

Acțiuni asociate machetelor

Pentru a fi mai ușor de înțeles cum funcționează facilitățile de design în zona de acțiuni asociate cu machetele, vom porni de la exemple practice, enumerând enunțuri de probleme și exemplificând soluția de rezolvare.

Vă solicit toată îngăduința pentru exemplele care sunt uneori cam puerile, dar important e să înțelegeți funcționarea. Pe baza acestor exemple veți putea dezvolta singuri proceduri proprii, pe caz real, după nevoi.

Pe factura de ieșire vom asocia, ca cerințe, următoarele acțiuni (acceptați enunțurile banale fără să judecați!):

- 1- validare la salvarea facturilor, astfel încât fiecare articol să aibă clasa de articol asociată;**
- 2- atenționare dacă pe factură sunt linii cu Cantitate < 3;**
- 3- posibilitatea de a afișa o listă cu valorile facturate pe gestiuni;**
- 4- posibilitatea de a afișa o listă cu codurile externe și cele interne pentru fiecare articol din factură.**
- 5- import rezervare pe factura de ieșire.**

Pentru ultima problemă sunt necesare niște explicații suplimentare.

Presupunem că firma lucrează cu rezervări în domeniul turismului.

Rezervările conțin informații legate de servicii care urmează a fi prestate și despre pachetele de turism oferite.

Tabelele în care se vor salva datele sunt deschise la opțiunea "Machete proprii" și nu le voi explica aici.

Folosind opțiunea de design, macheta pentru introducerea datelor la rezervări este deja configurată și arată ca în imaginile de mai jos:

Rezervari Subunitate: Sediul Central

Operat Blocat Anulat

Organizator (plătitor)
Bob Miraculos SRL

Rezervare

REZ ✓
 Numar: RZ
 din data: 01.07.2023

Moneda: Lei Curs: ...

Date generale

Check in: 11.09.2023 Check out: 18.09.2023 Numar nopti: 7

Adulti: 2 Copii: Total persoane: 2

Transfer

Tip camera: Dubla

Regim masa: pensiune completa

Observatii transport:
Pickup in ordinea: Bulevardul Pacii 23, Strada Sperantei 22A

Componenta pachet

#	Nume	Titular	Data nasterii	Pasaport	Telefon	Email	CNP
1	Ionescu Clara	Da	19.09.2023			777 b@b.ro	0
2	Popescu Ion	Nu	19.09.2023		0		0

[Import X](#)

Rezervari Subunitate: Sediul Central

Operat Blocat Anulat

Organizator (plătitor)
Bob Miraculos SRL

Rezervare

REZ ✓
 Numar: RZ
 din data: 01.07.2023

Moneda: Lei Curs: ...

Date generale

Componenta pachet

Nr	Denumire serviciu	Pret	Cantitate	Valoare	Observatii
1	Taxa oras	500	1	500	plata la receptie
2	Vesta de salvare	3.000	1	3.000	plata inainte de aventura
3	Costi prezentare muzeu	100	2	200	plata la intrare in muzeu

