



Personalizzare il CRM con SDK e Simple# 2.0

Validità: Giugno 2019

Questa pubblicazione è puramente informativa.
impresoft non offre alcuna garanzia, esplicita od implicita, sul contenuto.
I marchi e le denominazioni sono di proprietà delle rispettive società.

SOMMARIO

1. INTRODUZIONE	3
1.1 PRESENTAZIONE	3
1.2 REQUISITI	3
1.3 POSSIBILITÀ DI PERSONALIZZAZIONE	4
2. SIMPLE# AVANZATO	6
2.1 ESEMPI APPLICATIVI DI SIMPLE#	6
2.1.1 MULTI HOME PAGE	6
2.1.2 PERSONALIZZARE LINK VERSO PAGINE DEL CRM O VERSO PAGINE ESTERNE	10
2.1.3 CREARE NUOVE PAGINE DEL CRM	12
2.1.4 RENDERE OGGETTI SIMPLE# COLLEGABILI TRA LORO	14
2.2 RILASCIO IN PRODUZIONE DI SIMPLE#	36
3. SDK	37
3.1 CONFIGURAZIONE SDK	37
3.2 ESEMPI APPLICATIVI DI SDK	38
3.2.1 PLUGIN TRIGGER	38
3.2.2 PLUGIN STEP PERSONALIZZATO	39
3.2.3 PLUGIN STEP "GESTIONE OFFERTE" PERSONALIZZATO	46
COME NASCONDERE O SPOSTARE I CONTROLLI UTENTE DI UNO STEP	52
3.3 DEBUG E TEST	56
3.4 RILASCIO IN PRODUZIONE DI SDK	56
4. CONCLUSIONI	57

1. Introduzione

1.1 Presentazione

Il presente documento ha lo scopo di illustrare le modalità di personalizzazioni del CRM utilizzando i moduli SDK e il nuovo Simple# 2.0.

1.2 Requisiti

Per una corretta comprensione delle tematiche trattate sono necessari i seguenti requisiti:

- Conoscenza approfondita di Simple#
- Conoscenza CRM base
- Conoscenza dello strumento di sviluppo Visual Studio .Net e del linguaggio C#
- Competenze di SQL, Html, Javascript
- Conoscenza del database del CRM (possibile attraverso la documentazione fornita come file CHM)

Per poter mettere in pratica gli esempi forniti, è necessario avere a disposizione il seguente ambiente dedicato per programmatore, comprendente da:

- Visual Studio .Net 2012 Professional Edition (con installato linguaggio C#)
- CRM (licenziato per modificare il workflow) e con licenza SIMPLE# e SDK (codici 15 e 33)
- SQL server 2005 o 2008 (con database del CRM)

Ad integrazione del seguente documento, viene rilasciata una cartella applicativa comprendente i sorgenti per le personalizzazioni illustrate nel seguito.

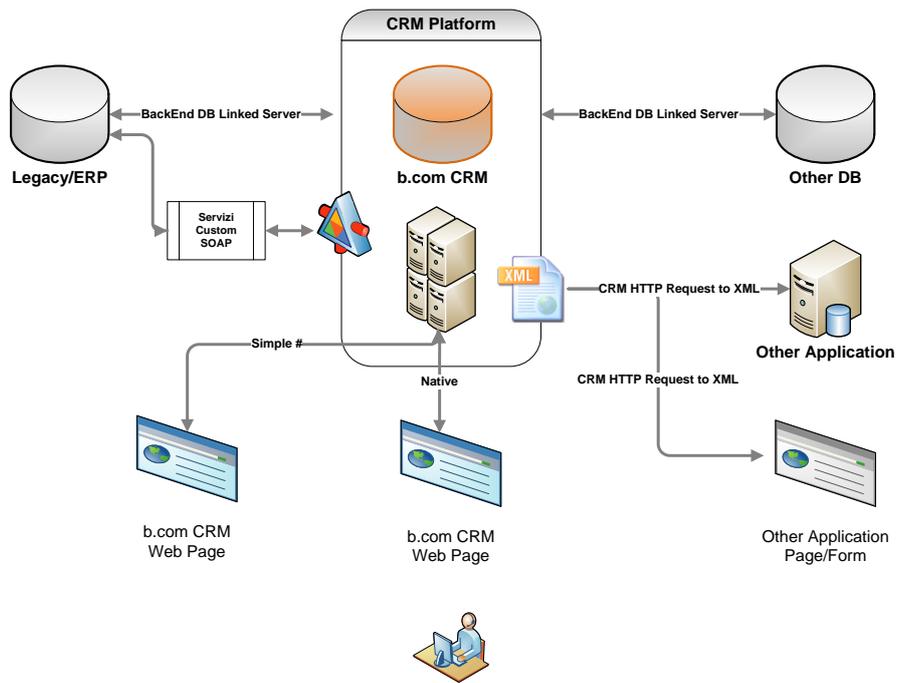
1.3 Possibilità di personalizzazione

Il CRM mette a disposizione diverse possibilità di personalizzazione del prodotto.

	Descrizione	Possibilità	Livello Difficoltà	Requisiti	Stumenti
Profili: funzionalità	Consente di definire i profili di accesso al sistema secondo le esigenze e la privacy	Creare gruppi di utenti, creare utenti, assegnare funzionalità e relativi permessi (es. Sola lettura, possibilità di esportare, ecc.)	Utente	*	Gestione Utenti Gestione Gruppi
Profili: anagrafiche	(opzionale) Consente di definire le regole di visibilità delle anagrafiche	Nelle realtà aziendali più semplici è possibile adottare la regola "tutti vedono tutto". Nelle organizzazioni commerciali più articolate è invece possibile impostare filtri normali di visibilità (Es. Venditori vedono solo il proprio portafoglio, visibilità di gerarchia,...) fino ad adottare regole di visibilità avanzate.	Utente	*	Parametri Filtri Avanzati
Stati	Nel sistema esistono molte tabelle di "Stato" che devono essere impostate.	Si consiglia di vedere la Guida Rapida all'uso per definire al meglio e secondo le proprie esigenze le varie tabelle di Stato, in particolare gli stati dei Contatti e del Backoffice.	Utente	*	..diverse.. Tabelle Di Stato
Parametri	Parametri della procedura	Il sistema consente di modificare molti comportamenti semplicemente agendo sui parametri della procedura	Utente	*	Gestione Parametri
Workflow	E' il workflow di gestione delle attività	Consente di definire i singoli possibili passi per gestire al meglio una attività. Vengono forniti già precompilati. Se è attivo il modulo di Marketing è possibile avere un workflow differente per ogni campagna. I Workflow possono essere salvati come modelli.	Utente	*	Gestione Campagne /Workflow
Stampe	Configurazione dei report.	E' possibile modificare e creare ulteriori report, che possono poi essere collegati alle varie funzionalità in base al gruppo di utenti. Si veda il manuale di personaliz. Stampe.	Reporting Manager	*, Crystal Report	Config. Stampe
Job Scheduler	Configurazione di task schedulati.	E' possibile creare dei comandi schedulati che aggiornano gli archivi applicando regole di business, oppure inviano MAIL ed SMS in modo automatico.	DBA	DB, SQL	Comandi Schedulati dall'ambiente Client /Server
Flussi XML	GetCRMInfo.aspx	Si veda il manuale Recupero dati dal CRM in formato XML	DBA	DB, SQL, XML	Comandi Schedulati
Simple#	Macrolinguaggio interno.	Modifica interfaccia del sistema in alcuni punti specifici.	Utente Avanzato + DBA	DB, SQL, HTML, JS	Editors Simple#
SDK	Software Development Kit	Consente di modificare il comportamento dell'applicazione ed aggiungere ulteriori funzionalità realizzato dei plug-in e step-custom	Sviluppato re	VS .Net, C#	Visual Studio, C# + SDK

* = è richiesta soltanto la conoscenza logica dell'applicativo e l'identificazione precisa dell'esigenza utente da assolvere. Con l'ausilio dell'help e i permessi corretti anche l'utente può personalizzare e configurare il sistema.

Di seguito viene ripreso lo schema generale per le integrazioni.



2. SIMPLE# Avanzato

E' consigliata la lettura dell'ultima versione del manuale Simple# prima di proseguire (presente nella cartella applicativa fornita con la seguente documentazione).

2.1 Esempi applicativi di Simple#

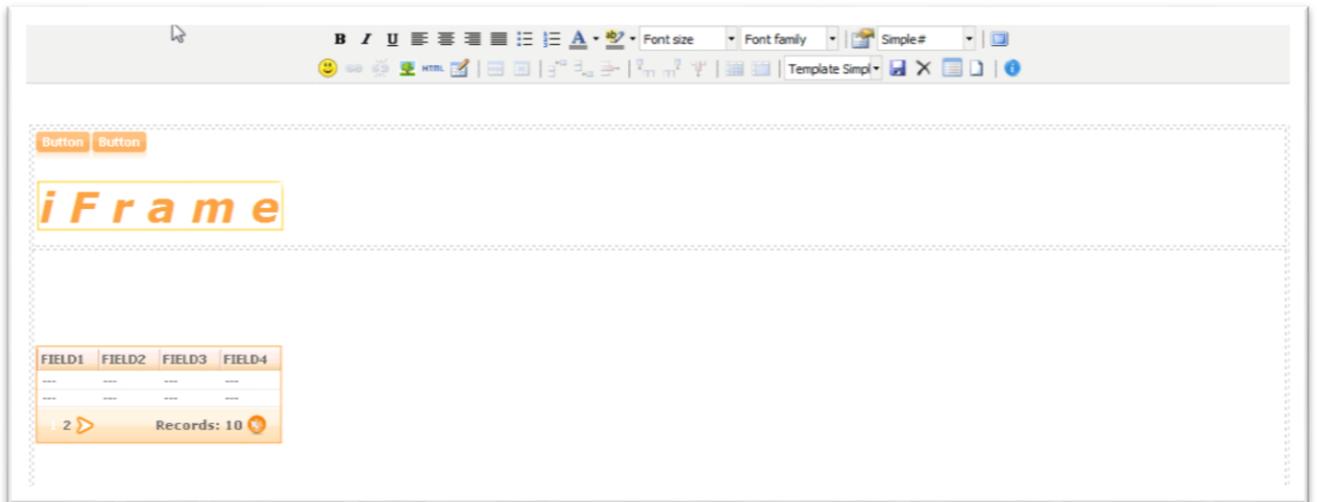
2.1.1 Multi home page

E' possibile trasformare l'home page Simple# in un elegante cruscotto: cliccando su dei bottoni Simple# dedicati, sarà possibile visualizzare la relativa schermata (contenente oggetti Simple#) evitando di far apparire le barre di scorrimento laterali della pagina.

IN CARICO A	CLIENTE	ANNO	N. OPPORTUNITÀ	IMPORTO	% CHIUS.	CHIUS. PREVISTA	STATO
Ugolini Rolando	TELECOM S.p.A.	2011	453_094903508	1800	25	Luglio 2011	TRATTATIVA
Ugolini Rolando	Ugolini Daniele	2010	313_110121111	1581,86	60	Febbraio 2011	TRATTATIVA
Ugolini Rolando	Ugolini Daniele	2010	302_100024156	1500	60	Febbraio 2011	TRATTATIVA
Ugolini Rolando	Ugolini Daniele781	2010	302_100024156	1500	60	Febbraio 2011	VINTA
Ugolini Rolando	Ugolini Daniele781	2010	308_102949123	1500	60	Febbraio 2011	TRATTATIVA
Ugolini Rolando	Ugolini Daniele	2010	316_144605547	1000	80	Settembre 2011	TRATTATIVA
Ugolini Rolando	Ugolini Daniele781	2010	221_155011983	292,31	80	Dicembre 2010	TRATTATIVA
Ugolini Rolando	Ugolini Daniele781	2010	216_125200567	100	80	Luglio 2011	TRATTATIVA
Ugolini Rolando	Ugolini Daniele781	2010	151_104236928	60	80	Giugno 2011	TRATTATIVA
Ugolini Rolando	Ugolini Daniele781	2010	152_104442465	60	10	Gennaio 2011	TRATTATIVA
Ugolini Rolando	Alfa Lum	2010	304_100137576	0	60	Febbraio 2011	TRATTATIVA

OGGETTO	DATA INS	DATA ULTIMA MOD	UTENTE ULTIMA MOD	TIPO	INIZIO SCAD	FINE SCAD	ENTITA'	COD ENTITA'	DESC ENTITA'
	17/05/2010 18.08.19	17/05/2010 18.08.19	ADMIN	Inviare Brochure			CLIENTI	42	dddddddddddddd
seconda prova - cose da fare	17/05/2010 18.08.13	17/05/2010 18.08.13	ADMIN	Senfire Collega			CLIENTI	42	dddddddddddddd
prova	17/05/2010 18.08.06	17/05/2010 18.08.06	ADMIN	Senfire Collega			CLIENTI	42	dddddddddddddd
	17/05/2010 15.40.45	17/05/2010 18.06.16	ADMIN	Stampa Contratto	18/05/2010 09.55.00	18/05/2010 14.25.00			
	17/05/2010 15.40.14	17/05/2010 15.40.14	ADMIN						

I controlli Simple# utilizzati sono i seguenti:

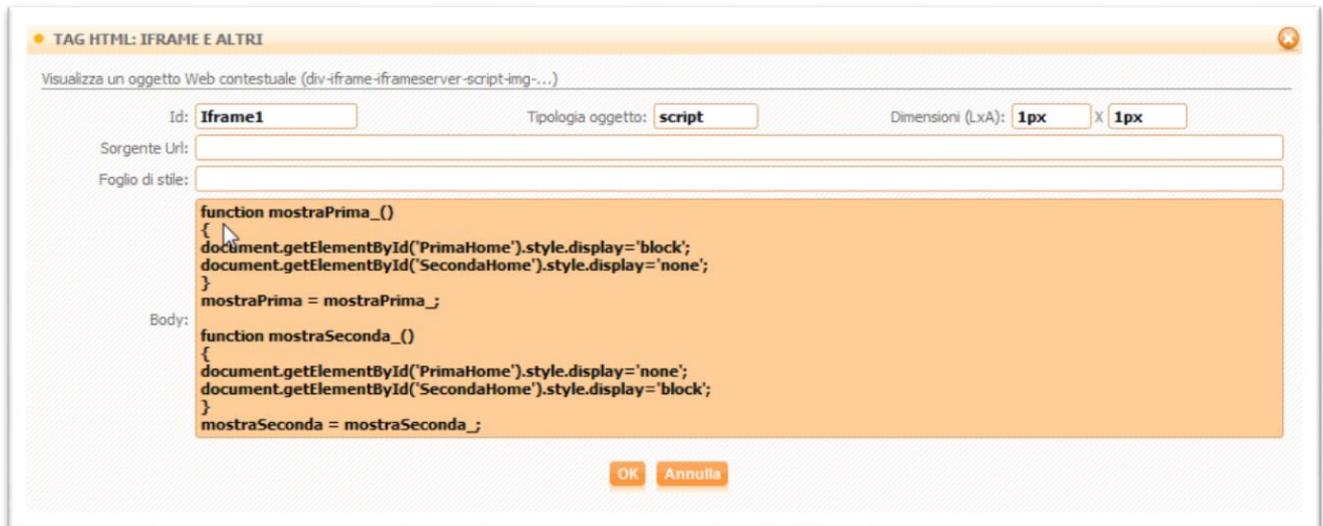


- 2 bottoni
- iFrame
- griglia

Il primo bottone visualizza la prima griglia attraverso la chiamata allo script javascript *mostraPrima()* mentre il secondo bottone mostra la seconda griglia attraverso la chiamata allo script javascript *mostraSeconda()*.



