

# GUIDA TECNICA C#

## ERGMEEN 3

### Sommario

Namespace Applicazioni .....	3
Namespace e DLL .....	3
Operazioni Preliminari Nuovi Programmi Ergmen3 + EF .....	4
Creazione Modello Entity Framework .....	5
Utile C# ed equivalente VB.NET .....	9
Lanciare un altro programma .....	9
Convenzioni Sintassi .....	10
Nomi Grafica .....	10
Nuova Grafica e DevExpress 19.2.6 .....	11
Layout Responsive e DataBinding .....	11
BarManager .....	11
Impostazioni Generali .....	12
Varie DX e Progettazione Form .....	13
Aggiunta check edit a un griglia con i flag S / N .....	13
Bottone a due stati (Esempio <b>Grassetto Corsivo Sottolineato</b> ) .....	13
Griglia in Sola Lettura EntityIstantFeedbackSource .....	13
Aggiunta filtro da codice .....	13
Gestione QuickSearchInfos Filtri .....	14
Istruzioni per la progettazione delle form di ricerca .....	15
Istruzioni da seguire nella progettazione (Da rivedere con tablepanel devexpress) .....	15
Gestione dei filtri .....	15
Entity Framework e Informix .....	16
Creazione Chiavi Primarie e Chiavi Esterne .....	16
Tabelle senza Indice Univoco oppure su elementi che potrebbero essere null (esempio CLSART) .....	16
Casi Particolari Gestione Chiavi Esterne (NON AGGIUNGERE FK) .....	16
Gestione Sinonimi .....	16
Ottimizzazione Velocità di Avvio Programmi .....	18
DLL in GAC e NativelImages (immagini compilate per la propria specifica architettura) .....	18
ShowMode della Form a caricamento finito, per eliminare flash grafico .....	18
Problemi e Workaround .....	19

DevExpress .....	19
Problemi CheckEdit in Griglia (servono due click per cambiare lo stato) .....	19
Driver IBM e Database .....	21
Problema Login Programmi con Entity Framework Driver IBM 11.5 + Windows 7 .....	21
Problema .NET (Anche ergmen2) i programmi devono essere lanciati come amministratore e 'LoaderLock' .....	21
Problema ergmen2 e (forse ergmen3) errore nel lanciare alcuni prog. VB6 se non lanciato come amministratore	22
Problema LOGIN e VERSIONE dentro file .edmx.....	22
Problema tabelle create da utente diverso da erg .....	22
Visual Studio.....	23
I componenti grafici ereditati risultano bloccati nella finestra di progettazione .....	23
Impossibile installare pacchetti NuGet .....	23
<i>Library not registered. (Exception from HRESULT: 0x8002801D (TYPE_E_LIBNOTREGISTERED))</i> .....	23
Errore Visual Studio ContextSwitchDeadlock .....	24

## Namespace Applicazioni

- Erg -> Namespace di base per programmi e librerie
- Db -> Namespace del modello e delle classi inerenti a database
- Forms -> Namespace delle forms delle applicazioni

## Namespace e DLL

- Erg.Base -> tutte le funzionalità che non necessitano di una connessione a DB (es. ex lib\_win, classi e funzione per la gestione dei tipi Getv() / Setv() e utility per le date ...).
- Erg.ErgApplication -> funzionalità comuni a tutte le applicazioni che necessitano di una connessione a DB e le funzioni di connessione e gestione delle transazioni.
- Erg.Report -> Classi di creazione/modifica/stampa report (ex lib\_xtrareport, r\_repcli, ...)
- Erg.Tools -> Classi di funzioni aggiuntive per le applicazioni come invio mail, cattura schermo, log, ...
- Erg.UserControls -> Controlli personalizzati (Griglie / Filtri / ...)
- Erg.CSharp -> Funzioni VB.NET che necessitano di un'implementazione differente in C#

### DLL Deprecate

Le DLL **Erg.WaitForm**, **Erg.Utils**, **Erg.EFModel** sono state **DEPRECATE** ed inglobate nella **Erg.ErgApplication** per ottimizzare il caricamento degli Assembly in fase di avvio.

Nei programmi che ancora le utilizzano sono state copiate localmente.

\*NOTE GENERALI PROGRAMMI

---

### **Suddivisione Cartelle**

i file di estensione delle classi EF si trovano nella cartella `ergdis\ergdis\vbnet\ef_est`

le icone dei nuovi programmi sono nella cartella `ergdis\ergdis\vbnet\img\icon`

le nuove ricerche sono nella cartella `ergdis\ergdis\vbnet\ef_ric`

**Icona Applicazione** .... /img/icon/ergmen3-color.ico

(non è essenziale ma bello quando l'utente si mette il collegamento sul desktop ad un programma)

## Operazioni Preliminari Nuovi Programmi Ergmen3 + EF

L'aggiunta dei pacchetti NuGet e la creazione del modello sono necessari per qualsiasi programma effettui la connessione al database (anche solo per fare il login ad Informix)

Il modello in fase di inizializzazione verifica che la propria struttura coincida con quella del database, in particolare che al DB corrente non manchi nulla rispetto alla struttura del modello.

---

1. Modifica Namespace progetto in "Erg"
  - DoubleClick su Properties (oppure RightClick sul progetto > Proprietà)
  - Scheda Applicazione
  - Spazio dei Nomi Predefinito
2. Aggiunta NuGet EntityFramework.IBM.DB2 e dipendenze
  - RightClick sul progetto > Gestisci Pacchetti NuGet

(L'installazione del pacchetto EntityFramework.IBM.DB2 provvederà ad aggiungere al progetto anche le dipendenze necessarie)