Total: 3.700

Avans: 1.000

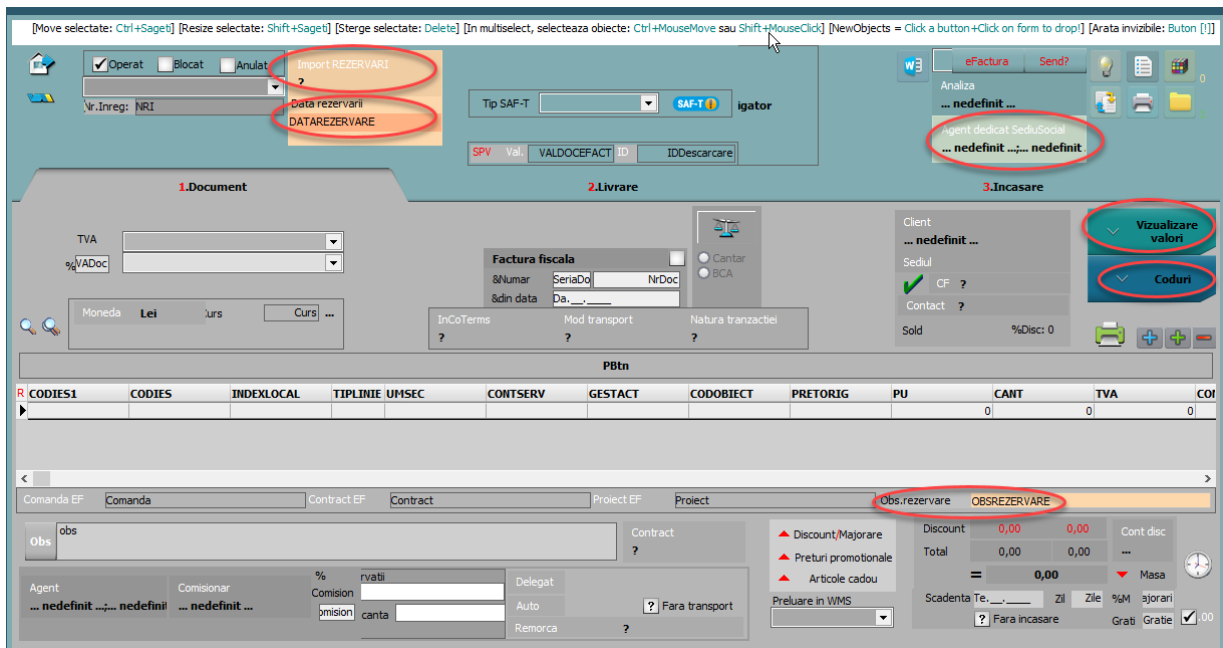
De achitat: 2.700

Buget [+](#) [-](#)

#	Plata	Tip Plata	Scadenta
1	2.000	Numerar	20.09.2023

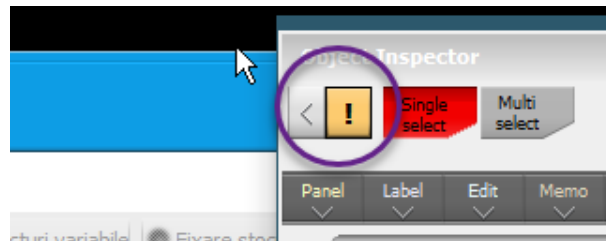
Taxe suplimentare

Pornim sa aplicăm aceste cerințe formulate mai sus pe factura de ieșire. Iată cum arată macheta de factură, cu toate elementele care rezolvă cerințele afișate la design:

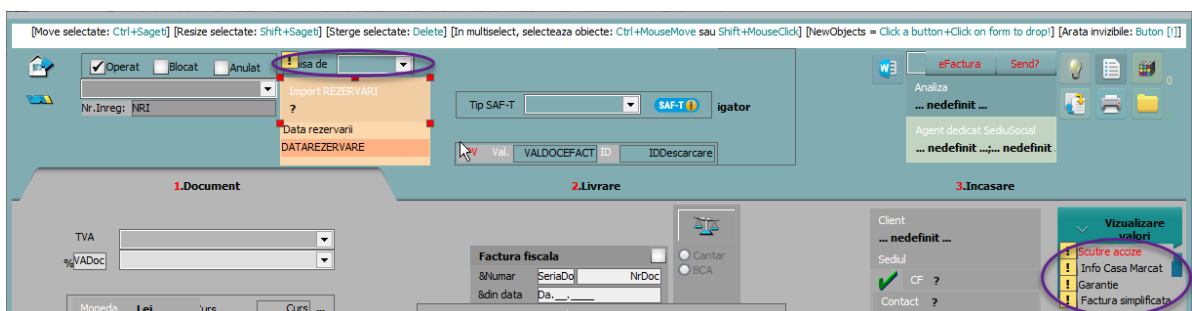


Elementele încercuite sunt controalele noi adăugate pe macheta originală. Pentru a le face loc, am ascuns alte controale care presupunem că sunt neutilizate de firmă.