Entrambi gli script sono registrati nel controllo iFrame:



Gli script lavorano sulla proprietà *display* dei controlli "PrimaHome" e "SecondaHome" che rappresentano le due TR della tabella principale:

```
<tr id="PrimaHome" style="width: 100%; height: 100%;">
  <td>
    <p>
      
    </p>
  </td>
</tr>
<tr id="SecondaHome" style="width: 100%; height: 100%; display: none;">
  <td>
    <p>
      
    </p>
  </td>
</tr>
```

Per testare il precedente esempio, è possibile copiare e incollare il codice sorgente Simple#:

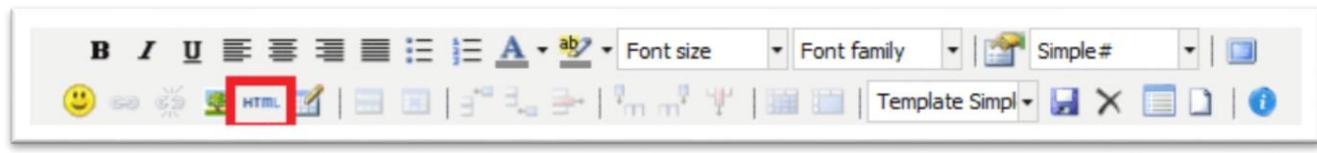
```
<table style="width: 100%; height: 100%;" border="0">
<tbody>
<tr>
<td>
<p></p>
</td>
</tr>
<tr id="PrimaHome" style="width: 100%; height: 100%;">
<td>
<p></p>

</td>  
</tr>

<tr id="SecondaHome" style="width: 100%; height: 100%; display: none;">  
<td></td>

</tr>  
</tbody>  
</table>

Si fa notare che la seconda griglia non è visibile nella modalità design del Simple# per la presenza della proprietà display posta al valore **none** (per poterla visualizzare almeno in formato HTML, cliccare sul bottone "Edit HTML Source" nell'editor del Simple#).



## 2.1.2 Personalizzare link verso pagine del CRM o verso pagine esterne

Il CRM offre la possibilità di personalizzare l'utilizzo di pagine più ricorrenti oppure l'utilizzo di applicazioni esterne (link verso siti preferiti) oppure pagine facenti parte di applicazioni corporate/istituzionali. L'esempio seguente illustra alcune modalità possibili:



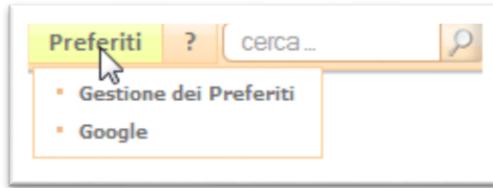
Attraverso l'utilizzo di un bottone Simple# la cui pressione apre una nuova finestra del browser predefinito navigando verso l'indirizzo URL indicato nel campo "Azione".



Attraverso la definizione di tag HTML a (anchor):

```
Esa Software
```

Attraverso l'utilizzo dei Preferiti (accessibili dall'home page sotto l'omonima applicazione oppure dalla pagina delle "Opzioni"):



**Opzioni** | Utente: 000003 | Rolando Ugolini [crm24]

GENERALE 
  IMPOSTAZIONI 
  CTI 
  EMAIL/SMS 
  PARAMETRI PROCEDURA 
  LOG OPE. UTENTI 
  REGISTRO IN/OUT 
  SIMPLE# HOMEPAGE

**Intestazione**

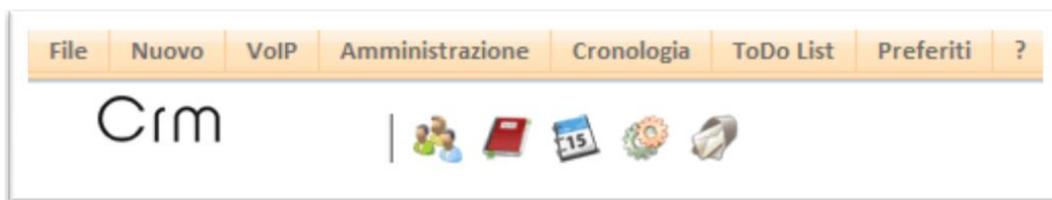
NOME AZIENDA: \_\_\_\_\_  
 Indirizzo completo: \_\_\_\_\_  
 P.Iva / Cod. Fisc.:   
 Telefono: \_\_\_\_\_  
 Timer per verifica Banner/Promemoria:  0 = nessuna verifica  
 Funzione predefinita: \_\_\_\_\_  
 Avvisa controllo Backup Automatici (solo Admin Tools):   
 Avvisa quando le dimensioni dell'archivio superano i:  MB - 0 non avvisa (solo Admin Tools)  
 Avvisa quando il LOG supera il numero di righe:  NR - 0 non avvisa (solo Admin Tools)

|            |                                     |                                                   |
|------------|-------------------------------------|---------------------------------------------------|
| Preferito1 | <input type="text" value="Google"/> | <input type="text" value="http://www.google.it"/> |
| Preferito2 | <input type="text"/>                | <input type="text"/>                              |
| Preferito3 | <input type="text"/>                | <input type="text"/>                              |
| Preferito4 | <input type="text"/>                | <input type="text"/>                              |

Attraverso la personalizzazione del parametro ELU605 nella pagina dedicata ai "Parametri della procedura":

| CODICE | DESCRIZIONE PARAMETRO                                                                                                                                | VALORE              | PAR. A LIVELLO UTENTE | GRUPPO | UTENTE ULTIMA MOD. | ULTIMA MODIFICA     | * | DEFAULT     |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|-----------------------|--------|--------------------|---------------------|---|-------------|
| ELU605 | Gestione "Preferiti" in HomePage: inserire il codice funzione (Gestione utenti -> Funzioni Disponibili) separati da virgola es.(300,515,622,670,100) | 300,515,622,670,100 | ADMIN                 | SIST   | ADMIN              | 14/03/2011 09.02.37 | N | 300,515,622 |

Inserendo il codice funzione (Gestione utenti→Funzioni Disponibili) separati da virgola (per es. 300,515,622,670,100) si otterrebbe:

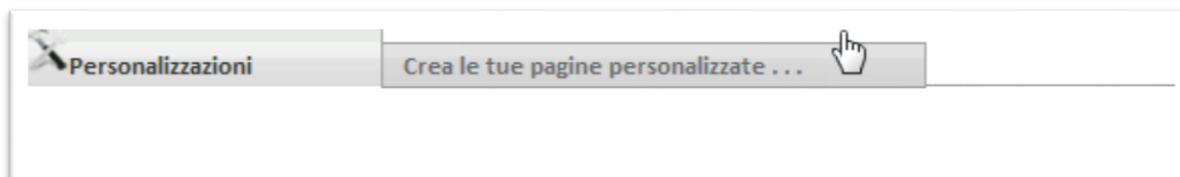


Per testare l'esempio precedente utilizzare il seguente codice e incollarlo in un editor html Simple# :

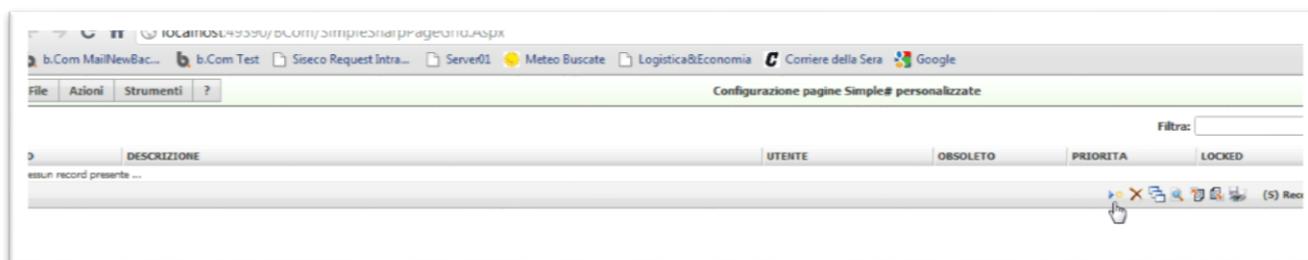
```
<table border="1">
<tbody>
<tr>
<td>Usare bottone Simple# </td>
<td></td>
</tr>
<tr>
<td>Usare tag "a" html</td>
<td>Esa Software</td>
</tr>
<tr>
<td>Usare Gestione Preferiti</td>
<td>Gestione Preferiti</td>
</tr>
<tr>
<td>Usare parametro ELU605</td>
<td>Parametro ELU605</td>
</tr>
</tbody>
</table>
```

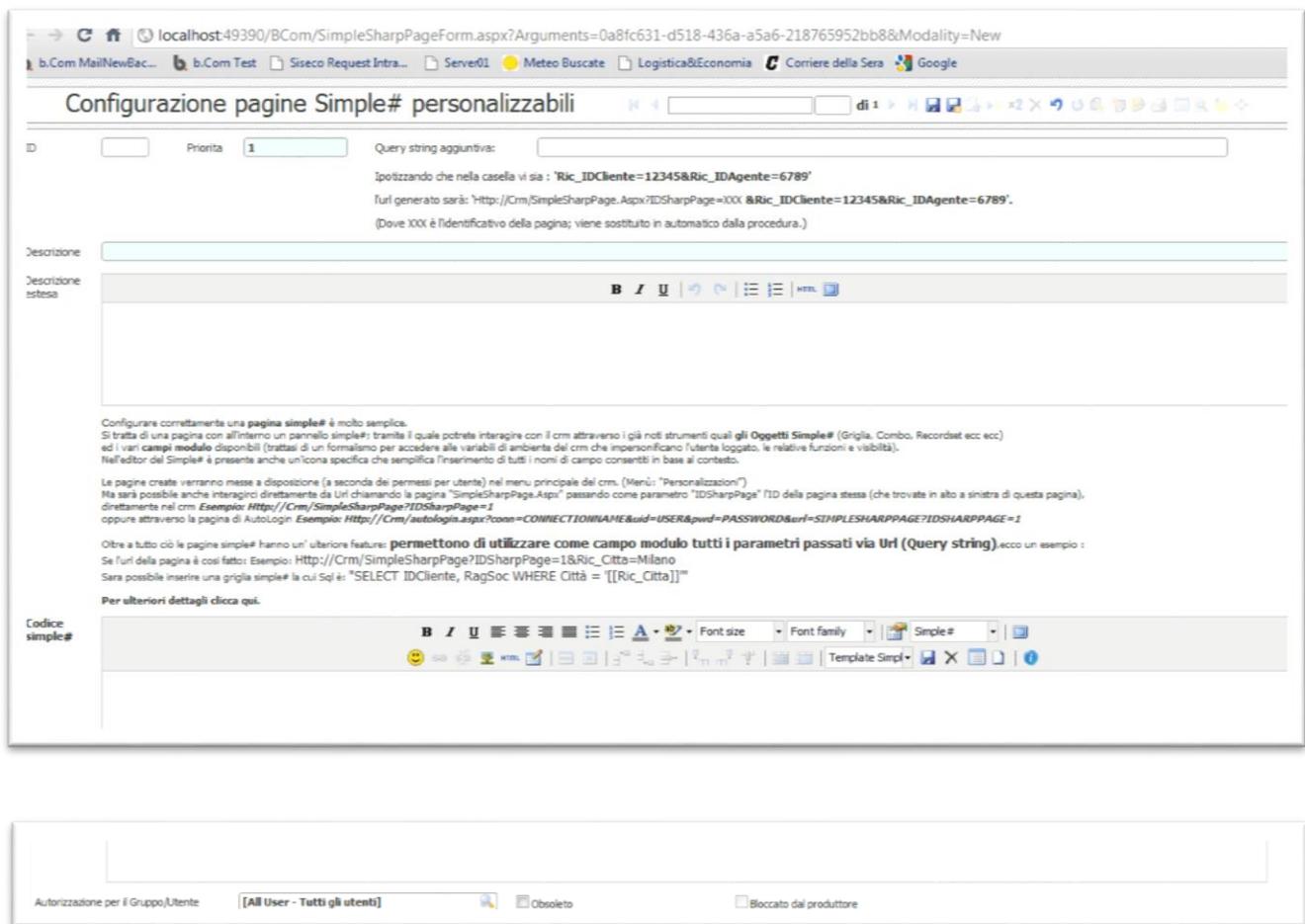
### 2.1.3 Creare nuove pagine del CRM

È possibile creare pagine personalizzate interamente disegnate sfruttando il macro linguaggio Simple# dal menù principale → Personalizzazioni → Crea le tue pagine personalizzate (la creazione di pagine custom andrebbe naturalmente dedicata solamente a particolari utenze del CRM lavorando sui privilegi della funzionalità Gestione Utenti).



La creazione delle pagine personalizzate viene presentata attraverso il designer del Simple# :





Sfruttando quindi le funzionalità e il designer del Simple#, configurare correttamente una pagina Simple# è molto semplice.

Le pagine create verranno messe a disposizione (a seconda dei permessi per utente) nel menu principale del CRM (Personalizzazioni), ma sarà possibile anche utilizzarle direttamente tramite browser richiamando la pagina "SimpleSharpPage.aspx" e passando come parametro in querystring "IDSharpPage" pari all'ID della pagina stessa, oppure "GUIDSharpPage" indicando il GUID della pagina.

Ad esempio:

<http://serverCrm/Crm/SimpleSharpPage?IDSharpPage=1>

oppure attraverso la pagina di AutoLogin:

<http://serverCrm/Crm/autologin.aspx?conn=CONNECTIONNAME&uid=USER&pwd=PASSWORD&url=SIMPLESHARPPAGE?IDSHARPPAGE=1>

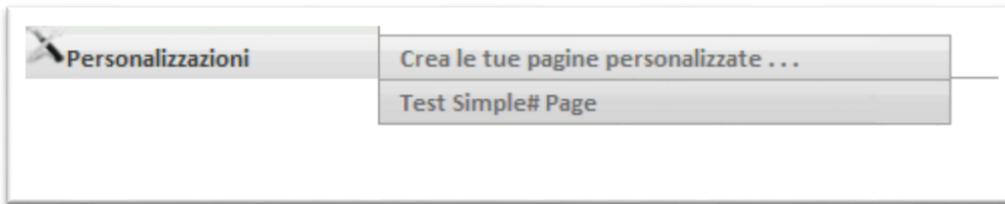
Inoltre, le pagine personalizzate Simple# permettono di utilizzare come campo modulo tutti i parametri passati via querystring alla pagina;

ad esempio se l'URL di chiamata alla pagina fosse:

[http://serverCrm/Crm/SimpleSharpPage?IDSharpPage=1&Ric\\_Citta=Milano](http://serverCrm/Crm/SimpleSharpPage?IDSharpPage=1&Ric_Citta=Milano)

sarà possibile inserire una griglia Simple# con la proprietà Sql pari a:

SELECT IDCliente, RagSoc WHERE Città = '[[Ric\_Citta]]'



**Test Simple# Page**

**Pagina di esempio Simple#**

IDTELEFONATA	IDELERCO	CAMPAGNA	OPERATORE	CONVERSAZIONE	DATAORAFINELAVORO	DURATA	DESCRIZIONE/STATO
411	2005	Campagna TT	Ugolini Rolando	Cambio stato Segnalazione n° 000002 da Da Fare a Chiuso	11/03/2011 13.00.40	0	Creazione Opportunità/Offerta
410	2005	Campagna TT	Ugolini Rolando	Cambio stato per l'Attività n° 3 da Aperto a Chiuso	11/03/2011 12.58.29	0	Creazione Opportunità/Offerta
409	2005	Campagna TT	Ugolini Rolando	Cambio stato per l'Attività n° 2 da Aperto a Chiuso	11/03/2011 12.57.13	0	Creazione Opportunità/Offerta
408	2005	Campagna TT	Ugolini Rolando	Ugolini Rolando - 11/03/2011 - Consulenza - Aperto - TYPICA OPERAZIONE A	11/03/2011 12.56.34	0	Creazione Opportunità/Offerta
407	2005	Campagna TT	Ugolini Rolando	Ugolini Rolando - 11/03/2011 - Consulenza - Aperto - TYPICA OPERAZIONE A	11/03/2011 12.54.47	0	Creazione Opportunità/Offerta
406	2005	Campagna TT	Ugolini Rolando	Ugolini Rolando - 000002 - Hardware - la linea asd non risponde. lu	11/03/2011 12.52.03	0	Creazione Opportunità/Offerta
405	1988	Relazioni con i clienti	Ugolini Rolando	Cambio stato Segnalazione n° 000001 da Da Fare a Chiuso	11/03/2011 12.22.05	0	Creazione Opportunità/Offerta
404	1988	Relazioni con i clienti	Ugolini Rolando	Ugolini Rolando - 11/03/2011 - Consulenza - Aperto - TYPICA OPERAZIONE A	11/03/2011 12.16.54	0	Creazione Opportunità/Offerta
403	2003	test personalizzazione step	Ugolini Rolando	Inserimento automatico	08/03/2011 09.49.03	0	d
402	1982	test presa ordine	Ugolini Rolando	asasasas	03/03/2011 09.24.54	429	Attività generica CRM

(5) Record: 103

Le pagine Simple# personalizzate vengono salvate nella tabella del database del CRM chiamata SintesiCrm (con valore della colonna Entità pari a P per differenziarle dalle Sintesi CRM e dai Desktop Operativi).

## 2.1.4 Rendere oggetti Simple# collegabili tra loro

Vengono di seguito riportati 3 esempi di estensione del concetto di collegamento tra oggetti Simple#.

### 2.1.4.1 Griglia testata/dettaglio

Con la nuova versione di SIMPLE# 2.0 è possibile creare interfacce di interazione più potenti ed articolate.

Questo nuovo oggetto Simple# permette di aggiornare specifiche aree dello schermo in modo che siano **dependenti** da altri oggetti griglia presenti sulla medesima pagina web.

L'utilizzo tipico consente di creare maschere del tipo: "**TESTATA - DETTAGLIO**".

Il nuovo oggetto all'interno dell'editor si presenta così:



## Le proprietà visive:

**PANNELLO SIMPLE#**

Permette di renderizzare un pannello simple# collegabile agli oggetti griglie simple#

Id:

Il pannello Simple# può essere collegato alle griglie Simple# tramite la proprietà impostabile sulla griglia: "**Pannelli Simple# da aggiornare:**".  
selezionando una singola riga (impostare sulla griglia "**Tipo selezione = Single**") verrà aggiornato il pannello Simple# passandogli tutti i valori delle colonne in base alla riga selezionata.  
Il formalismo per poter accedere a questi parametri all'interno del pannello è il seguente: `[[IDGriglia.NomeColonna | |ValoreDefault]]`, dove:

- Il primo valore (prima del punto) indica l'**ID della griglia testata**
- Il secondo (dopo il punto e prima del carattere pipe) indica **Indica il nome della colonna da cui verrà preso il valore in base alla riga selezionata.**
- Il terzo valore (dopo il carattere pipe) indica il **valore di default nel caso in cui non vi siano righe selezionate nella griglia testata**; è possibile specificare un valore puntuale o una query Sql

---

Esempio realizzazione griglia testata - dettaglio (master - detail)

- Griglia testata con le seguenti proprietà:
  - ID="GridTestata"
  - Sql="SELECT ID, Descrizione FROM MiaTabellaTestata"
  - PannelliSimple#DaAggiornare="SimplePanelDettaglio"
- Pannello Simple# con ID="SimplePanelDettaglio"
- Griglia di Dettaglio dentro il Pannello Simple# con le seguenti proprietà :
  - ID="SimplePanelDettaglio"
  - Sql="SELECT ID, Descrizione FROM MiaTabellaDettaglio Where IDTestata = [[GridTestata.ID | 0]]"

oppure

- Griglia di Dettaglio dentro il Pannello Simple# con le seguenti proprietà :
  - ID="SimplePanelDettaglio"
  - Sql="SELECT ID, Descrizione FROM MiaTabellaDettaglio Where IDTestata IN([[GridTestata.ID | SELECT ID FROM MiaTabellaTestata]])"

---

**Attenzione !**  
In fase di esitazione attività in Multimedia i recordset Simple# contenuti nei pannelli# sono esclusi dal processo di aggiornamento database.

OK Annulla

## Tutte le proprietà dell'oggetto:

```
<Item key="id" size="50" propertyinfo="ID" caption="Id" default="SimpleSharpPanel1" tooltip="Indicare un ID univoco per identificare l'elemento" />
</Items>
```

Vediamo un esempio concreto di utilizzo:

The screenshot shows a web application interface. At the top, there's a toolbar with various icons and a dropdown menu for 'Simple#'. The main content area is divided into two sections. The first section, titled 'TESTATA (ELENCO OFFERTE VISIBILI ALL'UTENTE)', contains a table with columns 'FIELD1', 'FIELD2', 'FIELD3', and 'FIELD4'. Below the table, there's a 'Records: 10' indicator. The second section, titled 'DETTAGLIO (RIGHE OFFERTA SELEZIONATA)', also contains a table with the same columns and a 'Records: 10' indicator. A red box highlights the detail section. To the right, a dropdown menu is open, showing 'Pannello simple#' selected. Below this, a configuration panel for 'PANNELLO SIMPLE#' is shown. It has an ID of 'SimpleSharpPanelDettaglio' and a description: 'Permette di rendere un pannello simple# collegabile agli oggetti griglie simple#'. It also lists properties for linking the panel to the grid, such as 'ID della griglia padre', 'nome della colonna da cui', and 'valore di default nel caso in cui non vi siano'. An example of the grid configuration is also provided.

Il pannello Simple# può essere collegato alle griglie Simple# tramite la proprietà impostabile sulla griglia: **"Pannelli Simple# da aggiornare:"**

The screenshot shows the configuration panel for a grid. The grid is titled 'GRIGLIA' and has an ID of 'GridTestata'. The configuration includes fields for 'Numero di righe' (10), 'Larghezza' (100%), 'Tipo selezione' (Single), 'Modalità' (DataTable), 'Colonna Tooltip' (12), 'Larghezza Colonne' (0;200;0), and 'Pannelli simple# da aggiornare' (SimpleSharpPanelDettaglio). The SQL Query is also visible: 'SELECT Ordini.IDOrdine, TIPI\_ORDINI.Descrizione as Tipo, Ordini.NumeroOrdine, dbo.dateformat(Ordini.DataOrdine, 'dd-mm-yy ddd') as Data, Ordini.Cliente as A, Ordini.OggettoTrattativa as Oggetto, (SELECT (CodiceAgente + ' - ' + Cognome) Ordini.IDAgente) As Agente, Stati.DescrizioneStato AS Stato, Ordini.Percentuale JOIN Stati ON Ordini.Stato = Stati.IDStato INNER JOIN TIPI\_ORDINI ON TIPI\_ORDINI.IDTipo = Ordini.IDTipo'. The 'Pannelli simple# da aggiornare' field is highlighted with a blue arrow.



Sul click di una singola riga della griglia che costituisce "la testata" di una maschera testata - dettaglio verrà **aggiornato il pannello Simple#** indicato, passandogli **tutti i valori** delle colonne in base alla riga selezionata.

E' importante sottolineare che la griglia di testata dovrà avere la proprietà "Tipo Selezione = SINGLE" al fine di consentire all'utente la selezione di una riga specifica.

Il formalismo per poter accedere a questi parametri nel pannello è il seguente:

**[[IDGriglia.NomeColonna||ValoreDefault]]**

- Il primo valore (prima del punto) indica l'ID della griglia padre
- Il secondo (dopo il punto prima del pipe) indica il nome della colonna da cui verrà preso il valore in base alla riga selezionata.
- Il terzo valore (dopo il pipe) indica il valore di default nel caso in cui non vi siano righe selezionate nella griglia padre; è possibile specificare un valore puntuale o una query SQL

### **Esempio realizzazione griglia testata - dettaglio**

- Griglia testata con le seguenti proprietà:
  - ID="GridTestata"
  - Sql="SELECT ID, Descrizione FROM MiaTabellaTestata"
  - PennelliSimple#DaAggiornare="SimplePanelDettaglio"
- Pannello Simple# con ID="SimplePanelDettaglio"
- Griglia di Dettaglio dentro il Pannello Simple# con le seguenti proprietà:
  - ID="SimplePanelDettaglio"
  - Sql="SELECT ID, Descrizione FROM MiaTabellaDettaglio Where IDTestata = [[GridTestata.ID||0]]"

#### **oppure**

- Griglia di Dettaglio dentro il Pannello Simple# con le seguenti proprietà:
  - ID="SimplePanelDettaglio"
  - Sql=""SELECT ID, Descrizione FROM MiaTabellaDettaglio Where IDTestata IN([[GridTestata.ID||SELECT ID FROM MiaTabellaTestata]])""

**Attenzione** - In fase di esitazione attività in Multimedia i recordset Simple# contenuti nei pannelli Simple# sono esclusi dal processo di aggiornamento del database.

**Per costruire l'intero Simple# del tipo "TESTATA - DETTAGLIO"** è necessario eseguire queste operazioni:

1. inserire una griglia con tutte le sue proprietà correttamente impostate. Questa costituirà la griglia della testata. Diamogli il nome (ad esempio) di "**GridTestata**"
2. impostare la proprietà "**Tipo Selezione = SINGLE**"
3. impostare la proprietà "**Pannelli Simple# da aggiornare = GridDettaglio**"
4. inserire un Pannello Simple#, facendo doppio click sul bordo del riquadro "rosso" impostare il nome dell'oggetto a "**GridDettaglio**"
5. a questo punto il Pannello Simple# è pronto per contenere tutto il Simple# che si desidera. All'interno del pannello sarà possibile utilizzare la seguente variabile per aggiornare in modo dinamico il contenuto

**[[IDGriglia.NomeColonna||ValoreDefault]]**

Un esempio reale di testata e dettaglio potrebbe essere il seguente: un elenco delle offerte visibili dall'utente, cliccando sulle quali viene visualizzato l'elenco delle righe dell'offerta selezionata.

**TESTATA (ELENCO OFFERTE VISIBILI ALL'UTENTE)** Nuova Offerta...

Doppio click apre

Filtra:

TIPO	NUMERO	REV	DATA	ANAGRAFICA	TEL1	TOT	OGGETTO	AGENTE	STATO	% CHIUS	
Appuntamento	229_150508163		07-01-11 Fri	prova siseco	03319351	0	Trattativa di tipo A	Agente Prova - prova	APPUNTAMENTO PER AGENTE		Apri
Appuntamento	228_150345666		07-01-11 Fri	"BARBERO 1891 - STERZI - ENRICO SERAFINO - RICCADONNA - S.P.A. O IN...	0173967111	41	Trattativa di tipo A	Agente Prova - prova	APPUNTAMENTO PER AGENTE		Apri
Appuntamento	227_150258694		07-01-11 Fri	prova siseco	03319351	0	Trattativa di tipo A	Agente Prova - prova	APPUNTAMENTO PER AGENTE		Apri
Opportunità	226_154204902		30-12-10 Thu	P&F DISTRIBUZIONE ITALIA SRL	029007711	0		Timkt_Simona - Nicola	ADESIONE IN LINEA		Apri
Appuntamento	225_191655728		29-12-10 Wed	"BARBERO 1891 - STERZI - ENRICO SERAFINO - RICCADONNA - S.P.A. O IN...	0173967111	0	Trattativa di tipo B	Agente Prova - prova	APPUNTAMENTO PER AGENTE		Apri
Ordine + Appuntamento	224_120404151		21-12-10 Tue	pippo	01119500676	6406,6	Trattativa di tipo C	Timkt_Simona - Nicola	CHIUSO OK		Apri
Opportunità	223_193519354		15-12-10 Wed	3 BM DI MARIO BEZZINI & C. SNC	0565851041	0		Timkt_Simona - Nicola	ADESIONE IN LINEA	0	Apri
Ordine	222_13511978		15-12-10 Wed	"MORETTONI S.P.A." PER BREVITA' ANCHE CON LE SIGLE "CM S.P.A." E "TENU	0758043514	0		Timkt_Simona - Nicola	ADESIONE IN LINEA	0	Apri
Opportunità	221_16102925		14-12-10 Tue	"MORETTONI S.P.A." PER BREVITA' ANCHE CON LE SIGLE "CM S.P.A." E "TENU	0758043514	0		Timkt_Simona - Nicola	Revisionato	0	Apri
Opportunità	219_123823870		30-11-10 Tue	RAZIONE SOCIALE PER TEST - FRANCESCO	0000111222	0	aaa 1 3 45	Timkt_Simona - Nicola	Revisionato	60	Apri

1 2 3 4 5 6 7 8 9 10 >

(D) Records: 141

**DETTAGLIO (RIGHE OFFERTA SELEZIONATA 228\_150345666)**

NUMRIGA	CODICE	DESCRIZIONE PRODOTTO	QUANTITÀ	UM	PREZZOUNIT	PREZZO	STATO RIGA
5	1234	DESCRIZIONE PRODOTTO A	1	3	3	3	Generico
15	52	DESCRIZIONE PRODOTTO C	10	3	30	30	Generico
10	52	DESCRIZIONE PRODOTTO B	1	3	3	3	Generico

(D) Records: 3

E' importante notare che è possibile l'elenco delle offerte soltanto a quelle visibili "dall'utente" grazie al parametro **[[FiltroVisibilitaClienti]]**, utilizzandolo nella query SQL che alimenta la griglia di testata.

La griglia di testata la chiameremo **"GridTestata"** e le proprietà sono così impostate:

**GRIGLIA**

Visualizza una tabella di informazioni contestuale (utilizzare [[nomecampo]]) - Nel caso delle azioni si può utilizzare anche [[nomecolonna]]  
 Nelle azioni utilizzare il parametro IDPrmKeyTable per aprire entità con la loro chiave principale

Id:  Numero di righe:  Larghezza:  Nascondi:  Visualizza tag HTML:  Abilita Stampa:  Abilita Export:  Abilita Ricerca:

Tipo selezione:  Modalità:  Colonna Tooltip:  Larghezza Tooltip:  Intestazione Tooltip:

Colonna colore:  Codifica colori:  Colonna colore Sfondo:  Codifica colori Sfondo:

SQL Query (usare [[nomecampo]]):  

```
SELECT Ordini.IDOrdine, TIPI_ORDINI.Descrizione as Tipo, Ordini.NumeroOrdine as Numero, Ordini.Revisione as Rev,
dbo.dateformat(Ordini.DataOrdine,'dd-mm-yy ddd') as Data, Ordini.Cliente as Anagrafica, Ordini.Tel1, Ordini.Tot,
Ordini.OggettoTrattativa as Oggetto, (SELECT (CodiceAgente + ' - ' + Cognome) AS Agente FROM Agenti WHERE IDAgente =
Ordini.IDAgente) AS Agente, Stati.DescrizioneStato AS Stato, Ordini.PercentualeChiusura as [% Chius], Ordini.Note FROM Ordini INNER
JOIN Stati ON Ordini.Stato = Stati.IDStato INNER JOIN TIPI_ORDINI ON TIPI_ORDINI.IdTipoOrdine=Ordini.IDTipo WHERE
```

Larghezza Colonne:  Colonne da sommare:

Pannelli simple # da aggiornare:

Messaggio no record:  Codice Stampa:

Azione 1:  Descrizione 1:

Messaggio Conferma 1:

## SQL Query

```
SELECT Ordini.IDOrdine, TIPI_ORDINI.Descrizione as Tipo, Ordini.NumeroOrdine as Numero, Ordini.Revisione as Rev, dbo.dateformat(Ordini.DataOrdine,'dd-mm-yy ddd') as Data, Ordini.Cliente as Anagrafica, Ordini.Tell, Ordini.Tot, Ordini.OggettoTrattativa as Oggetto, (SELECT (CodiceAgente + ' - ' + Cognome) AS Agente FROM Agenti WHERE IDAgente = Ordini.IDAgente) As Agente, Stati.DescrizioneStato AS Stato, Ordini.PercentualeChiusura as [% Chius], Ordini.Note FROM Ordini INNER JOIN Stati ON Ordini.Stato = Stati.IDStato INNER JOIN TIPI_ORDINI ON TIPI_ORDINI.IdTipoOrdine=Ordini.IDTipo WHERE ([[FiltroVisibilitaClienti]]) ORDER BY IDOrdine DESC
```

Inseriamo il pannello Simple# e assegniamo il seguente ID al pannello:

**PANNELLO SIMPLE#**

Permette di renderizzare un pannello simple

Id: **SimpleSharpPanelDettaglio**

Il pannello simple# può essere collegato  
Sul click di una singola riga verrà annor

Il Simple# comparirà così:

TESTATA (ELENCO OFFERTE VISIBILI ALL'UTENTE)  
Doppio click apre

FIELD1	FIELD2	FIELD3	FIELD4
---	---	---	---
---	---	---	---

1 2 > Records: 10

DETTAGLIO (RIGHE OFFERTA SELEZIONATA [[GridTestata.Numero||0]])

FIELD1	FIELD2	FIELD3	FIELD4
---	---	---	---
---	---	---	---

1 2 > Records: 10

La griglia all'interno del pannello è così configurata

**GRIGLIA**

Visualizza una tabella di informazioni contestuale (utilizzare [[nomecampo]] - Nel caso delle azioni si può utilizzare anche |[nomecolonna|])  
 Nelle azioni utilizzare il parametro IDPrmKeyTable per aprire entità con la loro chiave principale

Id: **GridDettaglio** Numero di righe: **10** Larghezza: **100%** Nascondi:  Visualizza tag HTML:  Abilita Stampa:  Abilita Export:  Abilita Ricerca:

Tipo selezione: **Single** Modalità: **DataTable** Colonna Tooltip: **-1** Larghezza Tooltip:  Intestazione Tooltip:

Colonna colore: **-1** Codifica colori:  Colonna colore Sfondo: **-1** Codifica colori Sfondo:

SQL Query (usare [[nomecampo]]):  

```
SELECT TOP 100 PERCENT Dettagli_ordini.IDDettagliOrdini, Dettagli_ordini.IDOrdine,
 Dettagli_ordini.NumRiga, Dettagli_ordini.NumeroOrdineRiga as NumOrdRiga, ListiFR.Codice, Dettagli_ordini.Descrizione as [Descrizione Prodotto],
 Dettagli_ordini.Quantità, Dettagli_ordini.UM, Dettagli_ordini.PrezzoUnit, Dettagli_ordini.Prezzo, Dettagli_ordini.Stati.DescrizioneStato
as [Stato Riga] FROM IDettagli_ordini INNER JOIN Dettagli_ordini_Stati ON Dettagli_ordini.IDStatoDett =
 Dettagli_ordini_Stati.IDStatoDett INNER JOIN ListiFR ON Dettagli_ordini.Codice = ListiFR.Codice WHERE (Dettagli_ordini.IDOrdine =
 [[GridTestata.IDOrdine|0]]) ORDER BY Dettagli_ordini.NumeroOrdineRiga, ListiFR.Descrizione
```

Larghezza Colonne: **0;0;0** Colonne da sommare:

Pannelli simple# da aggiornare:

Messaggio no record: **Nessuna riga** Codice Stampa:

Azione 1: **Ordini\_DettaglioForm.aspx?IDORDINE=|IDordine|**

Descrizione 1: **Apri Righe**

Come è possibile notare la query è parametrica:

```
SELECT TOP 100 PERCENT Dettagli_ordini.IDDettagliOrdini, Dettagli_ordini.IDOrdine,
 Dettagli_ordini.NumRiga, Dettagli_ordini.NumeroOrdineRiga as NumOrdRiga, ListiFR.Codice,
 Dettagli_ordini.Descrizione as [Descrizione Prodotto], Dettagli_ordini.Quantità, Dettagli_ordini.UM,
 Dettagli_ordini.PrezzoUnit, Dettagli_ordini.Prezzo, Dettagli_ordini.Stati.DescrizioneStato as [Stato Riga]
FROM IDettagli_ordini INNER JOIN Dettagli_ordini_Stati ON Dettagli_ordini.IDStatoDett =
 Dettagli_ordini_Stati.IDStatoDett INNER JOIN ListiFR ON Dettagli_ordini.Codice = ListiFR.Codice WHERE
 (Dettagli_ordini.IDOrdine = [[GridTestata.IDOrdine|0]]) ORDER BY Dettagli_ordini.NumeroOrdineRiga,
 ListiFR.Descrizione
```

E verrà visualizza in questo modo:

**DETTAGLIO (RIGHE OFFERTA SELEZIONATA 228\_150345666)**

NUMRIGA	CODICE	DESCRIZIONE PRODOTTO	QUANTITÀ	UM	PREZZOUNIT	PREZZO	STATO RIGA	
5	1234	DESCRIZIONE PRODOTTO A	1	3	3	3	Generico	Apri Righe
15	52	DESCRIZIONE PRODOTTO C	10	3	30	30	Generico	Apri Righe
10	52	DESCRIZIONE PRODOTTO B	1	3	3	3	Generico	Apri Righe

(D) Records: 3

Ecco l'esempio del comportamento finale. Facendo click su una riga delle offerte (testata) si aggiorna l'elenco delle righe sottostante.

**TESTATA (ELENCO OFFERTE VISIBILI ALL'UTENTE)**

Doppio click apre

Nuova Offerta...

Filtra:

TIPO	NUMERO	REV.	DATA	ANAGRAFICA	TEL1	TOT	OGGETTO	AGENTE	STATO	% CHIUSO	
Appuntamento	229_150508163		07-01-11 Fri	prova siseco	03319351	0	Trattativa di tipo A	Agente Prova - prova	APPUNTAMENTO PER AGENTE		Apri
Appuntamento	<b>228_150345666</b>		07-01-11 Fri	"BARBERO 1891 - STERZI - ENRICO SERAFINO - RICCADONNA - S.P.A. O IN...	0173967111	41	Trattativa di tipo A	Agente Prova - prova	APPUNTAMENTO PER AGENTE		Apri
Appuntamento	227_150258694		07-01-11 Fri	prova siseco	03319351	0	Trattativa di tipo A	Agente Prova - prova	APPUNTAMENTO PER AGENTE		Apri
Opportunità	226_154204902		30-12-10 Thu	P&F DISTRIBUZIONE ITALIA SRL	029007711	0		Timkt_Simona - Nicola	ADESIONE IN LINEA		Apri
Appuntamento	225_191655728		29-12-10 Wed	"BARBERO 1891 - STERZI - ENRICO SERAFINO - RICCADONNA - S.P.A. O IN...	0173967111	0	Trattativa di tipo B	Agente Prova - prova	APPUNTAMENTO PER AGENTE		Apri
Ordine + Appuntamento	224_120404151		21-12-10 Tue	pippo	01119500676	6406,6	Trattativa di tipo C	Timkt_Simona - Nicola	CHIUSO OK		Apri
Opportunità	223_193519354		15-11-10 Wed	3 BM DI MARIO BEZZINI & C. SNC	0565851041	0		Timkt_Simona - Nicola	ADESIONE IN LINEA	0	Apri
Ordine	222_13511978		15-12-10 Wed	"MORETTONI S.P.A." PER BREVITA' ANCHE CON LE SIGLE "C.S.P.A." E "TENU"	0758043514	0		Timkt_Simona - Nicola	ADESIONE IN LINEA	0	Apri
Opportunità	221_16102925		14-12-10 Tue	"MORETTONI S.P.A." PER BREVITA' ANCHE CON LE SIGLE "C.S.P.A." E "TENU"	0758043514	0		Timkt_Simona - Nicola	Revisionato	0	Apri
Opportunità	219_123823870		30-11-10 Tue	RAGIONE SOCIALE PER TEST - FRANCIOSO	0000111222	0	aaa 1 3 45	Timkt_Simona - Nicola	Revisionato	60	Apri

2 3 4 5 6 7 8 9 10 >

(D) Records: 141

**DETTAGLIO (RIGHE OFFERTA SELEZIONATA 228\_150345666)**

NUMRIGA	CODICE	DESCRIZIONE PRODOTTO	QUANTITÀ	UM	PREZZOUNIT	PREZZO	STATO RIGA	
5	1234	DESCRIZIONE PRODOTTO A	1	3	3	3	Generico	
15	52	DESCRIZIONE PRODOTTO C	10	3	30	30	Generico	
10	52	DESCRIZIONE PRODOTTO B	1	3	3	3	Generico	

(D) Records: 3

E' possibile copiare questo codice e incollarlo nell'HTML dell'editor Simple# della Home Page del CRM.

```
<table style="width: 1043px; height: 142px;" border="0">
<tbody><tr>
<td style="background-color: #0066ff;">TESTATA (ELENCO OFFERTE VISIBILI ALL'UTENTE)</td>
<td style="background-color: #99ccff;">Nuova Offerta...</td>
</tr>
<tr><td colspan="2">
<p></p>
</td>
</tr></tbody></table>
<table id="SimpleSharpPanelDettaglio" class="SimpleSharpPanel">
<tbody><tr>
<p>DETTAGLIO (RIGHE OFFERTA SELEZIONATA [[GridTestata.Numero|0]]</p>
<p>
</td> </td>
</tr></tbody></table>
```

### 2.1.4.2 Griglia e inserimento/modifica record

La finalità di questo esempio è quella di rappresentare attraverso il linguaggio Simple#, una casistica molto ricorrente negli applicativi gestionali (aspetto spesso ricorrente anche nel CRM) ovvero una griglia che sintetizza in formato tabellare dei dati e dare la possibilità di modificarli in maniera veloce oppure di crearne di nuovi. A scopo didattico si è utilizzata una tabella custom esterna al CRM (eseguire lo script seguente per la creazione).

```

CREATE TABLE [dbo].[TbITestSDK](
 [IDKey] [int] IDENTITY(1,1) NOT NULL,
 [Text1] [nvarchar](max) NULL,
 [Text2] [nvarchar](max) NULL,
 [Text3] [nvarchar](max) NULL,
 [Text4] [nvarchar](max) NULL,
 [Text5] [int] NULL
 CONSTRAINT [PK_TbITestSDK] PRIMARY KEY CLUSTERED
 (
 [IDKey] ASC
)
) ON [PRIMARY]
GO

```

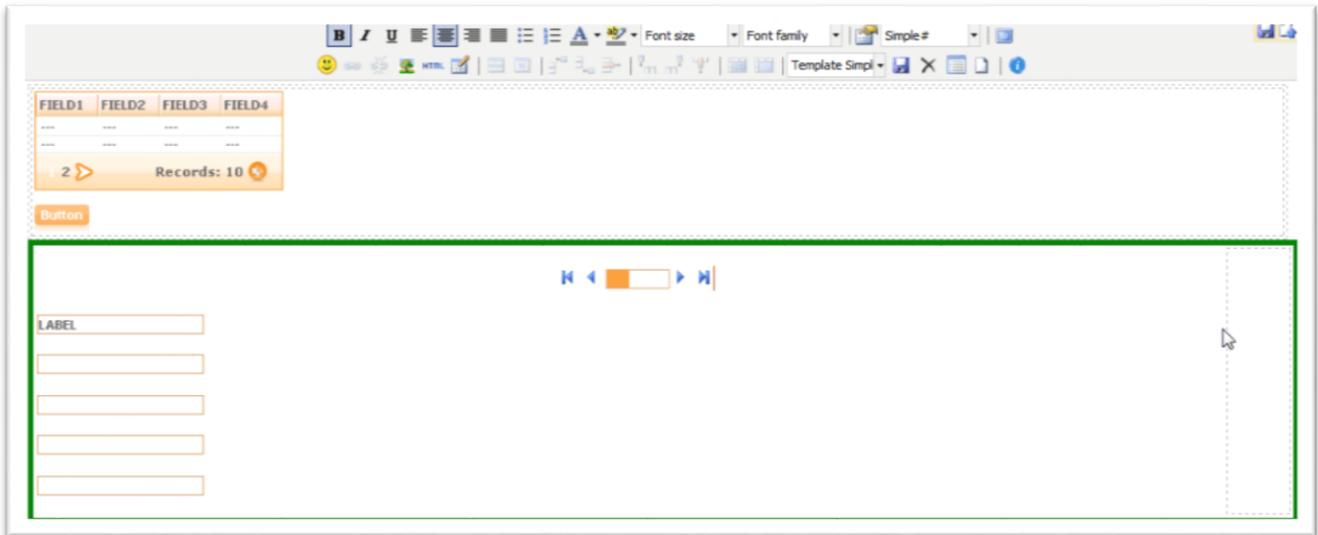
Per testare questo esempio, copiare e incollare il seguente codice nell'editor html del Simple# :