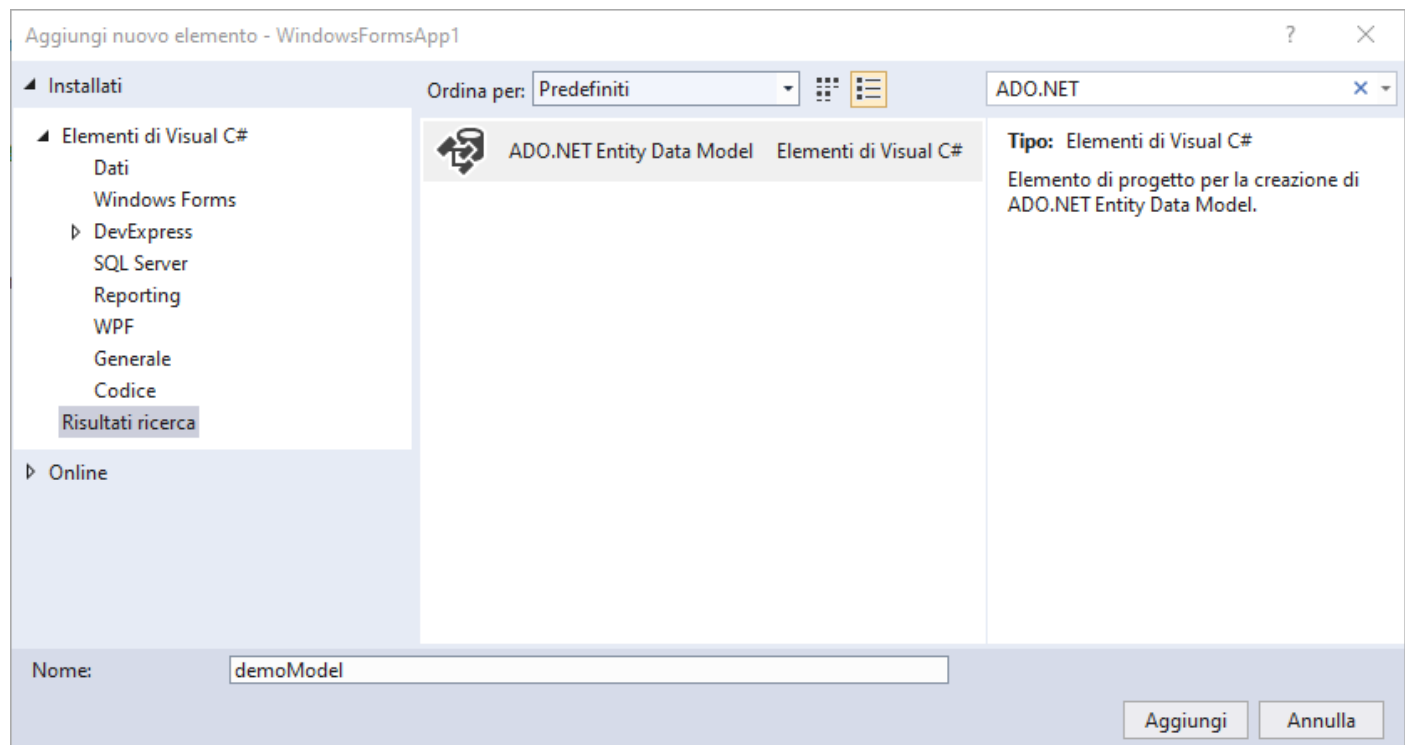
3. Creazione Modello EF (vedi sotto)
4. Modifica Namespace modello EF in "Db"
  - nelle proprietà dei 2 file .tt in VS2019
  - Spazio dei nomi dello strumento personalizzato
5. Aggiunta riferimenti alle librerie Erg.\* dalla cartella vbnet\_dll
6. Aggiunta libreria per il reperimento delle dll dalla cartella di rete (ergdis\ergdis\vbnet\ef\_lib.vb\lib\_dll.cs)
7. Aggiunta come riferimento ergdisEntities.custom.cs (ergdis\ergdis\vbnet\ef\_est)
8. Copia Classe Statica Program da un altro programma
9. Modifica Versione su file .edmx (Notepad) verificare che sia **"12.10.3000, 0, 0"**. Questo per non avere problemi futuri di versione.  
**SI** <Schema ... ProviderManifestToken="IDS/UNIX64, 12.10.3000, 0, 0" ... >  
**NO** <Schema ... ProviderManifestToken="IDS/UNIX64, 12.10.5000, 311684, 0" ... >

10. Aggiunta modelli utilizzati dal login nella stringa di connessione App.config

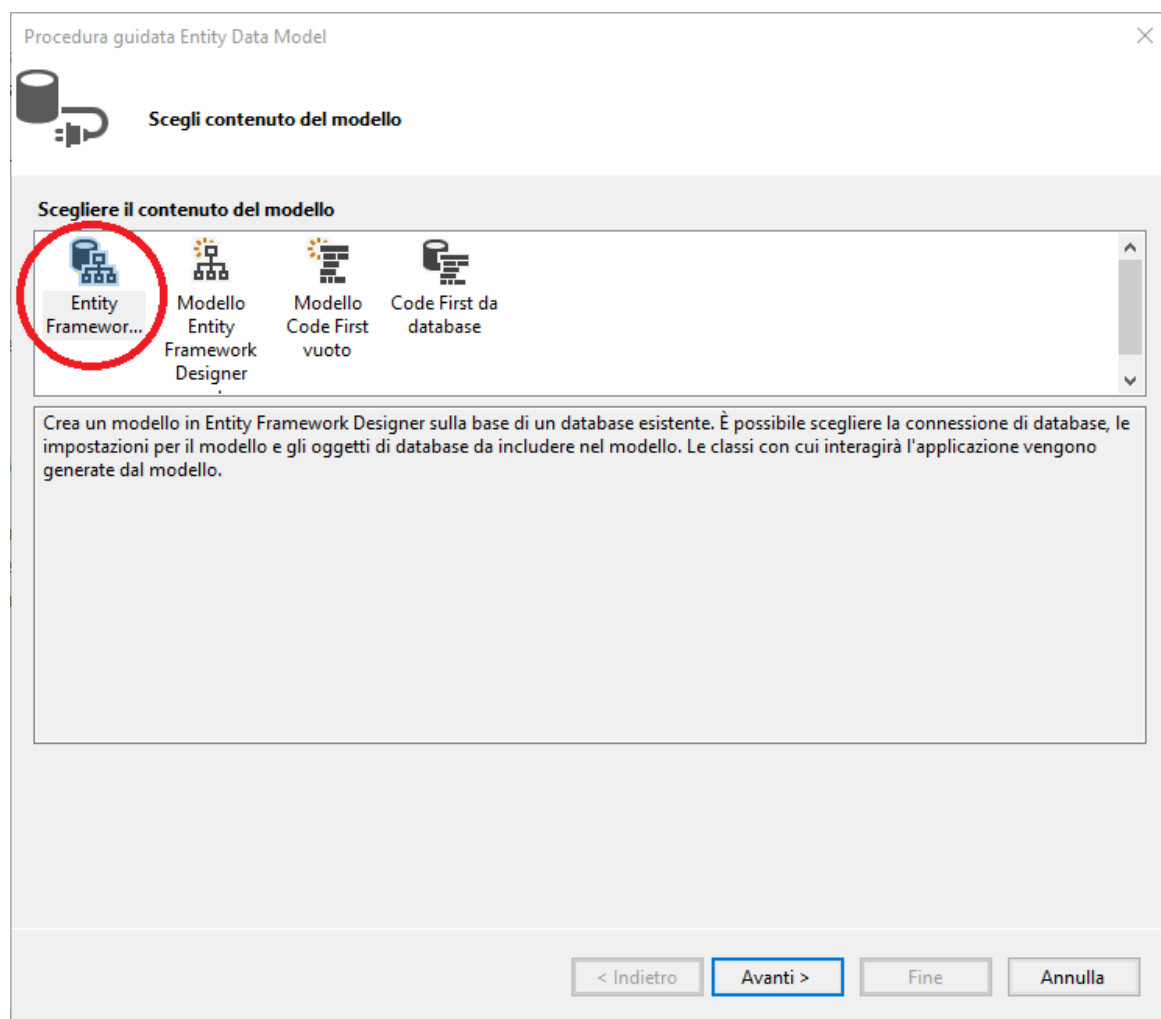
```
<connectionStrings>
  <add name="ergdisEntities"
    connectionString="metadata=res://*/innolvaModel.csdl|res://*/mioProgettoModel.ssdl|res://*/mioProgettoModel.msl|res://*/dbModel.csdl|res://*/dbModel.ssdl|res://*/dbModel.msl;provider=IBM.Data.DB2;provider connection string=&quot;Database=test;User ID=erg;Password=erg;Server=lxsviluppo:9089;Persist Security Info=True&quot;;providerName="System.Data.EntityClient" />
</connectionStrings>
```