Pentru a vizualiza ce controale am ascuns, apăș butonul "Show/Hide invisible", cel încercuit pe *Object inspector* în imaginea de mai jos:

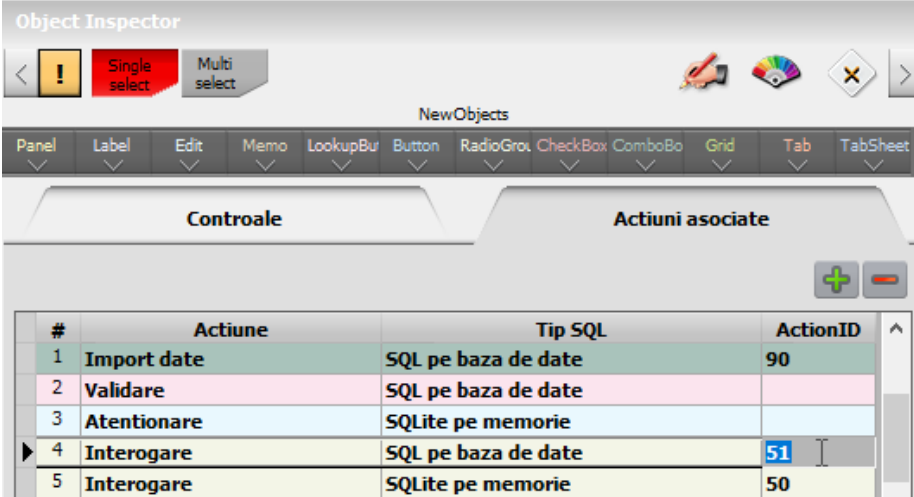


Efectul este următorul:



Observați controalele marcate cu pătrat galben la care s-a setat proprietatea *Visible = False*.

Rezolvarea punctelor 1..5 se face din design (F2 + buton "Design" pe o înregistrare nouă de factură), pagina "Actiuni asociate".



#	Actiune	Tip SQL	ActionID
1	Import date	SQL pe baza de date	90
2	Validare	SQL pe baza de date	
3	Atentionare	SQLite pe memorie	
4	Interogare	SQL pe baza de date	51
5	Interogare	SQLite pe memorie	50

Tipul de acțiune pentru cele 5 enunțuri este:

- 1 - **validare** (pentru validare la salvarea facturilor, astfel încât fiecare articol să aibă clasa de articol asociată);
- 2 - **atenționare** (pentru atenționare dacă pe factură sunt linii cu Cantitate < 3);
- 3 - **interogare** (pentru posibilitatea de a afișa o listă cu valorile facturate pe gestiuni);
- 4 - **interogare** (pentru posibilitatea de a afișa o listă cu codurile externe și cele interne pentru fiecare articol din factură);
- 5 - **import date** (importat rezervare pe factura de ieșire);

Pentru fiecare caz în parte se înregistrează un SQL. E important să înțelegeți ce înseamnă:

- a. "SQL pe baza de date": acesta folosește sintaxa **Oracle/PosgreSQL** pentru a aduce date tabelele originale (exclus tabelele temporare) din Ora sau PG.
- b. "SQLite pe memorie": acesta folosește sintaxa **SQLite** aplicată pe tabelele de pe memorie, cele ale căror nume și structură le vedeți în opțiunea "Tabele WME" pe document.