```

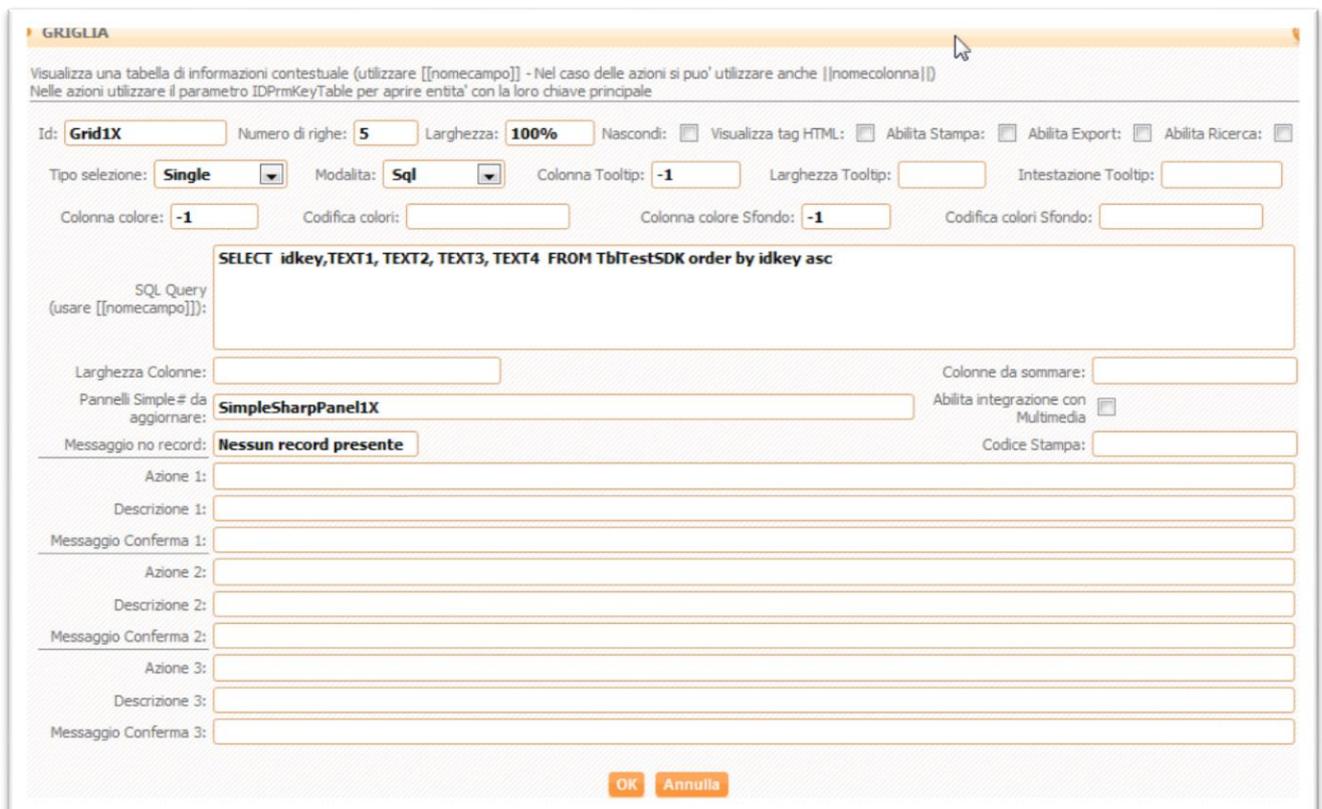
<table style="width: 1084px; height: 127px;" border="0">
<tbody>
<tr>
<td>
<p></p>
<p></p>
</td>
</tr>
</tbody>
</table>
<table id="SimpleSharpPanel1X" class="SimpleSharpPanel">
<tbody>
<tr>
<p style="text-align: center;">
<p>
</p>
<p style="text-align: left;">
<p>
<p></p>
</p>
</p>
</p>
<p></p>
<p></p>
<p></p>
<p></p>
<td> </td>
</tr>
</tbody>
</table>

```

Gli oggetti Simple# utilizzati sono i seguenti:



- una griglia Simple# che visualizza i record della tabella custom creata, dove al click su una singola riga, si tenterà di aggiornare il pannello Simple# sottostante



- un pannello Simple# contenente:
  - o recordset Simple# che punta a un nuovo record in caso di nuovo inserimento oppure alla riga selezionata nella griglia (è stato reso visibile il bottone di salvataggio)
  - o una label Simple# indicante l'operazione che si sta compiendo sul recordset
  - o 4 campi input Simple# per acquisire o modificare dati esistenti

## PANNELLO SIMPLE#

Permette di renderizzare un pannello simple# collegabile agli oggetti griglie simple#

Id:

Il pannello Simple# può essere collegato alle griglie Simple# tramite la proprietà impostabile sulla griglia: "Pannelli Simple# da aggiornare:". selezionando una singola riga (impostare sulla griglia "Tipo selezione = Single") verrà aggiornato il pannello Simple# passandogli tutti i valori delle colonne in base alla riga selezionata. Il formalismo per poter accedere a questi parametri all'interno del pannello è il seguente: `[[IDGriglia.NomeColonna | ValoreDefault]]`, dove:

- Il primo valore (prima del punto) indica l'**ID della griglia testata**
- Il secondo (dopo il punto e prima del carattere pipe) indica **Indica il nome della colonna da cui verrà preso il valore in base alla riga selezionata.**
- Il terzo valore (dopo il carattere pipe) indica il **valore di default nel caso in cui non vi siano righe selezionate nella griglia testata**; è possibile specificare un valore puntuale o una query Sql

Esempio realizzazione griglia testata - dettaglio (master - detail)

- Griglia testata con le seguenti proprietà:
    - ID="GridTestata"
    - Sql="SELECT ID, Descrizione FROM MiaTabellaTestata"
    - PannelliSimple#DaAggiornare="SimplePanelDettaglio"
  - Pannello Simple# con ID="SimplePanelDettaglio"
  - Griglia di Dettaglio dentro il Pannello Simple# con le seguenti proprietà :
    - ID="SimplePanelDettaglio"
    - Sql="SELECT ID, Descrizione FROM MiaTabellaDettaglio Where IDTestata = [[GridTestata.ID] | 0]"
- oppure
- Griglia di Dettaglio dentro il Pannello Simple# con le seguenti proprietà :
    - ID="SimplePanelDettaglio"
    - Sql="SELECT ID, Descrizione FROM MiaTabellaDettaglio Where IDTestata IN([[GridTestata.ID] | SELECT ID FROM MiaTabellaTestata])"

### Attenzione !

In fase di esitazione attività in Multimedia i recordset Simple# contenuti nei pannelli# sono esclusi dal processo di aggiornamento database.

## RECORDSET

Permette il salvataggio dei dati nel database

Id:

Visualizza bottone salva:

Testo bottone:

Messaggio di conferma:

SQL Query  
(usare  
[[nomecampo]]):

`SELECT idkey,TEXT1, TEXT2, TEXT3, TEXT4 FROM TBLTESTSDK where idkey=[[Grid1X.IDKey] | 0]]`

## LABEL

Visualizza un'etichetta contenente l'informazione desiderata estratta da un qualsiasi punto del CRM o di altre fonti dati

Id:

SQL Query  
(usare  
[[nomecampo]]):

`select case when count(idkey) > 0 then 'Modifica record' else 'Inserimento record' end from TblTestSDK Where idkey = [[Grid1X.IDKey] | 0]]`

Foglio di stile:

**INPUT**

Consente di visualizzare e memorizzare delle informazioni collegate ad un campo di un recordset.

Id:

Dimensioni (LxA):  X

Nome:

Descrittivo:

Formato (solo per campi datetime):

Espressione di validazione:

Valore:

BindField:

BindRecordSet:

**INPUT**

Consente di visualizzare e memorizzare delle informazioni collegate ad un campo di un recordset.

Id:

Dimensioni (LxA):  X

Nome:

Descrittivo:

Formato (solo per campi datetime):

Espressione di validazione:

Valore:

BindField:

BindRecordSet:

**INPUT**

Consente di visualizzare e memorizzare delle informazioni collegate ad un campo di un recordset.

Id:

Dimensioni (LxA):  X

Nome:

Descrittivo:

Formato (solo per campi datetime):

Espressione di validazione:

Valore:

BindField:

BindRecordSet:

**INPUT**

Consente di visualizzare e memorizzare delle informazioni collegate ad un campo di un recordset.

Id:

Dimensioni (LxA):  X

Nome:

Descrittivo:

Formato (solo per campi datetime):

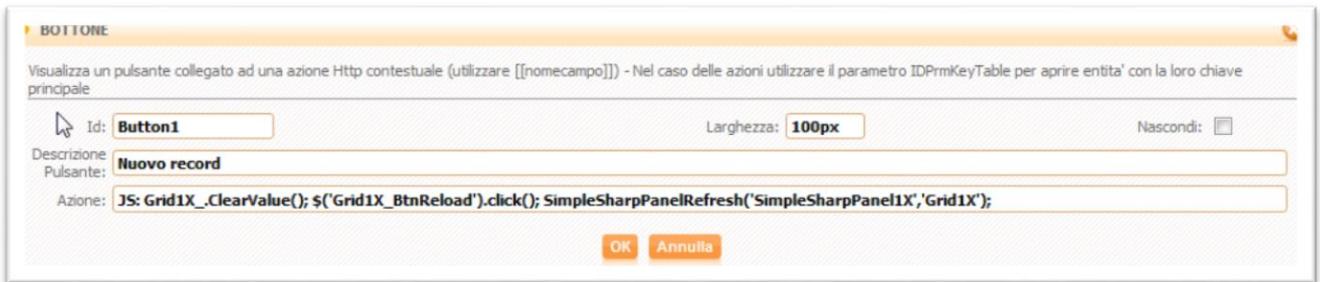
Espressione di validazione:

Valore:

BindField:

BindRecordSet:

- un bottone Simple# per creare un nuovo record



L'azione invocata sul click di questo bottone è quella indicata nel campo Azione: vengono invocati una serie di comandi via script Javascript (si noti che la griglia viene disegnata in formato HTML sul client con un nome che è la concatenazione di quello definito in sede di definizione dell'oggetto Simple# e il carattere "underscore").

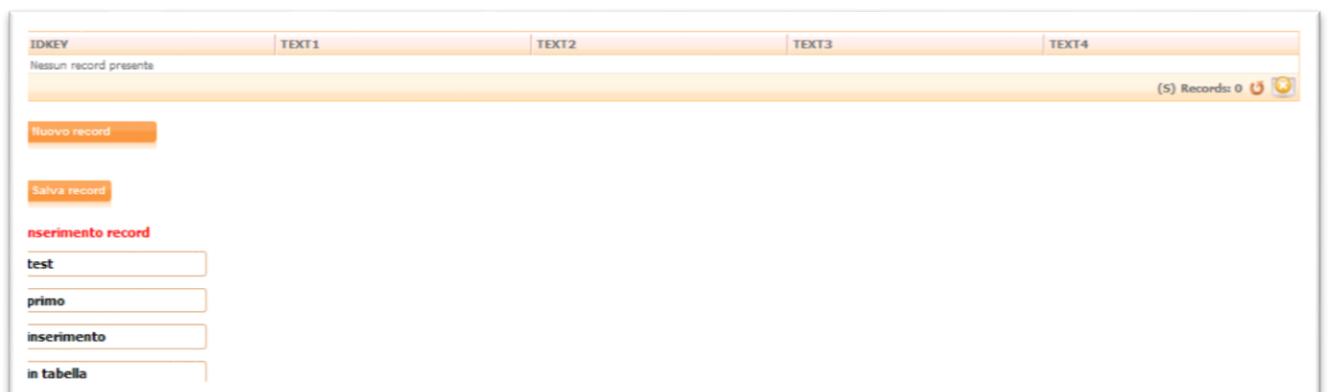
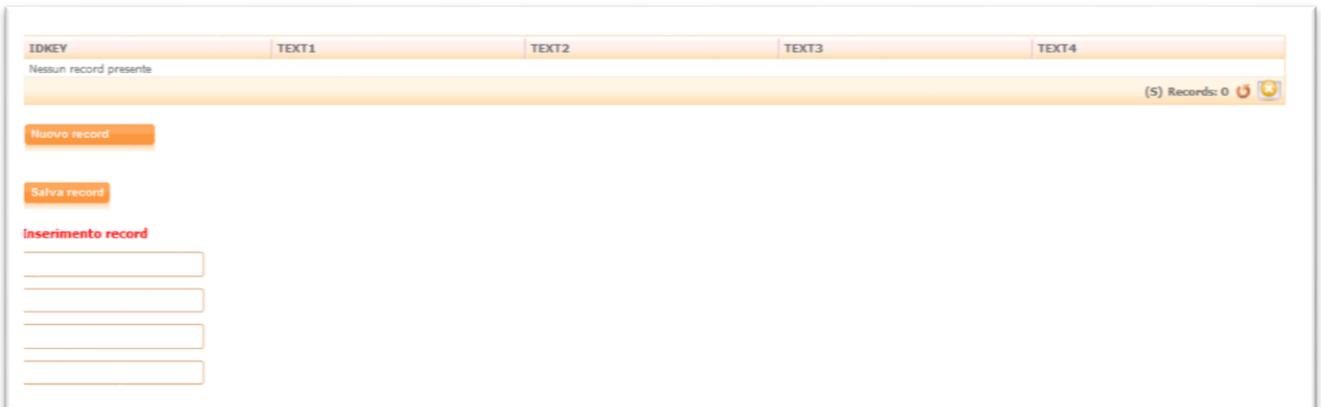
*Grid1X\_ClearValue()* → è l'invocazione dello script per deselezionare una possibile selezione di un record sulla griglia.

*\$('#Grid1X\_BtnReload').click()* → via Prototype (Framework Javascript utilizzato nel CRM) invoco lo script che effettua il requery della griglia

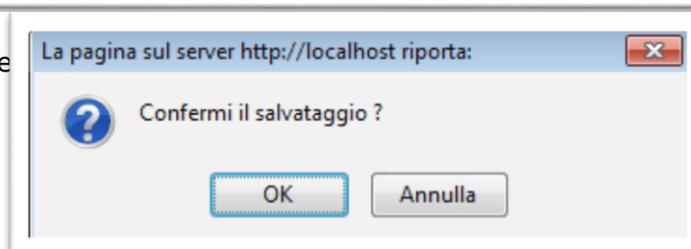
*SimpleSharpPanelRefresh('SimpleSharpPanel1X','Grid1X')* → invoco lo script di refresh del pannello Simple# (richiede come parametri il nome del pannello Simple# e il nome della griglia collegata al pannello)

Vediamo in azione lo script partendo dalla tabella custom utilizzata vuota.

Cliccando sul bottone "Nuovo record" sarà possibile inserire dei dati nei campi sottostanti alla griglia.



Cliccando sul bottone



conferma:

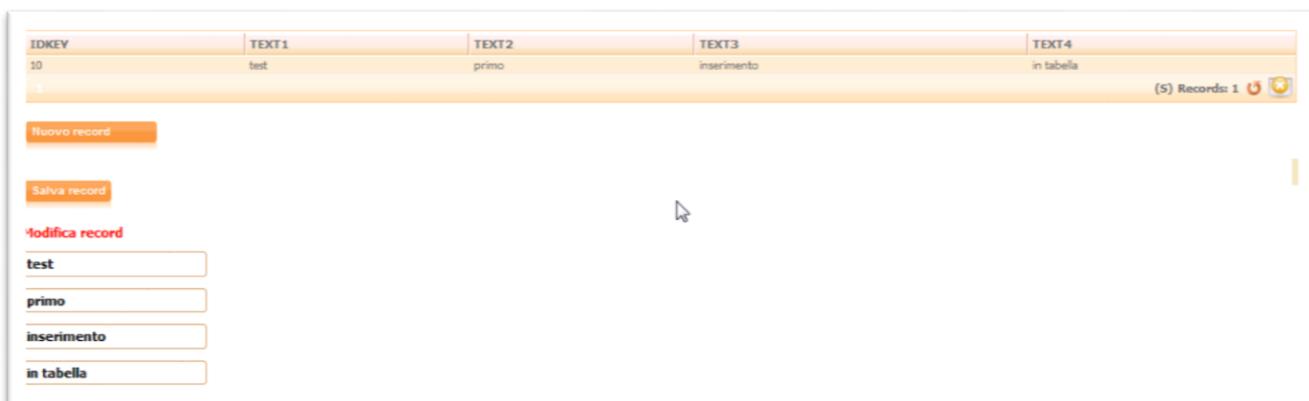
Confermando cliccando sul bottone "OK", verrà effettuato l'inserimento dei dati nella tabella utilizzata.

La versione attuale del Simple# non consente di aggiornare direttamente la griglia, infatti per poter vedere realmente il dato inserito sarà necessario cliccare sul bottone "Aggiorna elenco" della griglia.



IDKEY	TEXT1	TEXT2	TEXT3	TEXT4
10	test	primo	inserimento	in tabella

Cliccando invece sulla riga appena inserita, si caricheranno i dati della riga selezionata nei campi in basso consentendone la modifica.



IDKEY	TEXT1	TEXT2	TEXT3	TEXT4
10	test	primo	inserimento	in tabella

Nuovo record

Salva record

**Modifica record**

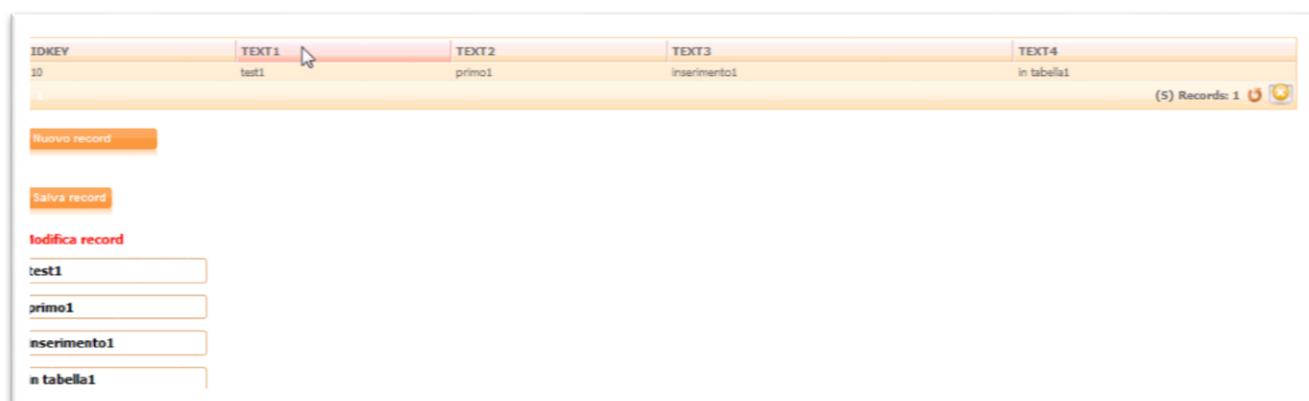
test

primo

inserimento

in tabella

Si noti come sia cambiata la label indicante l'operazione che si sta compiendo. Modificando a questo punto i dati e cliccando nuovamente sul bottone "Salva record" verranno aggiornati i dati della griglia.



IDKEY	TEXT1	TEXT2	TEXT3	TEXT4
10	test1	primo1	inserimento1	in tabella1

Nuovo record

Salva record

**Modifica record**

test1

primo1

inserimento1

in tabella1

Anche in questo caso sarà necessario aggiornare la griglia per poter verificare la reale applicazione della modifica al record della tabella.

### 2.1.4.3 Griglia con possibilità di aggiornare alcune caratteristiche utilizzate

Ipotizzando di avere una griglia con delle opportunità esistenti sulle quali sono associate 2 caratteristiche predefinite, si vuole permettere in questo esempio la modifica del valore di queste caratteristiche. Ipotizziamo l'esistenza di due caratteristiche così configurate:

The screenshot shows the 'Caratteristiche' form for ID 19. The form includes the following fields and options:

- ID: 19
- N° abbinamenti: 4
- Caratteristica: test1-opp
- Tipo Port.: [empty]
- Codice di controllo: [empty]
- Priorità: 0
- Note: [empty]
- Descrizione estesa: test1-opp
- Disabilitato:
- Stampa:
- Visibile sul palmare:
- Libero 01:
- Libero 02:
- Sola Lettura:
- Standard:
- Tipo Tabella: Opportunità
- Raggruppamento: PREVENTIVO1
- Tipo dato: Numero
- Sorgente: [empty]
- Limita valori:
- Tipo: Reale
- Tabella virtuale: [empty]
- Controlla: [button]

Data/ora inserimento	Data/ora ult.modifica	Utente	Utente ult.modifica	Numero modifiche
11/03/2011 17.32.00	11/03/2011 17.36.00	ADMIN	ADMIN	2

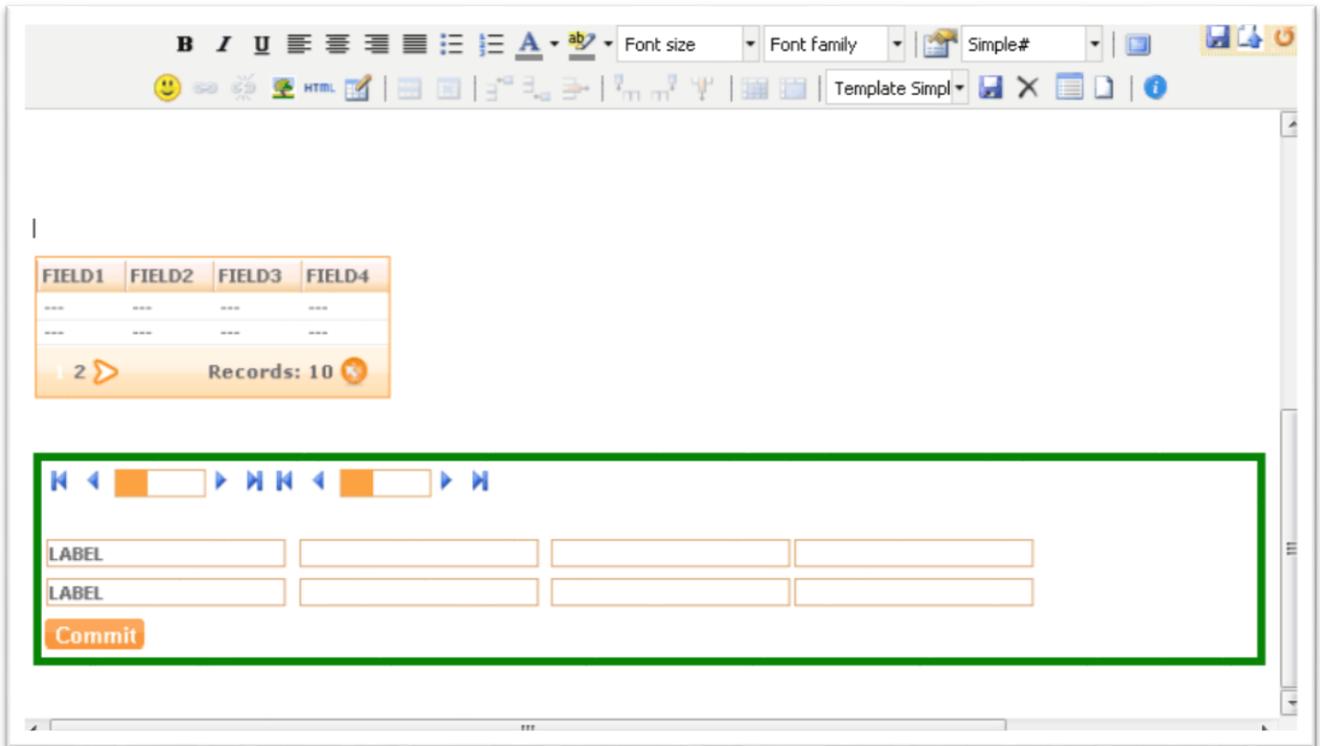
The screenshot shows the 'Caratteristiche' form for ID 20. The form includes the following fields and options:

- ID: 20
- N° abbinamenti: 4
- Caratteristica: test2-opp
- Tipo Port.: [empty]
- Codice di controllo: [empty]
- Priorità: 0
- Note: [empty]
- Descrizione estesa: test2-opp
- Disabilitato:
- Stampa:
- Visibile sul palmare:
- Libero 01:
- Libero 02:
- Sola Lettura:
- Standard:
- Tipo Tabella: Opportunità
- Raggruppamento: PREVENTIVO2
- Tipo dato: Numero
- Sorgente: [empty]
- Limita valori:
- Tipo: Reale
- Tabella virtuale: [empty]
- Controlla: [button]

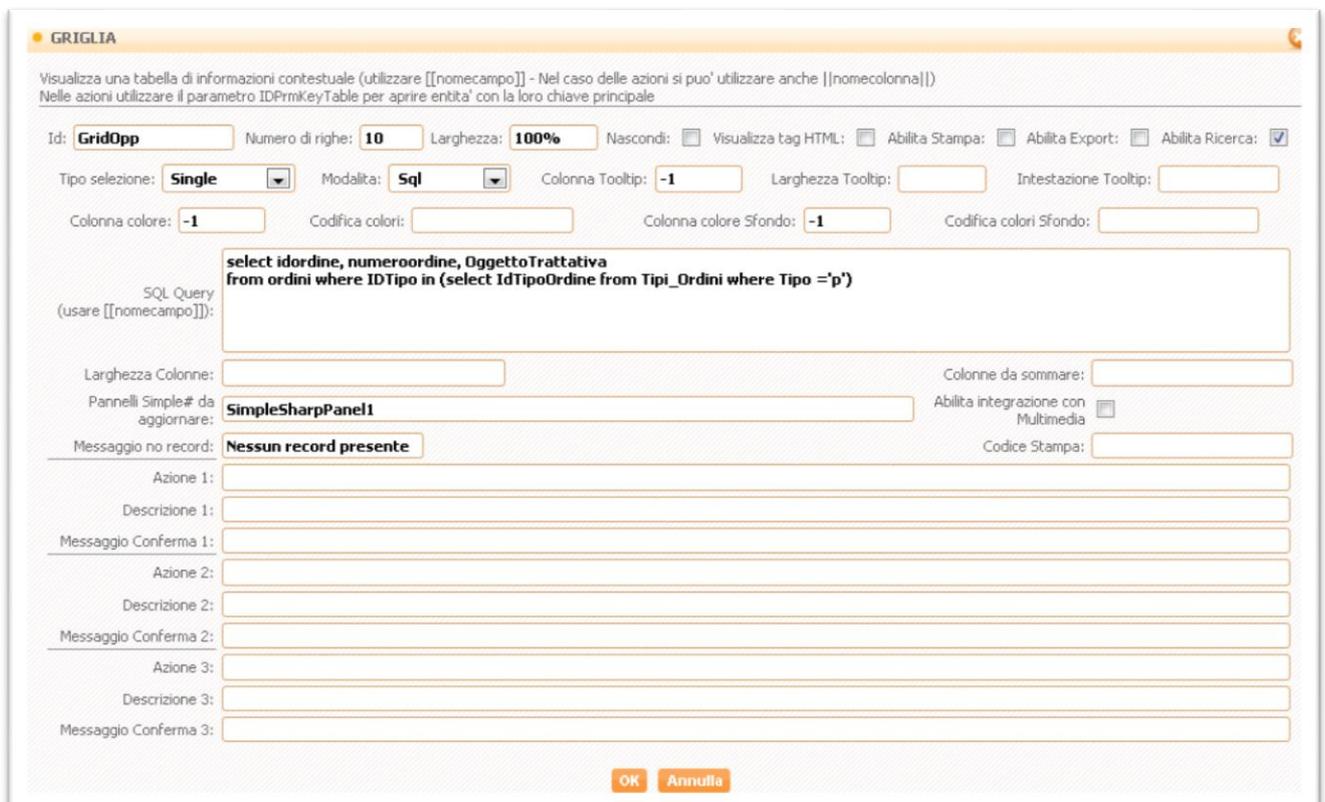
Data/ora inserimento	Data/ora ult.modifica	Utente	Utente ult.modifica	Numero modifiche
11/03/2011 17.33.00	11/03/2011 17.36.00	ADMIN	ADMIN	1

Per associarne il primo valore (e quindi l'abbinamento con l'opportunità) è possibile settarne il valore direttamente dal back office del CRM (ipotizzando inoltre che non ci siano vincoli di visibilità sulle caratteristiche per l'utenza del CRM in uso).

Ecco come appare l'esempio Simple# in oggetto:



Dove la griglia è l'elenco delle opportunità:



## La griglia aggiorna il pannello Simple#:

### PANNELLO SIMPLE#

Permette di rendere un pannello simple# collegabile agli oggetti griglie simple#

Id:

Il pannello Simple# può essere collegato alle griglie Simple# tramite la proprietà impostabile sulla griglia: "Pannelli Simple# da aggiornare:". selezionando una singola riga (impostare sulla griglia "Tipo selezione = Single") verrà aggiornato il pannello Simple# passandogli tutti i valori delle colonne in base alla riga selezionata. Il formalismo per poter accedere a questi parametri all'interno del pannello è il seguente: `[[IDGriglia.NomeColonna | |ValoreDefault]]`, dove:

- Il primo valore (prima del punto) indica l'**ID della griglia testata**
- Il secondo (dopo il punto e prima del carattere pipe) indica **Indica il nome della colonna da cui verrà preso il valore in base alla riga selezionata.**
- Il terzo valore (dopo il carattere pipe) indica il **valore di default nel caso in cui non vi siano righe selezionate nella griglia testata**; è possibile specificare un valore puntuale o una query Sql

---

Esempio realizzazione griglia testata - dettaglio (master - detail)

- Griglia testata con le seguenti proprietà:
  - ID="GridTestata"
  - Sql="SELECT ID, Descrizione FROM MiaTabellaTestata"
  - PannelliSimple#DaAggiornare="SimplePanelDettaglio"
- Pannello Simple# con ID="SimplePanelDettaglio"
- Griglia di Dettaglio dentro il Pannello Simple# con le seguenti proprietà:
  - ID="SimplePanelDettaglio"
  - Sql="SELECT ID, Descrizione FROM MiaTabellaDettaglio Where IDTestata = [[GridTestata.ID | |0]]"

oppure

- Griglia di Dettaglio dentro il Pannello Simple# con le seguenti proprietà:
  - ID="SimplePanelDettaglio"
  - Sql="SELECT ID, Descrizione FROM MiaTabellaDettaglio Where IDTestata IN([[GridTestata.ID | |SELECT ID FROM MiaTabellaTestata]])"

---

**Attenzione !**  
In fase di esitazione attività in Multimedia i recordset Simple# contenuti nei pannelli# sono esclusi dal processo di aggiornamento database.

## Nel pannello sono presenti i seguenti controlli Simple#:

- un recordset Simple# per ogni caratteristica da modificare
- un bottone per il "commit" dei recordset modificati
- label rappresentante il raggruppamento della caratteristica
- campi di input Simple# per ogni campo "not null" della tabella da aggiornare: quest'aspetto è di fondamentale importanza (infatti sono stati lasciati appositamente visibili al posto di nasconderli) in quanto il bottone di "commit" necessita di tutti i campi indispensabili per effettuare il corretto aggiornamento del recordset Simple# (in assenza di questi si otterrebbe un errore di violazione di *reference constraint* in quanto si tenterebbe di aggiornare in maniera errata la tabella delle caratteristiche)

### RECORDSET

Permette il salvataggio dei dati nel database

Id:  Visualizza bottone salva:  Testo bottone:  Messaggio di conferma:

SQL Query (usare [[nomecampo]]):  