## Creazione Modello Entity Framework

- Aggiungi > Nuovo Elemento > ADO.NET Entity Data Model (nomeProgrammaModel)



- EntityFrameworkDesigner da Database -> tutte le spunte



➤ Nuova Connessione

Proprietà connessione

Immettere le informazioni per la connessione all'origine dati selezionata o scegliere "Modifica" per selezionare un'origine dati e/o un provider differente.

Origine dati:

IBM DB2 and IDS Servers (IBM DB2 and IDS Data Provider for .NET Framework) Modifica...

Specificare quanto riportato di seguito per collegarsi ai dati IBM:

1. Selezionare o immettere il nome server:

lxsviluppo:9089 Aggiorna

2. Immettere le informazioni per collegarsi al server:

ID utente: erg

Password: \*\*\*

☐ Salva password

3. Selezionare o immettere un nome di database


test

+ Specificare le opzioni di connessione:

+ Specificare le opzioni di filtro:

- Nome Oggetto Modello ergdisEntities (così da poter sfruttare i file di estensione)

Procedura guidata Entity Data Model

 Scegliere la connessione dati

**Scegliere la connessione dati da utilizzare per connettere l'applicazione al database.**

erg@TEST [IDS/UNIX64 12.10.0000]10 Nuova connessione...

La stringa di connessione sembra contenere dati sensibili, ad esempio una password, necessari per connettersi al database. L'archiviazione dei dati sensibili nella stringa di connessione rappresenta un rischio per la sicurezza. Includere dati sensibili nella stringa di connessione?

☐ No, escludi dati sensibili dalla stringa di connessione. Queste informazioni verranno impostate nel codice dell'applicazione.

☒ Sì, includi i dati sensibili nella stringa di connessione.

Stringa di connessione:

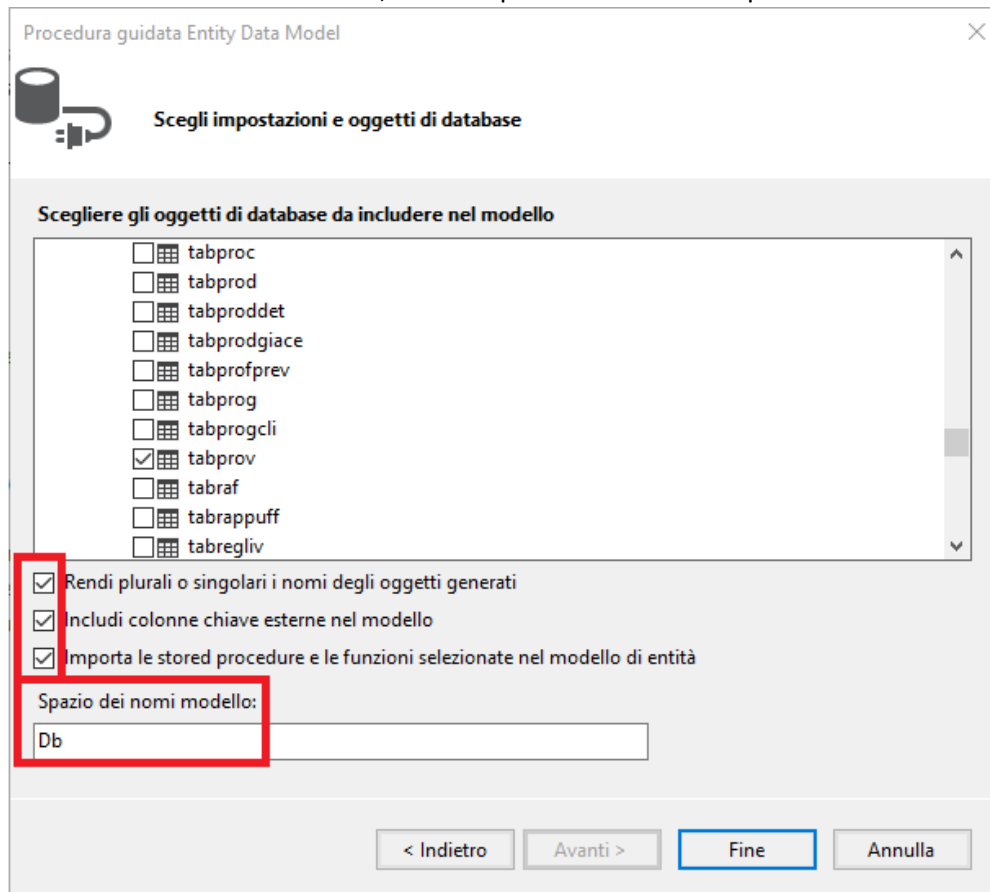
```
metadata=res://*/demoModel.csdl|res://*/demoModel.ssdl|res://*/demoModel.msl;provider=IBM.Data.DB2;provider
connection string="Database=test;User ID=erg;Password=erg;Server=lxsviluppo:9089"
```

☒ Salva impostazioni di connessione in App.Config come:

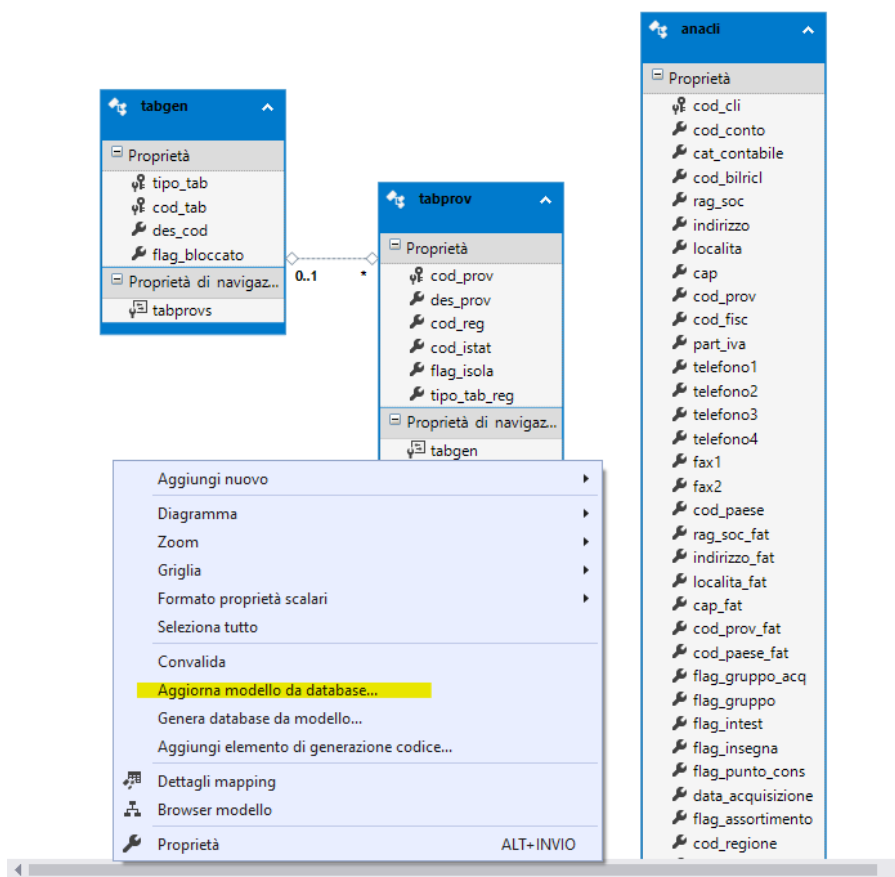
ergdisEntities

< Indietro Avanti > Fine Annulla

- Selezionare le tabelle da utilizzare, tutte le spunte e “Db” come spazio dei nomi del modello



- Per aggiornare il modello o aggiungere tabella (Aggiorna Modello da database)





## Utile C# ed equivalente VB.NET

Lanciare un altro programma

(Nota: consideriamo un qualsiasi programma interno al gestionale come la relativa voce in ergmen)

C#	VB.NET
<pre>var programma = new Erg.ErgApplication.db.ergman() programma.app_cli = "contgestcassa" programma.parameters = ..... programma.ExecuteAndWait(myApp.AppInfo) //programma.Execute(myApp.AppInfo)</pre>	<pre>Dim programma = New Erg.ErgApplication.db.ergman() programma.app_cli = "contgestcassa" programma.parameters = ..... programma.ExecuteAndWait(myApp.AppInfo) `programma.Execute(myApp.AppInfo)</pre>

Se invece il programma è un eseguibile esterno ad ERGDIS **Erg.Base.FileMng.Execute(.....)**

### FILE COME "libreria" (Module in VB.NET)

C#      Classe Static (public static class NomeClasse)

Per accedere dal programma chiamante.

1. [CONSIGLIATO] NomeClasse.NomeFunzione();
2. Aggiungere rif. statico *using static Namespace.NomeClasseStatica* poi nel codice posso invocare le funzioni senza dover utilizzare il nome della classe statica.

---

### NOME ASSEMBLY PROGRAMMA IN ESECUZIONE

VB.NET Application.Info.AssemblyName

C#      System.Reflection.Assembly.GetExecutingAssembly().GetName().Name

---

## Convenzioni Sintassi

<b>ELEMENTI GRAFICI</b>	snake_case con il tipo e nome a DB [chk = CheckEdit, btn = SimpleButton, ...] (chk_flag_cessato, btn_salva_archivia, ...) vedi Nomi Grafica
<b>VARIABILI LOCALI</b>	camelCase (nomeVariabileLocale)
<b>ARGOMENTI</b>	camelCase (nomeArgomento) _camelCase (se stesso nome di una variabile _codFisc e codFisc)
<b>COSTANTI</b>	MAIUSCOLO (NOME_COSTANTE)
<b>METODI</b>	PascalCase (NomeMetodo)
<b>CLASSI</b>	PascalCase (NomeClasse)

Classi e relativi Campi Dati autogenerati da EF e/o relativi a elementi a DB mantengono il nomi utilizzati convenzionalmente in ERGIS (anacli, tipi\_doc\_fisc, flag\_bloccato, ....).

## Nomi Grafica

btn --> SimpleButton, BarButtonItem

bte --> ButtonEdit

lbl --> LabelControl

list --> ListBoxControl

pic --> PictureEdit

calc --> CalcEdit

chk --> CheckEdit

chkc --> CheckedComboBoxEdit

lkp --> LookUpEdit

cmb --> ComboBoxItem

dte --> DateEdit

tms --> TimeSpanEdit

color --> ColorEdit

font --> FontEdit

img --> ImageEdit

imgc --> ImageComboBoxEdit

spin --> SpinEdit

txt --> TextEdit

time --> TimeEdit

cal --> CalendarControl

tab --> TabPage

tabc --> XtraTabControl

rich --> RichEdit

table --> TablePanel

grp --> GroupControl

range --> RangeControl

rad --> RadioGroup

col --> GridColumn

gridc --> CustomGridControl, GridControl

gridv --> CustomGridView, GridView

### NOTE

Nell'ottica di possibili sviluppi futuri è bene nominare i componenti come le griglie con un nome univoco

(gridv\_cli non grid\_view)... fatta eccezione per componenti tipo BarManager, HelpProvider, ToolTipProvider, BarAndDockingController che possono essere lasciati con il nome con cui vengono creati dal Visual Studio

## Nuova Grafica e DevExpress 19.2.6

### Layout Responsive e DataBinding

Per consentire il ridimensionamento delle form e necessario utilizzare dei contenitori per i controlli.

**TablePanel:** solo se ho una griglia con filtri o devo avere delle percentuali

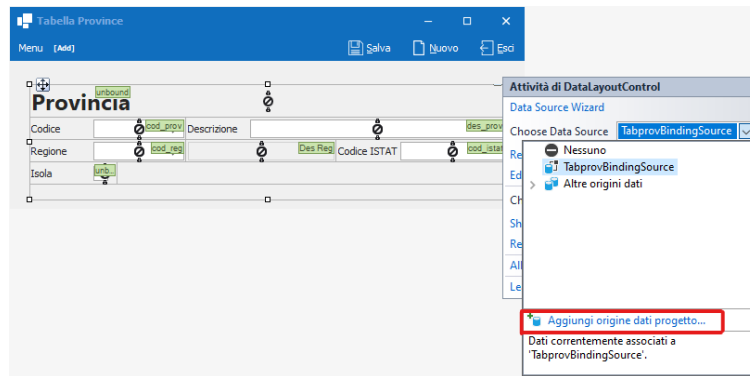
**LayoutControl:** Per tutti i programmi non associati direttamente ad una sola tabella.

