmeztető üzenettel kell visszatérnie.

Virtuális mező

A példában az Event BusinessObject paraméterezése van. Ide vonatkozó részek:

```
<RoleConnection type="Relation" deletingmode="Cascade" direction="Child" RoleName="EventPeople">
  <ToMinimum>0</ToMinimum>
  <ToMaximum>1</ToMaximum>
  <FromMinimum>0</FromMinimum>
  <FromMaximum>1</FromMaximum>
  <BusinessObject>BusinessObjectPeople</BusinessObject>
  <ConnectionTableName>EventPeople</ConnectionTableName>
  <ConnectionTableID name="EventID">EventID</ConnectionTableID>
  <ConnectionTableID name="PeopleID">Felelos</ConnectionTableID>
  <AliasValues>
    <AliasValue name="EventRoleID" type="SQL" default="1" return="int">1</AliasValue>
  </AliasValues>
</RoleConnection>
<Field name="Felelos" aliascolumn="PeopleID" aliasrolename="EventPeople">
  <DefaultValue type="Simple" return="int">[##Session.UserID##]</DefaultValue>
</Field>
```

Mire jó? Egy felhasználási esetben keresztül röviden: A paraméterezés folyamán kialakított két fogalom: esemény, és esemény felelőse kapcsán

- az eseményhez egy külön táblában tárolunk egy vagy több résztvevőt
- így ez egy 1..n adatkapcsolat az Event, és a résztvevőket tartalmazó EventPeople tábla között
- a résztvevők között szigorúan egy darab kiemelt ember van, ami az EventPeople.EventRoleID mező alapján van megkülönböztetve (minden különböző EventRoleID egy-egy szerepkörnek felel meg, ezek közül a „Felelős” az, ami most a megkülönböztetett)
- ahhoz, hogy a BO ezt le tudja kérni, azaz összeállítsa a megfelelő SQL selectet hozzá, illetve mentésnél tudja módosítani, a RoleConnection-t tudjuk segítségül hívni, a példában látható módon. Az Field tag aliasrolename mondja meg, hogy melyik roleconnection mondja meg a kapcsolat táblának az adatát (amelyben a Felelos mezőhöz tartozó rétket ki lehet venni), az aliascolumn attribútum mondja meg, hogy mi az oszlop neve a kapcsolt objektumban. A hivatkozott roleconnectionben le kell írni a kapcsolatot. Az AliasValues tag-ban felsorolt mezők értékét mire állítsa be. A jelen példában az Event létrehozásakor létrejön egy EventPeople sor is, amiben az EventID értéke az Event EventID oszlopának az értéke lesz, a PeopleID oszlop értéke a virtuális Felelos nevű oszlop lesz, az EventRoleID értéke pedig 1 (felelős szerep)