Mecanismul de lucru cu tabele în WMEnterprise, pentru a înțelege diferența între cele două tipuri de SQL, este următoarea pentru fiecare document:

- **Oracle/PosgreSQL** se execută fraze SQL care mută date din tabelele originale (ex: IESIRI) în tabelele temporare (ex: TMP_IESIRI);
- **WME** aduce înregistrările din tabelele temporare în niște tabele pe memorie, cu numele de tabelă și de câmpuri așa cum le vedeți în "Tabele WME";
- tabelele pe memorie au structura identică cu tabelele temporare, dar au alte denumiri, și anume fără prefix TMP_;
- tabelele de pe memorie astfel încărcate, stau în spatele tuturor controalelor vizuale de pe machete și pot fi editate după ce apăsați "Aduag" sau "Modific";

- controalele prin care date sunt introduse la interfața memorează valorile în tabelele de pe memorie;
- la opțiunea "Salvez" se rulează algoritmi de atenționare / validare privitor la datele introduse;
- în cazul în care toate verificările de corectitudine a datelor au trecut cu succes, **WME** trimite informațiile din tabelele de pe memorie înapoi în tabelele temporare din **Oracle sau PostgreSQL**;
- după ce au ajuns în temporare (ex: TMP_IESIRI), acestea se codifică unic și ajung în cele din urmă să fie integrate în tabelele originale (ex: IESIRI).

Se înțelege că datele prezente pe memorie înainte de salvare (înainte de a ajunge în **Ora/PG**) nu au cum să fie interogate cu fraze SQL clasice care se aplică doar pentru tabele **Oracle sau PostgreSQL**.

Pentru datele pe memorie se aplică fraze SQL specifice, cu sintaxa de la un motor de baza de date special conceput pentru lucru pe memorie, denumit **SQLite**.

E de înțeles că în SQLite nu au ce cauta tabele din Ora sau PG, acestea nefiind încărcate pe memorie!

SQL pe baza de date folosește exclusiv tabele din baza de date Ora/PG, iar SQLite pe memorie folosește exclusiv tabelele din modulul de date (vizualizate prin "Tabele WME").

Cu alte cuvinte vorbim despre 2 dimensiuni paralele, care nu au nicio legătura una cu alta!

Acum, pe rand, voi explica fiecare rezolvare.

1. Validare la salvarea facturilor, astfel incat fiecare articol sa aiba clasa de articol asociata.

	Actiune	Tip SQL	ActionID
1	Import date	SQL pe baza de date	90
2	Validare	SQL pe baza de date	
3	Atentionare	SQLite pe memorie	
4	Interogare	SQL pe baza de date	51
5	Interogare	SQLite pe memorie	50

Mesaj validare:

<DenObiect> nu are clasa declarată!

<DenObiect> reprezintă câmpul pentru denumirea articolului care nu trece validarea, așa cum e el pe memorie în înregistrarea din tabela IESIRI1.

SQL pe baza de date

```
Select O.Denumire DenObiect
```

```
From NartObj o
```

```
inner join NArt a on a.codarticol = o.codarticol
```

```
Where Coalesce(a.Clasa, 0) = 0 and O.CodObiect = <lesiri1.CodObiect>
```

Acest SQL este lansat în Ora/PG, după ce <lesiri1.CodObiect> este înlocuit cu CodObiect din linia tabelii de pe memorie lesiri1.

Dacă în SQL pe baza de date este folosită tabela de detaliu principală de pe memorie, în cazul de față lesiri1, atunci acest SQL se va executa pentru fiecare linie de pe factură, deci pentru fiecare linie din lesiri1.

Similar dacă în SQL este folosită tabela de EXTENSIE la detaliul principal de pe memorie, în cazul de față lesiri1EXT.

2. Atenționare daca pe factura sunt linii cu Cantitate < 3

#	Actiune	Tip SQL	ActionID
1	Import date	SQL pe baza de date	90
2	Validare	SQL pe baza de date	
3	Atentionare	SQLite pe memorie	
4	Interogare	SQL pe baza de date	51
5	Interogare	SQLite pe memorie	50

Mesaj atentionare:

<DenObiect>: Cantitate facturata < 3 (<Cant>)

<DenObiect> reprezintă câmpul pentru denumirea articolului care nu trece validarea, așa cum e el pe memorie în inregistrarea din tabela IESIRI1.