**DataLayoutControl:** Per tutti i manutenzione tabelle (tabprov, anaart, ....), questo permette di associare una specifica classe al layout, quindi assegnando un oggetto alla form è possibile popolare/modificare in automatico i valori dell'oggetto per poi salvarli a DB.

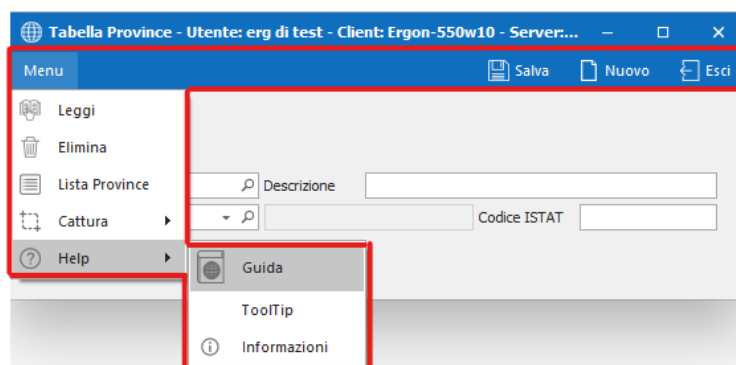
**\* le form che utilizzano il DataLayoutControl dovranno implementare l'interfaccia IForm (Erg.Base)**

Assegnazione Classe e Proprietà

- Choose Data Source
- Aggiungi origine dati progetto
- Oggetto
- Selezione della classe corrispondente
- Assegnazione dei campi a video alle proprietà dell'oggetto (cliccando sui rettangolini verdi)



### BarManager



Per ottenere la barra trascinare sulla form il componente BarManager; da qui è possibile aggiungere o dei "Button" o dei "Menu". Come standard è stato deciso che c'è un menu "Menu" in alto a sinistra e dei button con le azioni principali in alto a destra (solitamente "Nuovo", "Salva", "Esci"). Nel menu invece di solito è presente "Leggi", "Elimina", "Help" che contiene a sua volta "Guida" (apre la guida html), "ToolTip" (abilita/disabilita i tooltip) e "Info" per le informazioni sulla versione del programma.

**\* è possibile copiare la struttura dei componenti del bar manager da un programma esistente**

- RunDesigner (dal BarManager)
- Layouts (esportarlo su file dal programma esistente e reimportarlo dal programma nuovo)

## Impostazioni Generali

### **Disabilitare funzionalità di Quick Customization**

LayoutControl / DataLayoutControl

- AllowCustomization

BarManager

- AllowCustomization
- AllowQuickCustomization
- AllowAddNewItem

Barra Menu

- AllowQuickCustomization
- DrawDragBorder

## Varie DX e Progettazione Form

Per un controllo che apre una ricerca usare o un ButtonEdit o un LookUpEdit: il primo se si vuole mettere il solo bottone della lente d'ingrandimento per aprire la ricerca, il secondo se si vuole aggiungerne un altro che apre un menu a tendina nel quale vi è una griglia con gli stessi valori della ricerca (comodo per ricerche veloci).

Cercare di raggruppare i vari "blocchi" di controlli inerenti allo stesso ambito in un group: basta selezionare i vari LayoutControlItem di interesse, fare tasto destro e poi "Group".

LookUpEdit. Una lookUpEdit va così configurata:

DisplayMember e ValueMember = {nome proprietà relativa al controllo}

TextEditStyle = Standard

SearchMode = AutoComplete

[Andare in "All Properties" e aggiungere le colonne necessarie e per ognuna compilare il nome della proprietà collegata]

[Nel New della form associare all'evento {lookUpEdit}.Enter una Sub che gestisce l'assegnazione del datasource alla lista di elementi che si vogliono visualizzare (vedi ad esempio tabage)]

CheckEdit. Per gestire le checkedit aggiungere ad ognuna il tag [DATA] = { proprietà da bindare }

Aggiunta check edit a un griglia con i flag S / N

1. RepositoryItemCheckEdit
2. NullStyle -> Unchecked o (checked a seconda di come è stata impostato il flag a db)
3. ValueChecked -> String -> "S"
4. ValueUnchecked -> String -> "N"

Bottone a due stati (Esempio **Grassetto Corsivo Sottolineato**)

- ButtonStyle => "Check" (tra le proprietà del bottone)
- Nome\_bottone.Down => la property per vederne lo stato true/false

Griglia in Sola Lettura EntityIstantFeedbackSource

Dalla griglia:

- DataSourceWizard
- EntityFramework (ergdisEntity)
- Asynchronous Server-Side Data Processing
- Selezione Tabella
- Modifica dell'evento GetQueryable:

```
Db.ergdisEntities DataContext = new Db.ergdisEntities(thisAppInfo.EFConnection);  
IQueryable<Db.tabprov> item = DataContext.tabgens.AsNoTracking();  
e.QueryableSource = item;
```

Aggiunta filtro da codice

```
UcFilter newUcFilter = UcFiltri1.AddNewUcFilter();  
FilterInfo newUcFilterInfo = new FilterInfo(newUcFilter.Id);  
newUcFilterInfo.Column = new Column(col_flag_intest);  
newUcFilterInfo.FilterOperator = FilterOperator.Association.Find(s => s.Description == "Valori");  
newUcFilterInfo.Values = new List<object>{new List<object> { "Si" }, null};  
newUcFilterInfo.Id = newUcFilter.Id;  
newUcFilter.SetUcFilterFromInfo(newUcFilterInfo);  
UcFiltri1.FiltersInfo.AddOrUpdateFilterInfo(newUcFilter);
```

## Gestione QuickSearchInfos Filtri

C#	VB.NET
<pre>UcFiltril.UcFiltersSettings.QuickSearchInfos = new List&lt;QuickSearchInfo&gt; {     new QuickSearchInfo()     {         Column = new Column(colcod_agente),         DbSetName = "tabages",         CodColumnName = "cod_age",         DesColumnName = "des_age",         ElementType = GetType(Db.tabage)     } };</pre>	<pre>UcFiltril.UcFiltersSettings.QuickSearchInfos = New List(Of QuickSearchInfo) From {     New QuickSearchInfo() With {         .Column = New Column(colcod_agente),         .DbSetName = "tabages",         .CodColumnName = "cod_age",         .DesColumnName = "des_age",         .ElementType = GetType(Db.tabage)     } }</pre>

## Istruzioni per la progettazione delle form di ricerca

Abbiamo creato una form di ricerca base con integrato tutta una serie di sub/function per la gestione degli eventi. Questo per evitare di avere nel codice di ogni form di ricerca le stesse sub/function con il medesimo codice. Quindi, nella fase di progettazione di una nuova schermata di ricerca, sarà sufficiente ereditare da tale form. La classe base si chiama **BaseRic** e si trova nella DLL Erg.ErgApplication.