Példa

```
<?xml version="1.0" encoding="iso-8859-2"?>
<BusinessObject Name="Event">
  <Caption>Esemény</Caption>
  <DefaultTable>Event</DefaultTable>
  <UniqueID>EventID</UniqueID>
  <RoleConnections>
    <RoleConnection type="Relation" direction="Child" RoleName="EventDocument">
      <ToMinimum>0</ToMinimum>
      <ToMaximum>N</ToMaximum>
      <FromMinimum>0</FromMinimum>
      <FromMaximum>N</FromMaximum>
      <BusinessObject>BusinessObjectDocument</BusinessObject>
      <ConnectionTableName>EventDocument</ConnectionTableName>
      <ConnectionTableID name="EventID">EventID</ConnectionTableID>
      <ConnectionTableID name="DocumentID">DocumentID</ConnectionTableID>
    </RoleConnection>
    <RoleConnection type="Relation" direction="Child" RoleName="EventNotelD">
      <ToMinimum>0</ToMinimum>
      <ToMaximum>1</ToMaximum>
      <FromMinimum>0</FromMinimum>
      <FromMaximum>N</FromMaximum>
      <ConnectionIDFields>EventID</ConnectionIDFields>
      <BusinessObject>BusinessObjectEventNotes</BusinessObject>
    </RoleConnection>
    <RoleConnection type="Relation" deletingmode="Cascade" direction="Child" RoleName="EventPeopleN">
      <ToMinimum>0</ToMinimum>
      <ToMaximum>N</ToMaximum>
      <FromMinimum>0</FromMinimum>
      <FromMaximum>N</FromMaximum>
      <BusinessObject>BusinessObjectPeople</BusinessObject>
      <ConnectionTableName>EventPeople</ConnectionTableName>
      <ConnectionTableID name="EventID">EventID</ConnectionTableID>
      <ConnectionTableID name="PeopleID">PeopleID</ConnectionTableID>
      <ConnectionTableID name="EventRoleID">EventRoleID</ConnectionTableID>
    </RoleConnection>
    <RoleConnection type="Relation" deletingmode="Cascade" direction="Child" RoleName="EventPeople">
      <ToMinimum>0</ToMinimum>
      <ToMaximum>1</ToMaximum>
      <FromMinimum>0</FromMinimum>
      <FromMaximum>1</FromMaximum>
      <BusinessObject>BusinessObjectPeople</BusinessObject>
      <ConnectionTableName>EventPeople</ConnectionTableName>
      <ConnectionTableID name="EventID">EventID</ConnectionTableID>
      <ConnectionTableID name="PeopleID">Felelos</ConnectionTableID>
      <AliasValues>
```



```

<AliasValue name="EventRoleID" type="SQL" default="1" return="int">1</AliasValue>
</AliasValues>
</RoleConnection>
<RoleConnection type="Relation" direction="Parent" RoleName="ProjectID">
<ToMinimum>0</ToMinimum>
<ToMaximum>N</ToMaximum>
<FromMinimum>0</FromMinimum>
<FromMaximum>1</FromMaximum>
<ConnectionIDField>ProjectID</ConnectionIDField>
<BusinessObject>BusinessObjectProject</BusinessObject>
</RoleConnection>
<RoleConnection type="Relation" direction="Child" RoleName="Alfeladat">
<ToMinimum>0</ToMinimum>
<ToMaximum>1</ToMaximum>
<FromMinimum>1</FromMinimum>
<FromMaximum>1</FromMaximum>
<ConnectionIDField>EventID</ConnectionIDField>
<BusinessObject>BusinessObjectEventAlfeladat</BusinessObject>
</RoleConnection>
</RoleConnections>

```

```

<Fields>
<Field name="EventID">
</Field>
<Field name="CompanyID">
</Field>
<Field name="ProjectID">
</Field>
<Field name="StartDate">
<DefaultValue type="Constant" return="datetime">{##Special.Today##}</DefaultValue>
</Field>
<Field name="EndDate">
<DefaultValue type="Constant" return="datetime">{##Special.Today##}</DefaultValue>
</Field>
<Field name="CreatedByID">
<ReadOnly>true</ReadOnly>
<DefaultValue type="Simple" return="int">{##Session.UserID##}</DefaultValue>
</Field>
<Field name="Created">
<ReadOnly>true</ReadOnly>
<DefaultValue type="Constant" return="datetime">{##Special.Today##}</DefaultValue>
</Field>
<Field name="Description">
</Field>
<Field name="Task">
<DefaultValue type="Constant">true</DefaultValue>
</Field>
<Field name="Done">
<DefaultValue type="Constant">false</DefaultValue>
</Field>
<Field name="Felelos" aliascolumn="PeopleID" aliasrolename="EventPeople">
<DefaultValue type="Simple" return="int">{##Session.UserID##}</DefaultValue>
</Field>
<Field name="ObjectType">
<DefaultValue type="Constant" return="" default="false" />
</Field>
<Field name="Department">
<DefaultValue type="Constant">0</DefaultValue>
</Field>
<Field name="ParentEventID">
<DefaultValue type="Constant">0</DefaultValue>
</Field>
<Field name="RootEventID">
<DefaultValue type="Constant">0</DefaultValue>
</Field>
<Field name="RootEvent">
<DefaultValue type="Constant">false</DefaultValue>
</Field>
<Field name="PrivateEvent">
<DefaultValue type="Constant">false</DefaultValue>
</Field>
<Field name="AlreadyRead">
<DefaultValue type="Constant">false</DefaultValue>
</Field>
<Field name="WorkflowID">
<DefaultValue type="Constant">0</DefaultValue>
</Field>
</Fields>

```

Node-ok

Tartalomjegyzék