```
select IDAbbinamento, CodCaratteristica, nota,Valore, CodOrdine from Caratteristiche_Opportunita
where CodCaratteristica=19 and CodOrdine=[[GridOpp.idordine | |0]]
```

**RECORDSET**

Permette il salvataggio dei dati nel database

Id:  Visualizza bottone salva:  Testo bottone:  Messaggio di conferma:

SQL Query (usare [[nomecampo]]):  

```
select IDAbbinamento, CodCaratteristica, nota, Valore, CodOrdine from Caratteristiche_Opportunita
where CodCaratteristica=20 and CodOrdine=[[GridOpp.idordine | 0]]
```

Si noti che anche in questo caso i recordset Simple# vengono aperti in base al valore selezionato in griglia.

**LABEL**

Visualizza un'etichetta contenente l'informazione desiderata estratta da un qualsiasi punto del CRM o di altre fonti dati

Id:

SQL Query (usare [[nomecampo]]):  

```
select Gruppo from Caratteristiche where IDCaratteristica = 19
```

Foglio di stile:

**LABEL**

Visualizza un'etichetta contenente l'informazione desiderata estratta da un qualsiasi punto del CRM o di altre fonti dati

Id:

SQL Query (usare [[nomecampo]]):  

```
select Gruppo from Caratteristiche where IDCaratteristica = 20
```

Foglio di stile:

**INPUT**

Consente di visualizzare e memorizzare delle informazioni collegate ad un campo di un recordset.

Id:

Dimensioni (LxA):  x

Nome:

Descrittivo:

Formato (solo per campi datetime):

Espressione di validazione:

Valore:

BindField:  BindRecordSet:

Rende il campo non editabile:  Campo obbligatorio:

Rende il campo editabile con multi riga:  Nascondi:

**INPUT**

Consente di visualizzare e memorizzare delle informazioni collegate ad un campo di un recordset.

Id:

Dimensioni (LxA):  X

Nome:

Descrittivo:

Formato (solo per campi datetime):

Espressione di validazione:

Valore:

BindField:

BindRecordSet:

Rende il campo non editabile:  Campo obbligatorio:

Rende il campo editabile con multi riga:  Nascondi:

**INPUT**

Consente di visualizzare e memorizzare delle informazioni collegate ad un campo di un recordset.

Id:

Dimensioni (LxA):  X

Nome:

Descrittivo:

Formato (solo per campi datetime):

Espressione di validazione:

Valore:

BindField:

BindRecordSet:

Rende il campo non editabile:  Campo obbligatorio:

Rende il campo editabile con multi riga:  Nascondi:

**INPUT**

Consente di visualizzare e memorizzare delle informazioni collegate ad un campo di un recordset.

Id:

Dimensioni (LxA):  X

Nome:

Descrittivo:

Formato (solo per campi datetime):

Espressione di validazione:

Valore:

BindField:

BindRecordSet:

Rende il campo non editabile:  Campo obbligatorio:

Rende il campo editabile con multi riga:  Nascondi:

**INPUT**

Consente di visualizzare e memorizzare delle informazioni collegate ad un campo di un recordset.

Id:

Dimensioni (LxA):  X

Nome:

Descrittivo:

Formato (solo per campi datetime):

Espressione di validazione:

Valore:

BindField:

BindRecordSet:

Rende il campo non editabile:  Campo obbligatorio:

Rende il campo editabile con multi riga:  Nascondi:

**INPUT**

Consente di visualizzare e memorizzare delle informazioni collegate ad un campo di un recordset.

Id:

Dimensioni (LxA):  x

Nome:

Descrittivo:

Formato (solo per campi datetime):

Espressione di validazione:

Valore:

BindField:  BindRecordSet:

Rende il campo non editabile:  Campo obbligatorio:

Rende il campo editabile con multi riga:  Nascondi:

Il bottone di "commit" prevede di salvare tutti i recordset Simple# presenti nell'esempio:

**COMMIT**

Visualizza un pulsante che esegue il salvataggio degli oggetti recordset specificati

Id:  Larghezza:

Nascondi:

Descrizione Pulsante:

Messaggio di conferma:

Elenco RecordSet:

Ecco l'esempio in azione:

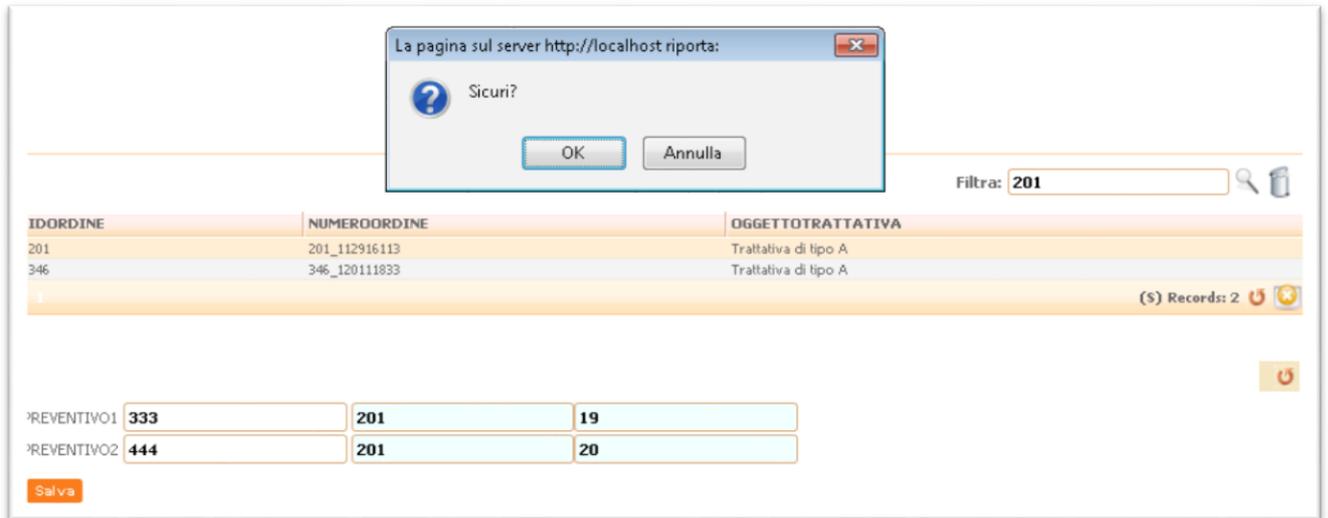
Filtra:   

IDORDINE	NUMEROORDINE	OGGETTOTRATTATIVA
201	201_112916113	Trattativa di tipo A
346	346_120111833	Trattativa di tipo A

(S) Records: 2  

PREVENTIVO1	<input type="text" value="111"/>	<input type="text" value="201"/>	<input type="text" value="19"/>
PREVENTIVO2	<input type="text" value="222"/>	<input type="text" value="201"/>	<input type="text" value="20"/>

Nell'esempio indicato si tenta di aggiornare le due caratteristiche sostituendo i valori esistenti e poi cliccando sul bottone Salva:



Cliccando sul bottone OK, verranno aggiornati i valore delle due caratteristiche.

Di seguito il codice html dell'esempio da incollare nell'editor HTML Simple#:

```
<p></p>
<p> </p>
<table id="SimpleSharpPanel1" class="SimpleSharpPanel">
<tbody>
<tr>
<td>

</tr>
<tr>

</tr>
<tr>
<td>

</tr>
<tr>
<td>
 <img
class="QBILICObjYvDRPSNkSGgwUTJ5a1QzSmthVzVsTVrBPxxkbUZZzFdvPSNXMXR1Y21sa1QzQndMbWxrYjNka2FXNwXmSHd3WFYwPXxbVz0Wkc5dWJ
Iaz0jWm1Gc2MyVT18YlHwC2RHbHhVzVs11ptRnNjMIU9fGntVnhkV2x5WldRPSNabUZZyZJVPXxkMmxrZEdnPSNNVfV3Y0hnPxxhR1ZwWjJoMCNNVfZ3ZUE

```



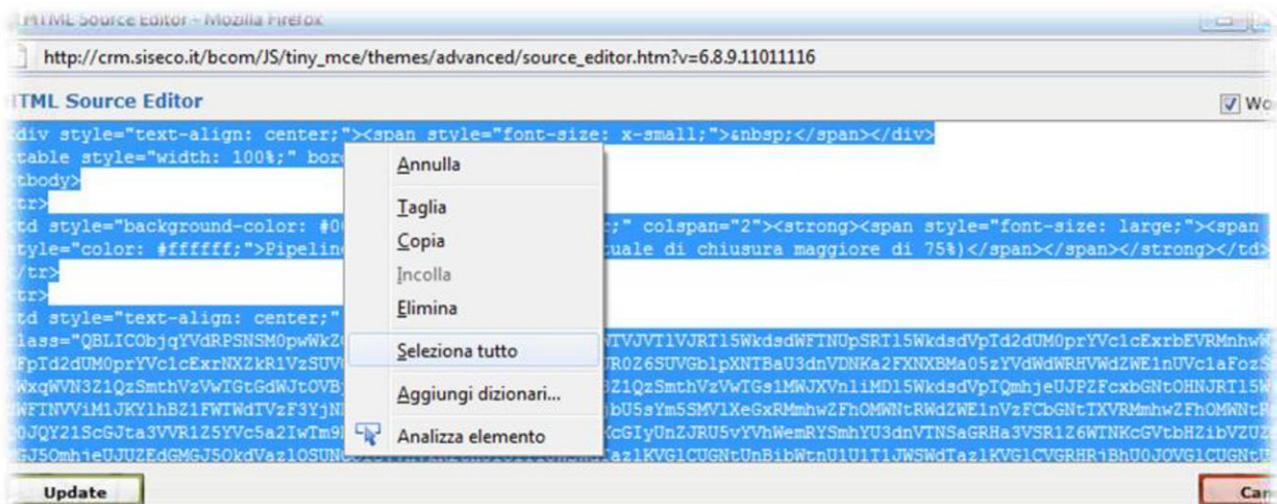
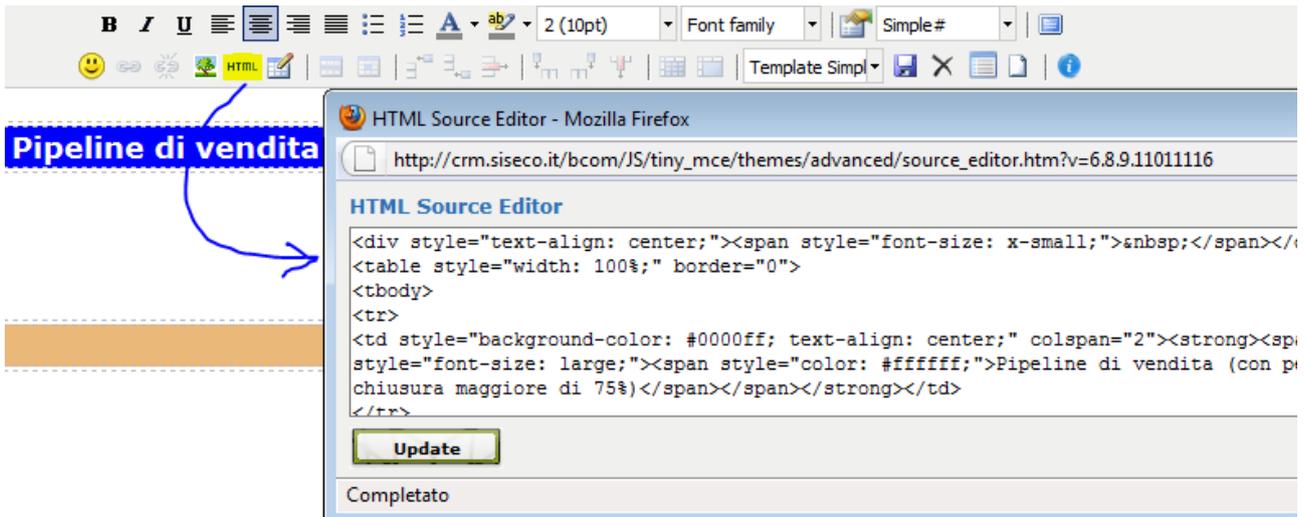
## 2.2 Rilascio in produzione di Simple#

E' possibile realizzare le personalizzazioni Simple# solamente utilizzando un'installazione in cui è attivo il modulo 15 - Macrolinguaggio Simple#.

Una volta terminate le personalizzazioni e configurazioni utilizzando questo linguaggio è possibile trasportare il codice realizzato in modo molto semplice.

Accedere all'editor Simple# e premere il tasto "HTML". Compare l'editor del sorgente HTML del pannello Simple#.

Selezionare tutto il contenuto (click destro, Seleziona tutto), quindi copiare il testo e salvarlo in un qualsiasi file di testo.



A questo punto è sufficiente andare sull'installazione priva del modulo 15 - Macrolinguaggio Simple# e procedere allo stesso modo, incollando il codice prodotto. **Sarà così molto semplice recuperare e riutilizzare il codice Simple# prodotto.**

**NB:** si consiglia di sviluppare le personalizzazioni utilizzando una copia del database di destinazione.

E' possibile, ad esempio, salvarsi anche molte versioni dell'home page del CRM, come diversi "template".

## 3. SDK

Verificare di aver disponibile la licenza 33 del CRM: anche se tutto il sistema fosse correttamente configurato, in assenza di tale modulo, le modifiche SDK verrebbero ignorate.

### 3.1 Configurazione SDK

Il livello di personalizzazione più spinto è costituito dall' SDK, il quale consente una profonda personalizzazione del CRM permettendo di introdurre nuove logiche di business, modificando il comportamento di componenti esistenti ed aggiungendo nuovi elementi applicativi nel rispetto delle linee guida generali. Riprendendo quanto detto nei requisiti, è necessario avere l'intero ambiente di personalizzazione su Visual Studio .Net e compilare il codice sorgente realizzato al fine di produrre una libreria DLL che verrà poi rilasciata sul sistema test e poi successivamente in produzione.

Esistono due tipologie di personalizzazioni attuabili attraverso l'SDK:

- Plugin Trigger: estendendo il comportamento delle griglie/form esistenti implementando degli appositi eventi
- Plugin Step: realizzando degli step personalizzati da aggiungere al workflow

La configurazione prevede l'inserimento di alcuni record nel database del CRM.

Le tabelle in questione sono:

- THIRDPARTS\_ADDINS: rappresenta il contenitore delle DLL esterne di tutti i plugin presenti nel CRM
- THIRDPARTS\_EVENTS: rappresenta il contenitore di tutti gli eventi che possono essere estesi attraverso il Plugin Trigger
- THIRDPARTS\_PAGES: rappresenta il contenitore di tutte le entità (form/grid) che possono essere estese con gli eventi Plugin Trigger

La prima tabella viene normalmente alimentata attraverso degli script SQL dedicati (si veda più in avanti negli appositi esempi dedicati), mentre le altre due tabelle vengono alimentate attraverso due modalità: una procedura manuale (impresoft fornisce lo script sql per riempire le due tabelle) oppure una procedura automatica attivabile dalla pagina dei log tecnici.



Loggandosi come amministratore e aprendo la pagina dei log tecnici, è stata aggiunta una sezione che prevede di abilitare la modalità DEBUG MODE: tale modalità se abilitata, prevede che navigando sulle pagine oppure sulle griglie del CRM, vengano inseriti tutte le pagine e tutti gli eventi estendibili attraverso la modalità Plugin Trigger. Questa modalità è naturalmente da considerarsi solo a livello preparatorio in quanto rallenta le performance del sistema e andrebbe quindi fatta solo se necessaria ovvero per preparare la configurazione dei Plugin Trigger: lo sviluppatore dovrà quindi dopo aver abilitata la modalità DEBUG MODE, navigare in tutte le pagine e griglie che desidera estendere, controllare che le tabelle THIRDPARTS\_EVENTS e THIRDPARTS\_PAGES sono state modificate dopodiché ritornare nella pagina dei log tecnici e disabilitare la modalità DEBUG MODE cliccando sull'apposito bottone:

Abilita/disabilita Debug Mode

Disabilita modalità DEBUG

La modalità DEBUG MODE agisce solo a livello di SESSION pertanto si limita la possibilità di dimenticarsi di disabilitarla.

Essendo tale *feature* un aspetto di recente introduzione (disponibile dalla versione 7.0.4), se non fosse presente nella versione del CRM disponibile, sarà possibile applicare uno script SQL per l'inserimento dei record nelle tabelle THIRDPARTS\_EVENTS e THIRDPARTS\_PAGES.

## 3.2 Esempi applicativi di SDK

La cartella applicativa che viene distribuita contiene la soluzione del progetto Visual Studio .Net e gli script SQL per registrare correttamente i Plugin.

### 3.2.1 Plugin Trigger

Prevede di creare una classe che implementi l'interfaccia Siseco.IThirdPartsAddIns e in essa creare un metodo chiamato Do con la firma indicata nel prototipo indicato:

```
class ClassName : Siseco.IThirdPartsAddIns
{
 public bool Do(Siseco.WebPage Page, object Args, Siseco.CBindableRecordset rS)
 {
 ...
 return true;
 }
}
```

La classe dovrà essere registrata nell'apposita tabella di configurazione chiamata THIRDPARTS\_ADDINS con uno script SQL adeguato (si veda la specifica della tabella nella documentazione CHM distribuita).

Vengono distribuiti 2 Trigger Plugin per estendere gli eventi della griglia dei Comuni e della form dei Listini attraverso i seguenti script SQL:

```
INSERT INTO THIRDPARTS_ADDINS
([NOTE],[OBJECTNAME],[EVENTNAME],[CLASSNAME],[EXECUTIONORDER],[DLLPATH],[DLL],[VERSION],[IDUTENTE],[DISABLED])
VALUES
('Test SDK','ListinoForm','RsListino_AfterRequery','TestSDK.RsListino_AfterRequery',1,'C:\\CRM\\Addins\\Test.dll',';',1.0.0.0,';',0)
```

```
INSERT INTO THIRDPARTS_ADDINS
([NOTE],[OBJECTNAME],[EVENTNAME],[CLASSNAME],[EXECUTIONORDER],[DLLPATH],[DLL],[VERSION],[IDUTENTE],[DISABLED])
VALUES
('Test SDK','ComuniGrid','Page_OnAfterPreRender','TestSDK.Page_OnAfterPreRender',1,'C:\\CRM\\Addins\\Test.dll',';',1.0.0.0,';',0)
```

I seguenti script prevedono che la DLL Plugin Trigger sia localizzata nel path C:\CRM\Addins e abbia nome Test.dll (adeguare opportunamente lo script in caso di modifica del percorso o del nome della DLL).

Se fosse necessario, estendere i diritti di "Full Control" all'utenza "Everyone" sulla cartella utilizzata per contenere i plugin.

Il Plugin Trigger della griglia dei comuni prevede di modificare l'origine SQL della griglia evitando di estrarre le ultime tre colonne (latitudine, longitudine e sede tribunale): tale obiettivo viene raggiunto implementando l'evento *Page\_OnAfterPreRender* sull'entità

ComuniGrid presenti rispettivamente nelle tabelle THIRDPARTS\_EVENTS e THIRDPARTS\_PAGES.

Il Plugin Trigger della form dei listini ha lo scopo di tracciare nella tabella custom (utilizzata negli esempi del Simple#) alcune informazioni del listino visualizzato nella form, implementando l'evento *RsListino\_AfterRequery* sull'entità ListinoForm presenti rispettivamente nelle tabelle THIRDPARTS\_EVENTS e THIRDPARTS\_PAGES.

### 3.2.2 Plugin Step personalizzato

Prevede di creare uno step personalizzato da utilizzare direttamente in Multimedia.

Oltre alle tabelle di configurazione citate nel paragrafo precedente, esistono altre tabelle che devono essere compilate correttamente per la configurazione del Plugin Step personalizzato: WORKFLOWPAGETYPES, WORKFLOWSTEPS e WORKFLOWSTEPTYPES contenenti informazioni relative anche agli altri step dei workflow presenti nel database del CRM utilizzati in Multimedia e vengono normalmente valorizzate quando si agisce sulla pagina di Gestione Campagne per la personalizzazione del workflow.

Un macro scheletro del Plugin Step personalizzato è il seguente:

```
namespace TestSDK
{
 public class Gestione : Siseco.MultimediaSteps.CMultimediaExternalGuestPagePlus
 {
 #region Costruttore
 public Gestione(WebPage Page, string PageGuid) : base(Page, PageGuid)
 {
 ...
 }
 #endregion

 #region Proprietà e definizioni
 //elenco di tutti i controlli utilizzati normalmente creati come proprietà e delle tabelle Webcontrols table
 //utilizzate per contenere i controlli e per disegnare la pagina (non esiste designer)
 #endregion

 #region Metodi
 //elenco dei metodi utilizzati per la gestione dello step

 //Metodo per disegnare tabelle e renderizzare controlli
 private void OnPreRender()
 {
 ...
 }

 //Inizializza i controlli
 private void OnInitControls()
 {
 ...
 }

 //Contiene gli script javascript utilizzati per i controlli
 private string GetClientScript()
 {
 ...
 }

 //Metodo centrale per disegnare lo step
 public override string GetHtml(string Parameters)
 {
 base.GetHtml(Parameters);

 ...
 }
 #endregion
 }
}
```

```

//classe contenete i metodi Ajax utilizzati nello step
public class AjaxGestione : GatAjaxMultimediaCommonStepData
{
 Siseco.WebPage m_Page = null;

 //costruttore
 public AjaxGestione(Siseco.WebPage Page)
 {
 m_Page = Page;
 }
 ...
}

//Classe per definire azioni da fare al momento dell'esitazione
public class GestioneUpdateToEsito : Siseco.Multimedia.IMultimediaWorkFlowSaveData
{
 public GestioneUpdateToEsito()
 : base()
 {
 }
 public CStepReturnValue Commit(CMultimediaData CurrentTab, string PageGuid)
 {
 CStepReturnValue Rval = new CStepReturnValue();
 Rval.ShowResult = false;

 //TODO commit...

 return Rval;
 }
 public CStepReturnValue CommitAfter(CMultimediaData CurrentTab, string PageGuid)
 {
 //TODO commit after...
 return new CStepReturnValue();
 }
}

//Classe per aggiungere lo step personalizzato tra quelli disponibili in sede di definizione del workflow di una
//campagna: viene invocato quando dalla form Gestione Campagne si apre la combo degli step disponibili
public class PluginLoader : Siseco.MultimediaSteps.CMultimediaAddinStep
{
 public PluginLoader() { }

 public override void GetStepData(ref string StepCode, ref string StepDescription, ref
List<CMultimediaPage> Pages)
 {
 //configura alcune proprietà che verranno inserite nella tabella WorkFlowPageTypes:
 // - ClassName
 // - PageCode
 // - UpdateClass
 // - EditorClass
 // - PageDesc
 // - Required

 //configura alcune proprietà che verranno salvate nella tabella WorkFlowStepTypes:
 // - StepCode
 // - StepDescription
 }
}

//classe per la definizione dei parametri personalizzabili per lo step nella form Gestione Campagne
public class GestioneEditorParameters : CMultimediaExternalGuestEditorPage
{
 #region Costruttori
 public GestioneEditorParameters(Siseco.WebPage Page)
 : base(Page) { }

 public GestioneEditorParameters(Siseco.WebPage Page, string PageGuid)
 : base(Page, PageGuid) { }
 #endregion

 #region Proprietà e definizioni

```

```

//elenco di tutti i controlli utilizzati normalmente creati come proprietà e delle tabelle Webcontrols table
utilizzate per contenere i controlli e per disegnare la pagina (non esiste designer)
#endregion

#region Event
protected override void CreateChildControls()
{
 base.CreateChildControls();
 ...
}
#endregion
}
}

```

Il Plugin dovrà essere registrato nell'apposita tabella di configurazione THIRDPARTS\_ADDINS con uno script SQL adeguato (si veda la specifica della tabella nella documentazione CHM distribuita):

```

INSERT INTO THIRDPARTS_ADDINS
([NOTE],[OBJECTNAME],[EVENTNAME],[CLASSNAME],[EXECUTIONORDER],[DLLPATH],[DLL],[VERSION],[IDUTENTE],[DISABLED])
VALUES
('Test SDK','MotiviForm','LOADSTEPADDINS','TestSDK.PluginLoader',1,'C:\CRM\Addins\Test.dll','1.0.0.0',0)

```

Lo script prevede che la DLL Plugin Step sia localizzata nel path C:\CRM\Addins e abbia nome Test.dll (adeguare opportunamente lo script in caso di modifica del percorso o del nome della DLL).

Se fosse necessario, estendere i diritti di "Full Control" all'utenza "Everyone" sulla cartella utilizzata per contenere i plugin.

Per poter utilizzare correttamente l'esempio, a puro scopo didattico, è stata utilizzata anche una tabella di appoggio creabile con il seguente script:

```

CREATE TABLE [dbo].[TblDatiCombo](
 [IDKey] [int] IDENTITY(1,1) NOT NULL,
 [Descrizione] [nvarchar](max) NULL
 CONSTRAINT [PK_TblDatiCombo] PRIMARY KEY CLUSTERED
 (
 [IDKey] ASC
)
) ON [PRIMARY]
GO