Istruzioni da seguire nella progettazione (Da rivedere con tablepanel devexpress)

1. Creare una nuova XtraForm e chiamarla come si desidera

2. Entrare nel designer della form appena creata e sostituire "Inherits DevExpress.XtraEditors.XtraForm" con "Inherits Erg.ErgApplication.BaseRic" . Precedentemente è dunque necessario importare anche il riferimento a Erg.ErgApplication.dll. Fatto ciò nella schermata di progettazione si dovrebbe avere una form popolata con i controlli della form base

3. Aggiungere un TableLayoutPanel nel quale inserire il controllo dei filtri e la griglia. Questo però non è possibile farlo direttamente dalla schermata di progettazione ma bisogna seguire le seguenti:

i. Entrare nel designer e rimuovere le istruzioni:

```
Me.Controls.SetChildIndex(Me.UcFiltri1, 0)
Me.Controls.SetChildIndex(Me.mainGrid, 0)
```

ii. Sempre nel designer aggiungere in corrispondenza dei settaggi di TableLayoutPanel1:

```
Me.TableLayoutPanel1.Controls.Add(Me.UcFiltri1, 0, 0)
Me.TableLayoutPanel1.Controls.Add(Me.mainGrid, 0, 1)
```

Così facendo i controlli dei filtri e la griglia saranno dentro al TableLayoutPanel.

### Gestione dei filtri

Per quanto riguarda la gestione dei filtri si appunta che:

1. è possibile impostare di default il filtro sul flag\_bloccato = "N";
2. è possibile inoltre gestire il filtro su colonne che a database contengono codici. Per essere più chiari, in anafor vi è la colonna flag\_ita\_est che a db assume i soli valori I (italia), E (estero).

Nella proprietà UcFiltri.Info.Mapping possiamo andare a specificare un dizionario del tipo gridColumn => associazione in cui associazione è una lista di oggetti del tipo IBaseMultiFlag con proprietà Value, DbCodes, Description. In DbCodes è possibile andare a specificare più codici per gestire eventuali valori di default - ad esempio se "" in flag\_ita\_est è rimandato di default come "I" allora DbCodes = {"I", ""}; specificare sempre per primo il codice non default (nel nostro caso "I" e non ""). Per questa parte si rimanda alla spiegazione della gestione dei multiflag e alla visione di una ricerca che implementi il mapping (r\_anafor).

## Entity Framework e Informix

### Creazione Chiavi Primarie e Chiavi Esterne

#### Inserimento chiave primaria:

Creazione indice univoco sul/sui campo/i, in questo modo l'indice avrà il nome che usiamo di solito, altrimenti avrà un id numerico creato in automatico dal DB.

```
ALTER TABLE tabprov ADD CONSTRAINT PRIMARY KEY (cod_prov) CONSTRAINT pk_tabprov;
```

#### Inserimento chiave esterna:

Nel caso la chiave coinvolga la tabella tabgen bisogna creare nella tabella di partenza un campo con il tipo\_tab corrispondente, perché la chiave esterna deve comprendere tutta la chiave primaria della tabella associata.

```
ALTER TABLE tabprov ADD tipo_tab_reg CHAR(3) DEFAULT 'REG';
```

```
ALTER TABLE tabprov ADD CONSTRAINT (FOREIGN KEY (tipo_tab_reg, cod_reg) REFERENCES tabgen CONSTRAINT fk_tabprov_tabgen);
```

### Tabelle senza Indice Univoco oppure su elementi che potrebbero essere null (esempio CLSART)

Uno dei requisiti fondamentali della chiave primaria è che i campi di cui è composta devono essere tutti valorizzati. Nel caso quindi, di tabelle in cui non è possibile identificare un campo (o un insieme di campi) come chiave, la scelta migliore consiste nell'inserimento di un campo id serial da utilizzare come chiave della tabella. Nel caso specifico di CLSART, l'utilizzo / estrazione delle classificazioni rimane invariato.

**L'UTILIZZO DI UNA TABELLA SENZA UNA CHIAVE PRIMARIA È FORTEMENTE SCONSIGLIATO**, nonostante Visual Studio permetta di aggiungere al modello una tabella priva di chiave, questa viene comunque dedotta tramite dei meccanismi automatici, che poi comportano possibili **ERRORI** nell'estrazioni dei record difficilmente identificabili.

### Casi Particolari Gestione Chiavi Esterne (NON AGGIUNGERE FK)

Nel caso di tabelle con collegamenti multipli alla stessa tabella (es. ordfor, ordforndet, ordspe, ordspedet → anacli) le chiavi esterne NON vanno create ed il join va eseguito manualmente. Lo stesso vale per i riferimenti circolari nella stessa tabella.

### Gestione Sinonimi

- Creare il modello dal DB principale del cliente, ovvero quello senza le tabelle sinonimo, il fatto di lanciare poi il programma su un DB che contiene i sinonimi al posto delle tabelle fisiche non crea problemi.
- Ne caso la prima opzione non sia percorribile è possibile aggiungere un collegamento tra tabelle direttamente dal modello all'interno del progetto. L'opzione uno resta comunque sempre la migliore

**NOTA: usando le proprietà di navigazione, le chiavi esterne servono per far sapere al modello EF e al driver che costruisce la query quali tabelle mettere in join con la principale.**





## Ottimizzazione Velocità di Avvio Programmi

- Compilazione Programmi con Configurazione 32bit
- Abilitare utilizzo DirectX per elaborazioni grafiche DevExpress [DevExpress Project Settings > Use DirectX]
- Multicore JIT (Caricamento riferimenti sfruttando il multicore)

Nel File Program.vb / Program.cs - > My.MyApplication.Enable\_MultiCore\_JIT();

### DLL in GAC e NativeImages (immagini compilate per la propria specifica architettura)

1. GAC (con Developer Command Prompt oppure con gacutil.exe)

```
gacutil -If nome.dll
```

2. Native Images NGEN (cmd come amministratore)

```
cd %WINDIR%\Microsoft.NET\Framework\v4.0.30319\  
ngen install <Assembly path>
```

### ShowMode della Form a caricamento finito, per eliminare flash grafico

NON è propriamente un'ottimizzazione ma su programmi con parecchi controlli evita che gli elementi compaiano uno per volta. ATTENZIONE: La visualizzazione della form viene rimandata di una frazione di secondo.