<Cant> reprezintă câmpul pentru cantitate la articolul care nu trece validarea, așa cum e el pe memorie în inregistrarea din tabela IESIRI1.

SQLite pe memorie

```
Select I1.DenObiect, I1.Cant
From Iesiri1 I1
Where I1.Cant < 3
```

Acest SQL e lansat in SQLite pe tabelele pe memorie care nu sunt încă în baza de date.

3. Afișare listă cu valorile facturate pe gestiuni

#	Actiune	Tip SQL	ActionID
1	Import date	SQL pe baza de date	90
2	Validare	SQL pe baza de date	
3	Atentionare	SQLite pe memorie	
4	Interogare	SQL pe baza de date	51
5	Interogare	SQLite pe memorie	50

Titlu macheta de afișare:

Total valori pe gestiuni

Textul e afișat ca titlu pe macheta care afișează grila cu valorile

SQLite pe memorie

```

Select Rez.Gestiune, Sum(Rez.V) Valoare
from
(
Select Coalesce(Ix1.DenGest, Lt.Gestiune) Gestiune, I1.Cant*I1.PU V
from Iesiri1 I1
left join Iesiri1Ext Ix1 on Ix1.Codles1 = I1.Codles1
left join
( select
  Coalesce(x.CodlesInt1, 0) Codles1,
  Case when x.Gest = 0
    then Null
    else Substr(x.DenGestloArt, 1, InStr(x.DenGestloArt, Chr(13))-1)
  end Gestiune
  from LivrTot x
) Lt on Lt.Codles1 = I1.Codles1
) Rez
group by Rez.Gestiune
    
```

Acest SQL e lansat în SQLite pe tabelele pe memorie care nu sunt încă în baza de date. De observat cum am extras gestiunea, deoarece aveam următoarele probleme.

Pe Iesiri1 nu exista câmp cu denumirea de gestiune asociată serviciilor!

Mi-am "procurat" singur acest câmp pe extensia la Iesiri1!

Cu alte cuvinte, am adăugat un câmp temporar pe extensia Iesiri1EXT : DenGest.

Iesiri1EXT.DenGest se actualizează la schimbarea valorii Iesiri1.GestAct (vezi structura Iesiri1EXT!).

A doua problemă este faptul ca pe LivrTot (pagina de livrare din factura) există un câmp ce conține Denumirea gestiunii + Denumirea tipului contabil.

Am observat că cele 2 denumiri sunt separate prin caracterul CR, adica #13.

Așadar, pentru liniile de stoc, care au gestiunea pe câmpul LivrTot.DenGestIOArt, am copiat textul pana la #13 (adica Chr(13) in SQL).



4. Afișare lista cu codurile externe și cele interne pentru fiecare articol din factura.

#	Actiune	Tip SQL	ActionID
1	Import date	SQL pe baza de date	90
2	Validare	SQL pe baza de date	
3	Atentionare	SQLite pe memorie	
4	Interogare	SQL pe baza de date	51
5	Interogare	SQLite pe memorie	50

Titlu machetă de afișare:

Coduri articole

Textul e afișat ca titlu pe macheta care afișează grila cu valorile

SQL pe baza de date

Select

distinct O.Denumire, O.CodExtern, O.CodIntern

From NartObj o

Where O.CodObiect = <lesiri1.CodObiect>

Atenție! SQL pe baza de date are întotdeauna o legătură cu un câmp de pe memorie, acesta făcând legătura cu datele introduse pe machetă care încă nu se găsesc în Ora/PG.

Acest SQL este lansat în Ora/PG, după ce <lesiri1.CodObiect> este înlocuit cu CodObiect din linia tabeli de pe memorie lesiri1.

Dacă în SQL pe baza de date este folosită tabela de detaliu principală de pe memorie, în cazul de față lesiri1, atunci acest SQL se va executa pentru fiecare linie de pe factură, deci pentru fiecare linie din lesiri1.

Similar, dacă în SQL este folosită tabela de EXTENSIE la detaliul principal de pe memorie, în cazul de fata lesiri1EXT.