#### VB

```
Protected Overrides ReadOnly Property ShowMode As FormShowMode  
    Get  
        Return FormShowMode.AfterInitialization  
    End Get  
End Property
```

#### C#

```
protected override FormShowMode ShowMode  
{  
    get  
    {  
        return FormShowMode.AfterInitialization;  
    }  
}
```

## Problemi e Workaround

### DevExpress

Problemi CheckEdit in Griglia (servono due click per cambiare lo stato)

11. Se il selection mode è sulla singola cella basta impostare:

OptionsBehavior.EditorShowMode = MouseDown sulla GridView

12. Altrimenti se il selection mode è sulla riga bisogna gestire l'evento MouseDown della GridView

----- VB -----

```
Private Sub GridView_MouseDown(ByVal sender As Object, ByVal e As MouseEventArgs)
```

```
    Dim view = TryCast(sender, GridView)
```

```
    Dim hitInfo As GridHitInfo = view.CalcHitInfo(e.Location)
```

```
    If Not hitInfo.InRowCell OrElse Not (TypeOf hitInfo.Column.RealColumnEdit Is RepositoryItemCheckEdit) Then
```

```
        Return
```

```
    End If
```

```
    view.FocusedColumn = hitInfo.Column
```

```
    view.FocusedRowHandle = hitInfo.RowHandle
```

```
    view.ShowEditor()
```

```
    Dim edit As CheckEdit = TryCast(view.ActiveEditor, CheckEdit)
```

```
    If edit Is Nothing Then
```

```
        Return
```

```
    End If
```

```
    edit.Toggle()
```

```
    view.CloseEditor() ' call this method if you want to keep the view scrollable using the mouse wheel
```

```
    DXMouseEventArgs.GetMouseArgs(e).Handled = True
```

```
End Sub
```

----- C# -----

```
void GridView_MouseDown(object sender, MouseEventArgs e) {
```

```
    var view = sender as GridView;
```

```
    GridHitInfo hitInfo = view.CalcHitInfo(e.Location);
```

```

if (!hitInfo.InRowCell || !(hitInfo.Column.RealColumnEdit is RepositoryItemCheckEdit))
    return;

view.FocusedColumn = hitInfo.Column;
view.FocusedRowHandle = hitInfo.RowHandle;
view.ShowEditor();
CheckEdit edit = view.ActiveEditor as CheckEdit;
if (edit == null)
    return;
edit.Toggle();
view.CloseEditor(); // call this method if you want to keep the view scrollable using the mouse wheel
DXMouseEventArgs.GetMouseArgs(e).Handled = true;
}

```

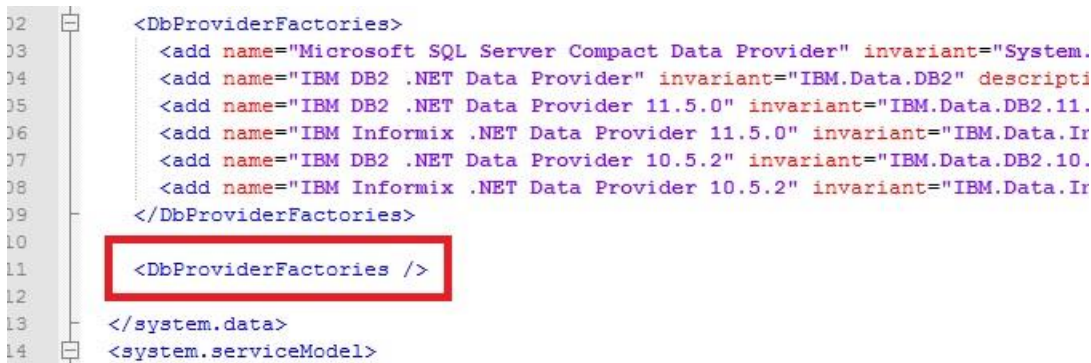
## Driver IBM e Database

### Problema Login Programmi con Entity Framework Driver IBM 11.5 + Windows 7

Rilevato su pc con windows 7 dopo l'installazione del driver IBM 11.5, il pc dava sempre login errato...

SOLUZIONE: verificare che nel file al percorso sottostante non sia presente duplicato il tag <DbProviderFactories>, basta rimuovere il tag vuoto.

C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config\machine.config



```
12 <DbProviderFactories>
13   <add name="Microsoft SQL Server Compact Data Provider" invariant="System.
14   <add name="IBM DB2 .NET Data Provider" invariant="IBM.Data.DB2" descripti
15   <add name="IBM DB2 .NET Data Provider 11.5.0" invariant="IBM.Data.DB2.11.
16   <add name="IBM Informix .NET Data Provider 11.5.0" invariant="IBM.Data.Ir
17   <add name="IBM DB2 .NET Data Provider 10.5.2" invariant="IBM.Data.DB2.10.
18   <add name="IBM Informix .NET Data Provider 10.5.2" invariant="IBM.Data.Ir
19 </DbProviderFactories>
10
11 <DbProviderFactories />
12
13 </system.data>
14 <system.serviceModel>
```

Eccezione lanciata dalla dll Erg.ErgApplication è la seguente:



Database=eb;User ID=...,Password=...,Server=eb-ix:9089;IsolationLevel=ReadCommitted;Pooling=false;  
System.Configuration.ConfigurationErrorsException: La sezione 'DbProviderFactories' può essere presente una sola volta in ogni file di configurazione.  
(C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config\machine.config line 171)  
in System.Configuration.BaseConfigurationRecord.EvaluateOne(String[] keys, SectionInput input, Boolean isTrusted, FactoryRecord factoryRecord, SectionRecord sectionRecord, Object parentResult)  
in System.Configuration.BaseConfigurationRecord.Evaluate(FactoryRecord factoryRecord, SectionRecord sectionRecord, Object parentResult, Boolean getLkg, Boolean getRuntimeObject, Object& result, Object& resultRuntimeObject)  
in System.Configuration.BaseConfigurationRecord.GetSectionRecursive(String configKey, Boolean getLkg, Boolean checkPermission, Boolean getRuntimeObject, Boolean requestIsHere, Object& result, Object& resultRuntimeObject)

Rif.

<https://stackoverflow.com/questions/13822312/the-dbproviderfactories-section-can-only-appear-once-per-config-file>

<https://www.ibm.com/support/pages/dbproviderfactories>

Problema .NET (Anche ergmen2) i programmi devono essere lanciati come amministratore e 'LoaderLock' (rif. Matteo / Gianluca) (Server EUROBEVANDE EB.SRV e nuovo PC Massimiliano macchina fisica)

Il driver IBM per EF **NON** deve essere installato come amministratore

Opz. 1

Reinstallare il driver lanciando il setup senza eseguirlo come admin.