5. Import rezervare pe factura de ieșire.

#	Actiune	Tip SQL	ActionID
1	Import date	SQL pe baza de date	90
2	Validare	SQL pe baza de date	
3	Atentionare	SQLite pe memorie	
4	Interogare	SQL pe baza de date	51
5	Interogare	SQLite pe memorie	50

Titlu macheta de import:

neutilizat

SQL pe baza de date

```
Select R1.CODOBIECT, R1.CANTITATE as Cant, R1.PRET as PU,
      R1.CodRezervare1, R1.Observatii as ObsRezervare,
      R.DataDoc as DataRezervare, R.CodPart as Partener
from rezervare R
inner join Rezervare1 R1 on R1.CodRezervare = R.CodRezervare
where R.CodRezervare = <lesiriExt.CodRezervare>
```

Atentie! SQL pe baza de date are întotdeauna o legătura cu un câmp de pe memorie, acesta făcând legătura cu datele introduse pe macheta care încă nu se găsesc în Ora/PG.

Acest SQL este lansat în Ora/PG, după ce <lesiriExt.CodRezervare> este înlocuit cu CodRezervare din linia tabelii extensie la lesiri de pe memorie, lesiriEXT. Desigur, pentru a avea acest câmp în lesiriEXT, l-am adăugat în structura extensiei la lesiri.

El poate fi câmp original (se va și salva pe tabela din Ora/PG lesiriEXT!), sau doar temporar, caz în care este memorat doar pe TMP_lesiriEXT (caz în care nu se salvează pentru această înregistrare în lesiriEXT);

Acest câmp va fi completat folosind un control de alegere, denumit buton Lookup, așa cum o să vedeți mai jos, iar imediat după alegerea rezervării se va executa SQL de import, acesta aducând liniile rezervării pe lesiri1.

Observație:

Cursorul obținut la execuția SQL are următoarele câmpuri: **CODOBIECT, CANT, PU, CodRezervare1, ObsRezervare, DataRezervare, Partener.**

Algoritmul e suficient de sofisticat și folosește câmpurile astfel:

***CODOBIECT, CANT, PU** le găsește în structura IESIRI1, drept care adaugă o linie nouă în lesiri1 (implicit și în lesiri1EXT care e sincron cu lesiri1);

***CodRezervare1, ObsRezervare** le găsește pe lesiri1EXT (extensia detaliului principal), drept care le pune acolo;

CodRezervare1 e camp original, se va salva în Ora/PG pentru a avea evidența liniilor importate;

ObsRezervare e câmp temporar, se incarca doar la interfață pentru a da informații legate de linia importată din rezervări;

**DataRezervare îl găsește în extensia IesiriEXT, fiind un câmp temporar adăugat de mine;*

Se încarcă doar la interfața pentru a da informații legate de rezervare;

**Algoritmul identifică un buton Lookup pe macheta de ieșiri care e legat la câmpul Partener (butonul "Client");*

Partener e câmpul standard în Ieșiri si odată identificat, algoritmul îl propune pentru alegere pe view mic de parteneri, acționând automat butonul "Client".

De remarcat în grilă coloana ActionID introdusă la import (91) și la interogări (50, 51). ActionID se vor lega de butoanele care vor executa importul sau interogarea.

#	Actiune	Tip SQL	ActionID
1	Import date	SQL pe baza de date	90
2	Validare	SQL pe baza de date	
3	Atentionare	SQLite pe memorie	
4	Interogare	SQL pe baza de date	51
5	Interogare	SQLite pe memorie	50

Titlu macheta de afisare
Total valori pe gestiuni

SQLite pe memorie > Test SQL

De asemenea, de remarcat pe pagina de acțiuni asociate un butonul "Test SQL", care desigur va permite să testati fraza SQL pe care ați produs-o!

Acțiunea de import necesită alegerea în prealabil a unui cod, în cazul nostru al unui cod de rezervare.