Opz. 2

C:\ProgramData\IBM\DB2\DB2COPY1\DB2

Dare permessi "Accesso Completo" a Everyone ai file:

- db2system
- db2diag.log

(nel server un programma andava in crash perché DB2 non riusciva a scrivere nel file di log)

## Problema ergmen2 e (forse ergmen3) errore nel lanciare alcuni prog. VB6 se non lanciato come amministratore

(rif. Gianluca)

- Da cmd (come amministratore) lanciare `psexec.exe -i -d -s "c:\windows\regedit.exe"`
- Chiave di registro: `HKEY_CURRENT_USER > SOFTWARE (> Wow6432Node) > ODBC > ODBC.INI`
- scorri le fonti odbc e vedi se c'è qualcuno senza autorizzazioni e gliele imposti

## Problema LOGIN e VERSIONE dentro file .edmx

(rif. Nicola, Matteo)

Vedere guida nuovo ergdis > Operazioni preliminari > Punto 4

## Problema tabelle create da utente diverso da erg

E' emerso che EF dà un errore quando un programma sviluppato, ad esempio, sul database test viene eseguito su un altro database e va a scrivere su una tabella con proprietario diverso da quello trovato in fase di progettazione. Di fatto quello che succede è che quando viene creato il modello viene salvato anche il proprietario della tabella all'interno del file .edmx (da aprire con un editor di testo) all'interno dell'attributo "Schema"; quando poi EF lancia un query per scrivere sul db usa:

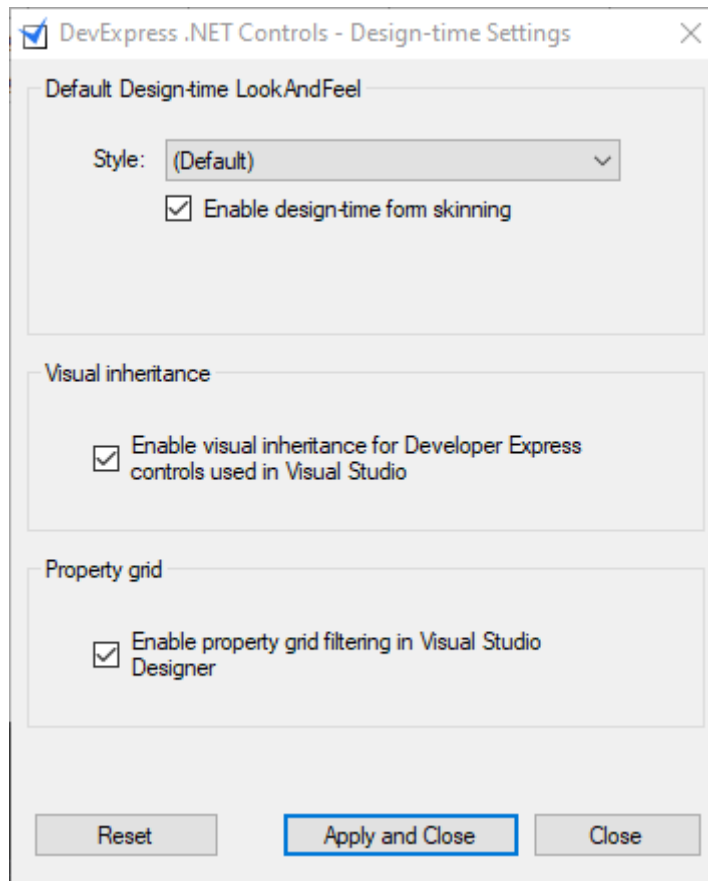
```
INSERT INTO {proprietario}.{nome tabella}
```

e quindi va chiaramente in errore se su un altro database la tabella è stata creata da un altro utente. Il metodo più veloce che abbiamo trovato per risolvere la cosa è lanciare un query di update su systables in modo da uniformare il valore del campo owner su tutti i database, lasciando però intatte le tabelle di sistema. Tale query va lanciata con l'utente informix.

## Visual Studio

I componenti grafici ereditati risultano bloccati nella finestra di progettazione (esempio BaseRic -> maingrid)

Visual Studio => DevExpress => WinForms Controls vX.X.X => Change Design-Time settings => **Enable visual inheritance for Developer Express controls used in Visual Studio**



## Impossibile installare pacchetti NuGet

Nel caso Visual Studio dovesse generare il seguente errore durante l'installazione dei pacchetti attraverso NuGet:

*Library not registered. (Exception from HRESULT: 0x8002801D (TYPE\_E\_LIBNOTREGISTERED))*

È causato dall'accesso negato ad alcune chiavi di registro.

Il percorso delle chiavi in RegEdit che causano l'errore è il seguente:

`HKEY_CLASSES_ROOT > TypeLib`

È possibile individuare le chiave senza permessi scorrendo l'intera lista con le freccette della tastiera dove comparirà un messaggio di errore, o alternativamente, utilizzando il software Process Monitor e andando ad installare i pacchetti in NuGet. Il software rileverà tutte le richieste negate.

È necessario dare i permessi di accesso agli utenti desiderati alle chiavi individuate oppure andando ad impostare al livello della cartella TypeLib i permessi di accesso.

Per intervenire sulle chiavi è necessario usare il tool *psexec* (di PsTools) avviandolo da un prompt dei comandi con permessi di amministratore e avviare *regedit*

```
psexec.exe -i -d -s "c:\windows\regedit.exe"
```

Riferimenti:

- <http://devopswork.blogspot.com/2014/09/library-not-registered-exception-from.html>
- Process Monitor è un tool fornito da Microsoft e consente di vedere tutti gli accessi negati alle chiavi di registro

#### Errore Visual Studio ContextSwitchDeadlock

Nel caso Visual Studio dovesse generare il seguente errore durante un'operazione che prevede l'esecuzione di un gran numero di istruzioni (o in generale che richiedono una certa quantità di tempo). Esempio esecuzione 500 000 query durante un'importazione dati. Oppure l'eliminazione di tutti i record di una tabella tramite EF (esegue le query di eliminazione una per una).

Visual Studio in DEBUG va a "valutare" le funzioni e superato il tempo da lui previsto blocca l'esecuzione del programma lanciando un'eccezione del tipo *ContextSwitchDeadlock*.

SOLUZIONE:

**Debug > Exceptions (oppure Ctrl+Alt+E)**

**> Managed Debugging Assistants > Disabilitare ContextSwitchDeadlock**