Fiind vorba de o alegere, vom folosi pentru aceasta controlul potrivit, și anume un *LookupButton*.

Object Inspector

Single select Multi select

NewObjects

Panel Label Edit Memo Lookup Button Radio Check Comb Grid Tab

Controale **Actiuni asociate**

Control	Parent	Visible	Custom
WME_Label	xPObs	Yes	Yes
WME_NButt	Panel4	Yes	Yes
WMS	Panel8	Yes	No
xB11	Panel4	Yes	Yes
xBAgentSS	xPAgSS	Yes	Yes
xBImportLinii11	xPColor	Yes	Yes
xDataRezerv	xPColor	Yes	Yes
xLDataRezerv	xPColor	Yes	Yes
xPAgSS	PAnaliza	Yes	Yes
xPColor	PTop	Yes	Yes
xPObs	PSuplimEF	Yes	Yes

ClassName: TWinMDBLookup

Name	Value
Name	xBImportLinii11
Caption	Import REZERVARI
Left	1
Top	1
Width	152
Height	45
Visible	<input checked="" type="checkbox"/> Yes
Align	Top
DataSource	DSExtMaster
DataField	CODREZERVARE
LookupSource	DcCust_1012
LookupField	CODREZERVARE
LookupDisplay	SERIADOC;NRDOC;DATADOC
ActionID	90

Observați proprietatea ActionID = 90.

Aceasta proprietate va declanșa SQL-ul atasat liniei de import cu ActionID 90 și va aduce datele pe factură.

Butonul e legat la Datasource = DSExtMaster (adica in final la lesiriEXT).

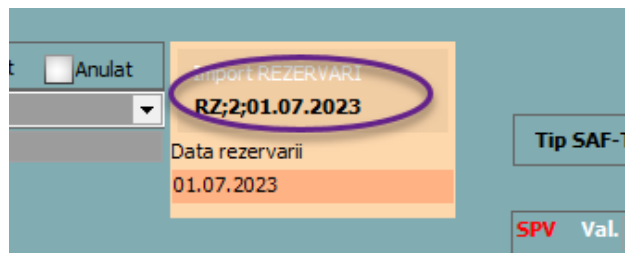
Codul ales se stochează în DataField = CodRezervare , adică lesiriEXT.CodRezervare.

Pentru a afișa pe buton informații legate de rezervare (altfel CodRezervare nu ar avea înțeles pentru operator), se folosește LookupSource = DcCust_1012.

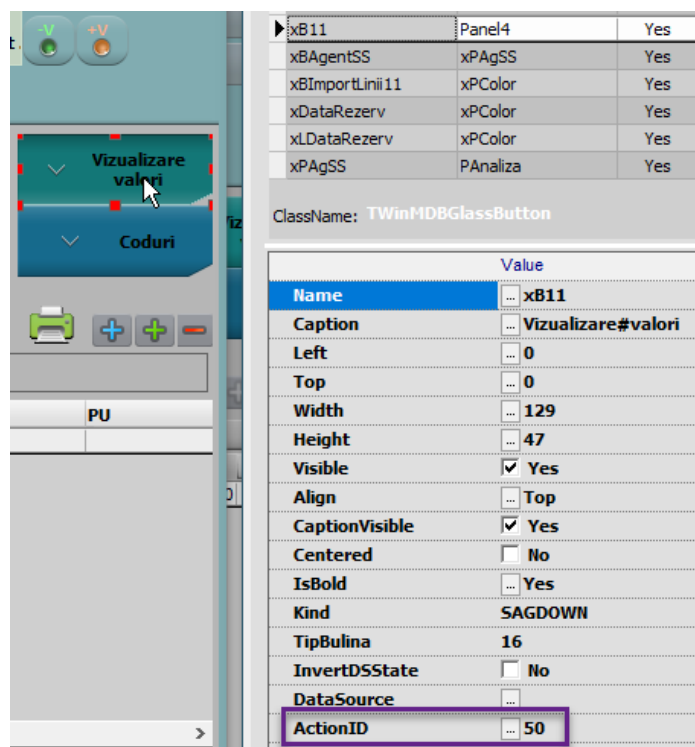
Acest mod de codificare a unui view mic asociat, DcCust_1012, identifică prin 1012 cod de identificare a machetei proprii "rezervări" , care are ID egal cu 1012.

Pentru a aduce informațiile cerute în LookupDisplay, și anume SERIADOC;NrDOC;DataDoc (care sunt din tabela Rezervari!), butonul leagă câmpul lesiriEXT.CodRezervare cu înregistrarea din DcCust_1012 (care e tabela Rezervari) prin LookupField = CodRezervare (adica Rezervari.CodRezervare).
Rezervari.CodRezervare = lesiriEXT.CodRezervare, unde Rezervari.CodRezervare este **LookupField**, iar lesiriEXT.CodRezervare este **DataField**.

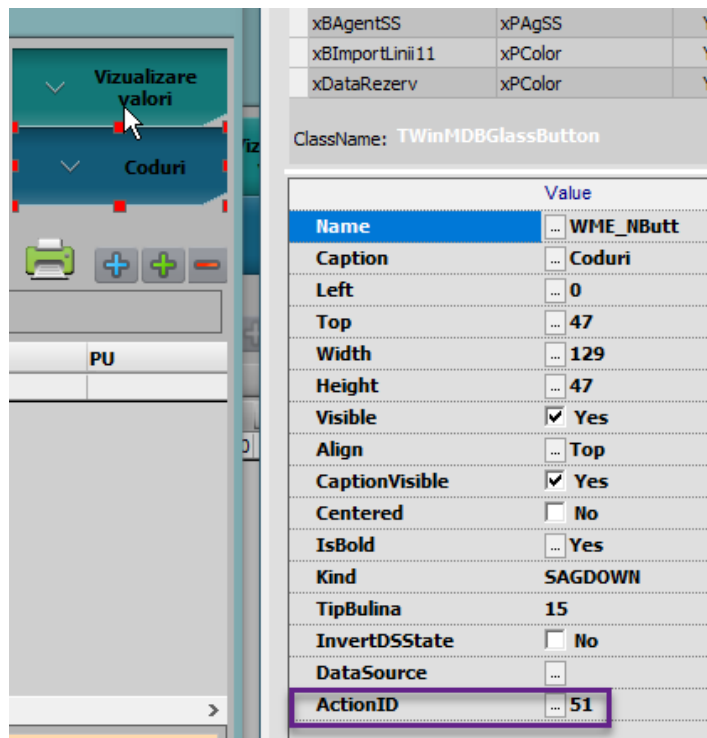
De pe această linie se afișează pe butonul lookup informațiile următoare:



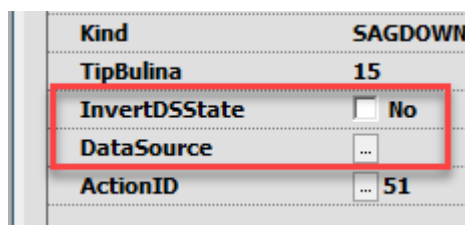
ActionID 50 din grila de "Acțiuni asociate" e legat de un buton simplu, care are și el proprietate ActionID potrivită.



Similar, ActionID 51 din grila de "Acțiuni asociate" e legat de un buton simplu, care are și el proprietate ActionID potrivită.



O ultimă observație legată de butoanele simple.



Există la butoanele simple o proprietate denumită **DataSource**.

*Dacă aici NU e specificată o valoare, atunci butonul e activ indiferent dacă macheta e în editare sau înregistrarea a fost salvată.

*Dacă aici e specificată o valoare, atunci butonul e activ astfel:

- când InvertDSSState = No, atunci butonul e activ doar când ești în editate pe machetă;
- când InvertDSSState = Yes, atunci butonul e activ doar după salvarea înregistrării;

Review-uri document

Rev. 1.0 07.01.2024 Documentație inițială