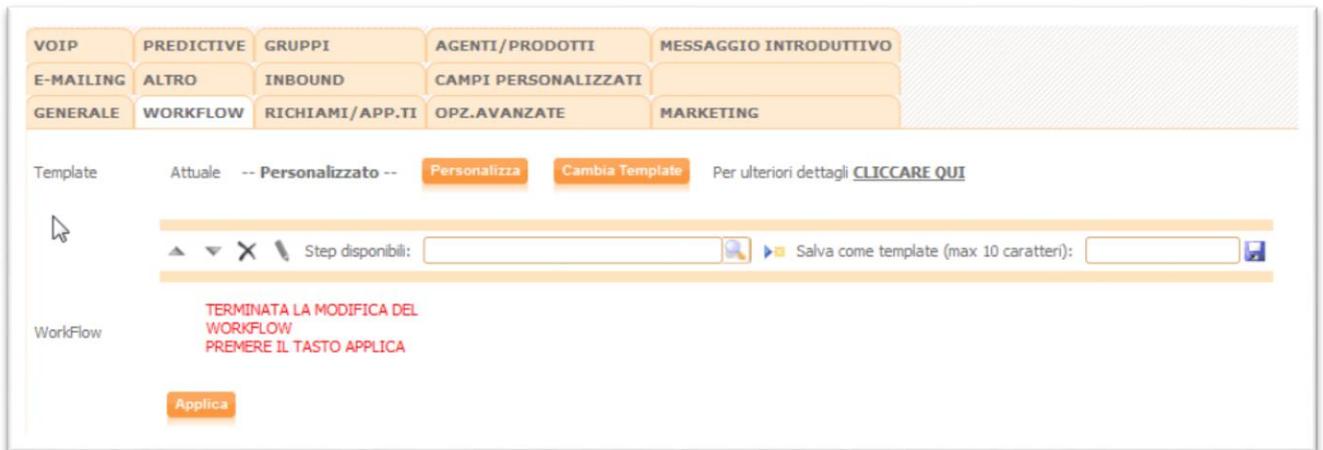
insert into TblDatiCombo values ('Descrizione1');
insert into TblDatiCombo values ('Descrizione2');

```

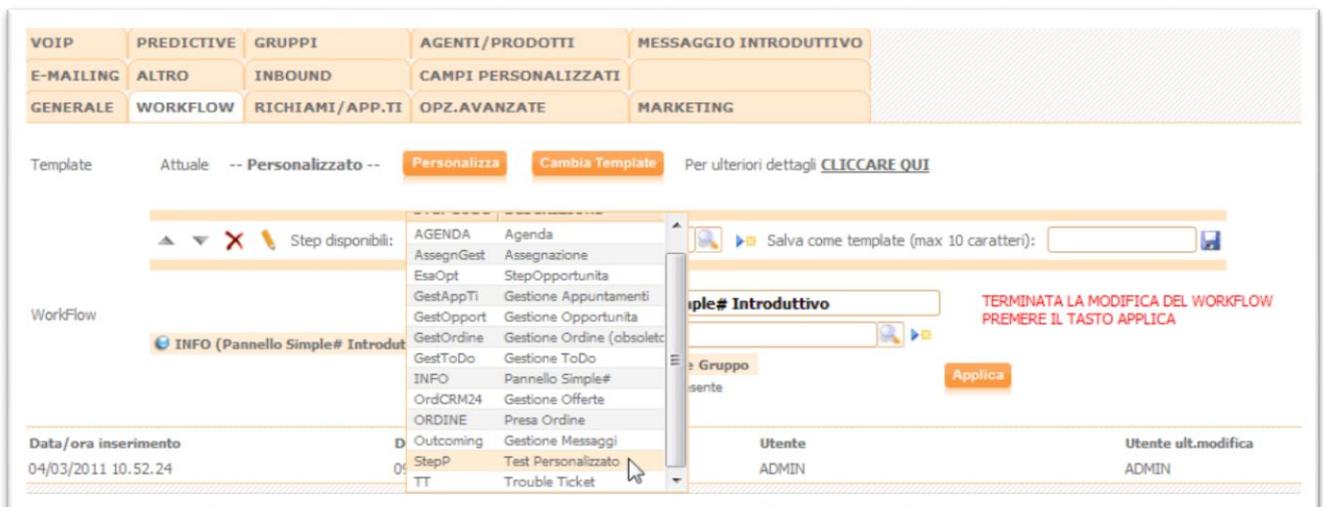
Vediamo come utilizzare il Plugin Step personalizzato.

Accedere al CRM e andare sulla Gestione Campagne, creare o selezionare la campagna nella quale si desidera modificare il workflow aggiungendo lo step personalizzato, aprire la campagna e posizionarsi sul pannello WORKFLOW.

Dopo aver cliccato sul bottone "Personalizza" si ottiene una situazione di questo tipo:



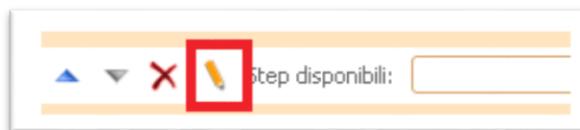
Normalmente si procede aggiungendo uno step Info in prima posizione selezionandolo dalla combo degli Step disponibili dopodiché si aggiungerà lo step personalizzato.



Si ricorda che è necessario cliccare sul bottone "Applica" per salvare le modifiche al workflow.  
Si arriverà dunque a una situazione di questo tipo:



Cliccando sul bottone di modifica delle impostazioni dello step personalizzato



sarà possibile modificare una proprietà specifica dell'esempio:

VOIP	PREDICTIVE	GRUPPI	AGENTI/PRODOTTI	MESSAGGIO INTRODUTTIVO
E-MAILING	ALTRO	INBOUND	CAMPI PERSONALIZZATI	
GENERALE	WORKFLOW	RICHIAMI/APP.TI	OPZ.AVANZATE	MARKETING

Template Attuale -- Personalizzato -- [Personalizza](#) [Cambia Template](#) Per ulteriori dettagli [CLICCARE QUI](#)

Step disponibili:  Salva come template (max 10 caratteri):

Workflow  
Descrizione:  **TERMINATA LA MODIFICA DEL WORKFLOW  
PREMERE IL TASTO APPLICA**  
Stati disponibili:   
Stati selezionati: **ID Descrizione Gruppo**  
Nessun dato presente [Applica](#)

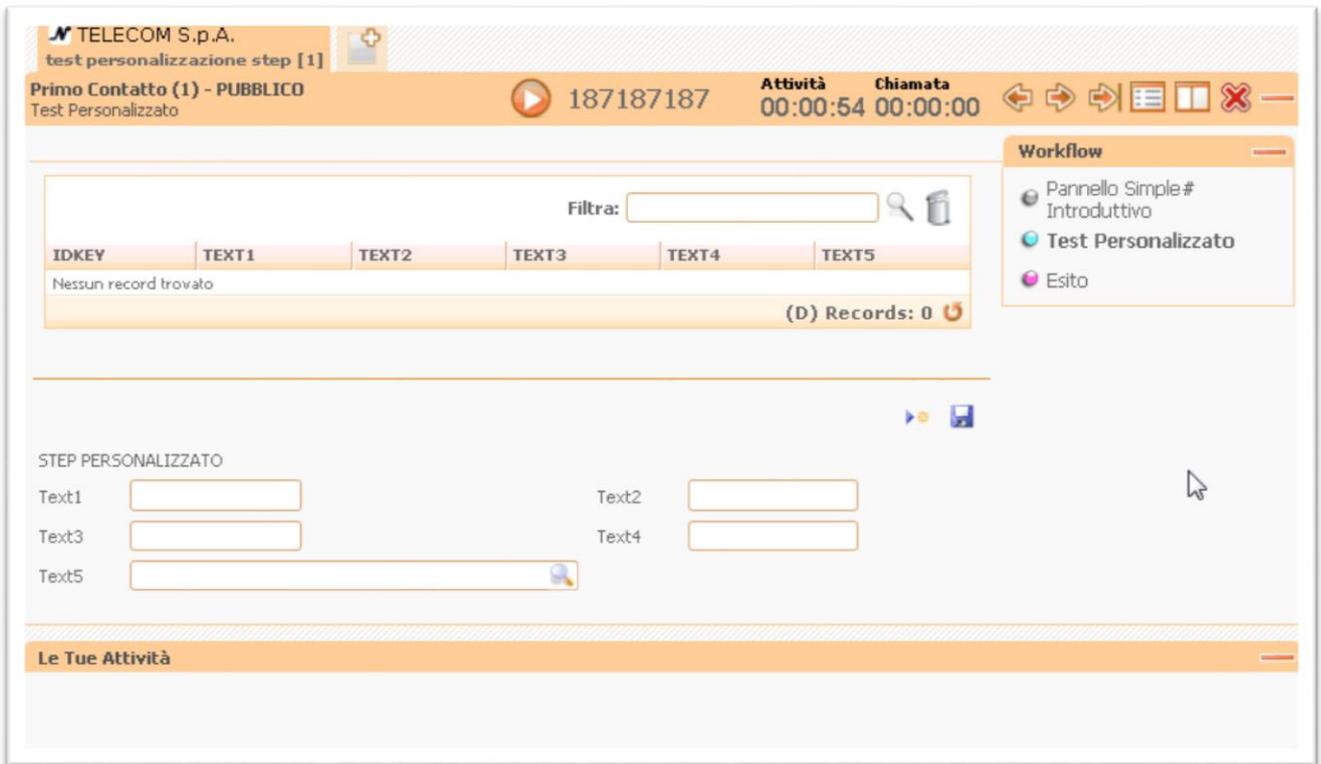
Extra data  
Visibilità della colonna descrizione:

Salvare la campagna con tutte le informazioni necessarie e proseguire.

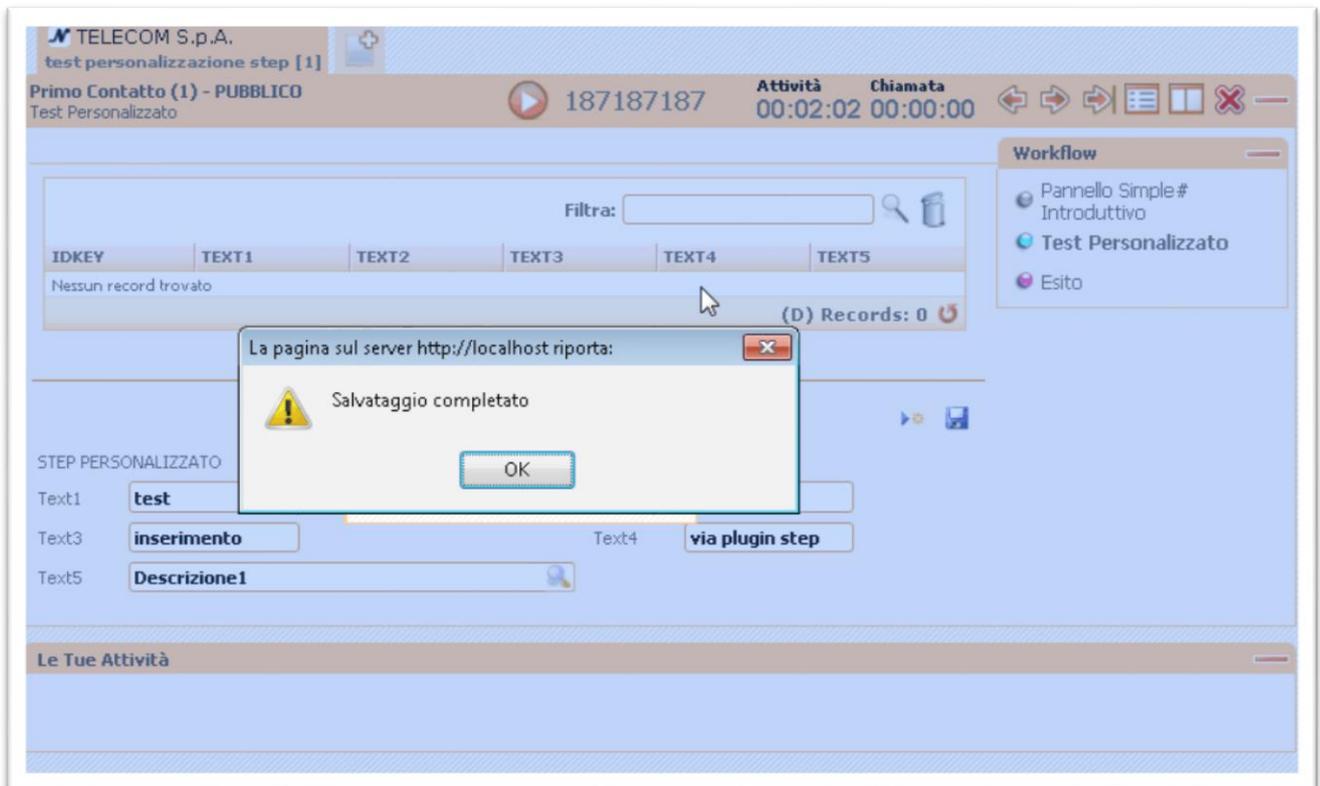
Andando in multimedia, aprendo la campagna che contiene il workflow con lo step personalizzato, si otterrà una situazione simile alla seguente:

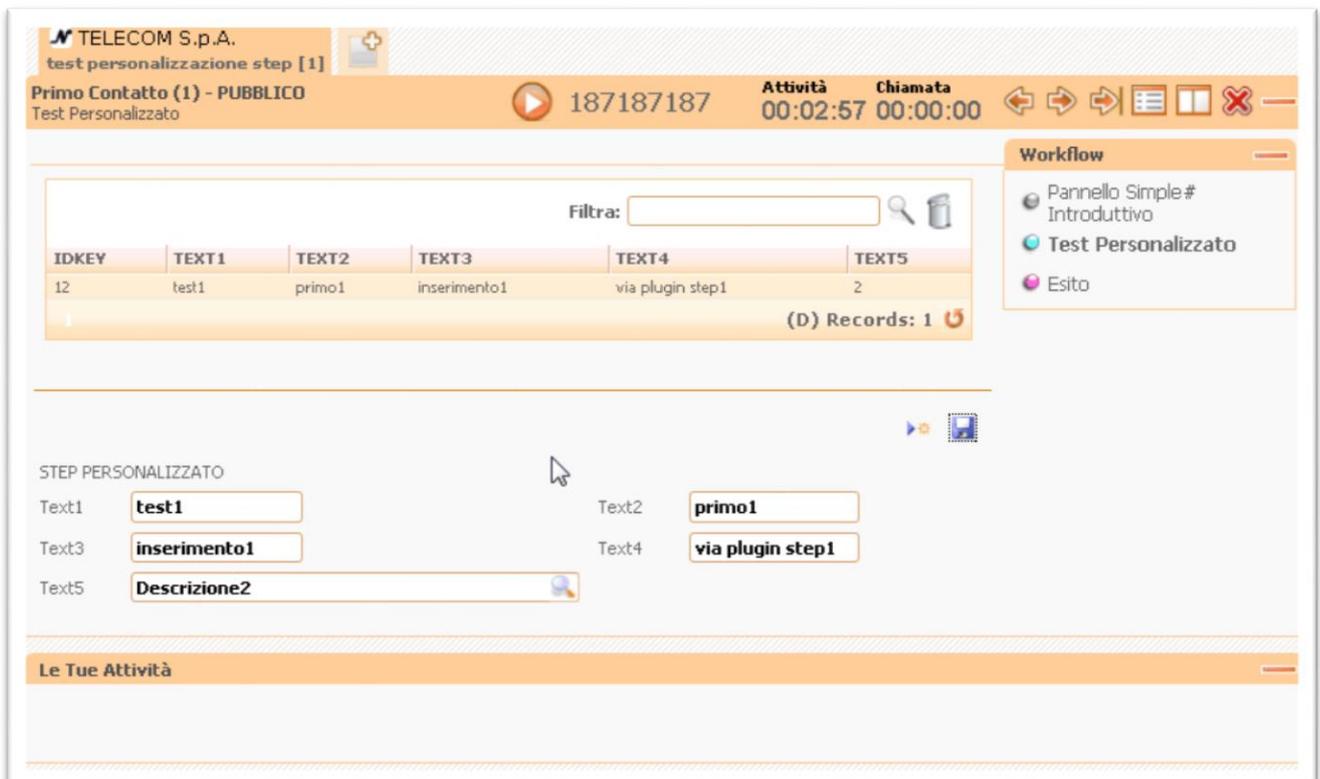
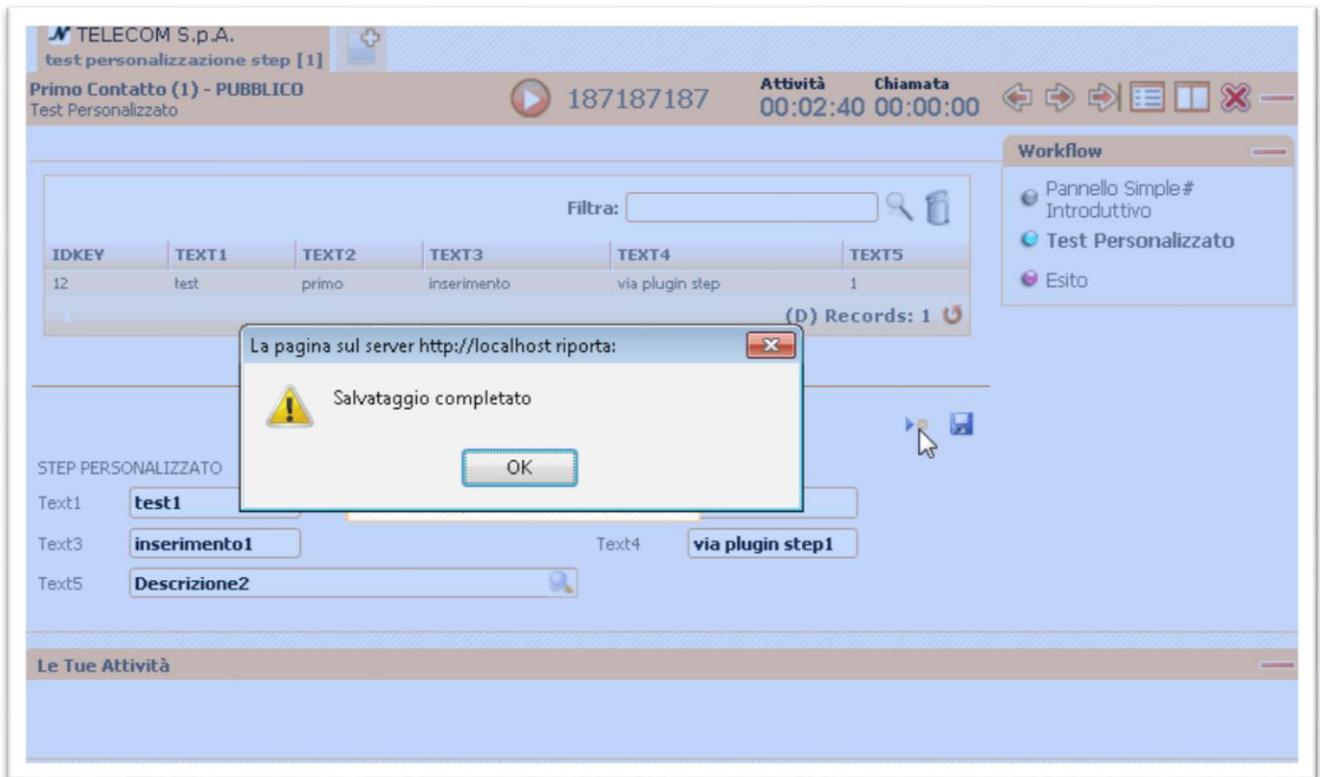


Cliccando sullo step "Test Personalizzato" nel workflow, verrà visualizzato lo step: questo esempio è stato appositamente costruito in maniera del tutto simile a quello fornito nel capitolo del Simple# per mostrare analogie e potenzialità dell'SDK rispetto al macrolinguaggio Simple#.



Sarà possibile creare nuovi record per la tabella di appoggio oppure modificare record esistenti.





Come definito in precedenza, la label "STEP PERSONALIZZATO" viene visualizzata se nella form Gestione Campagna è stato personalizzato lo step del workflow impostando la visualizzazione della label (al valore SI).

Mettendo a confronto questo esempio sviluppato in modalità SDK con quello proposto con il Simple# si possono dedurre delle importanti differenze riguardanti le due tecnologie:

- In seguito ad una modifica oppure inserimento di un record, l'aggiornamento della griglia, in modalità Simple#, deve essere fatto manualmente cliccando sull'icona di requery, mentre nella modalità SDK, viene effettuato in automatico in quanto è stato inserito lo script di requery della griglia direttamente nel metodo Ajax di aggiornamento e di inserimento dati
- Nella modalità Simple# non è possibile implementare una cancellazione fisica del record di una griglia bensì solo una cancellazione logica (si pensi di aggiungere un flag alla tabella per mantenere lo stato di obsolescenza e un campo checkbox Simple# per salvarne il valore), mentre nella modalità SDK è facilmente implementabile un nuovo metodo Ajax registrabile nella classe *AjaxGestione* dell'esempio che venga invocato al click di un nuovo bottone da accostare a quelli esistenti (nuovo e salvataggio).

### 3.2.3 Plugin Step "Gestione Offerte" personalizzato

Per personalizzare lo step delle offerte è necessario avere a disposizione i sorgenti della DLL dello step che andrà ad integrarsi in CRM Multimedia.

Dopo aver compilato la DLL è necessario:

1. Copiare la DLL in una Directory specifica (Es. "C:\CRM\Addins\StepOfferta.dll");
2. Eseguire il seguente script sul database del CRM sostituendo il percorso corretto della DLL:

```
INSERT INTO THIRDPARTS_ADDINS (
 NOTE,
 OBJECTNAME,
 EVENTNAME,
 CLASSNAME,
 EXECUTIONORDER,
 DLLPATH,
 DLL,
 VERSION,
 IDUTENTE,
 DISABLED
) VALUES (
 '',
 'MotiviForm',
 'LOADSTEPADDINS',
 'StepOfferta.StepOrdineLoader',
 1,
 'C:\CRM\Addins\StepOfferta.dll',
 '',
 '1.0.0',
 '',
 0
)
```

3. A questo punto, nella pagina di Gestione Campagne, lo step sarà compreso nell'elenco degli step disponibili in corrispondenza della voce **StpOff (StepOfferta)**:

The screenshot displays the 'Gestione Campagne' interface. At the top, there is a navigation menu with tabs: 'GENERALE', 'WORKFLOW' (selected), 'RICHIAMI/APP.TI', 'OPZ.AVANZATE', 'DC'. Below this, there are sub-tabs: 'MARKETING', 'VOIP', 'PREDICTIVE', 'GRUPPI', 'AC'; and 'MESSAGGIO INTRODUTTIVO', 'E-MAILING', 'ALTRO', 'INBOUND', 'CA'. Below the tabs, there are buttons: 'Template', 'Attuale', '-- Personalizzato --', 'Personalizza', and 'Cambia Template'. The main content area shows a 'Workflow' section with a tree view of categories: 'info (Dati Lead)', 'Outcoming (Gestione Messaggi)' (with sub-items 'Email' and 'Sms'), and 'StpOff (Step Offerta)' (with sub-items 'Offerte', 'Sezioni', 'Articoli', 'Condizione Offerta', and 'Riepilogo Offerta'). To the right, a dropdown menu titled 'Step disponibili:' is open, showing a list of steps. The 'StpOff StepOfferta' step is highlighted in yellow. The list includes: 'GestOrdine Gestione Ordine (obsolet)', 'GestToDo Gestione ToDo', 'INFO Pannello Simple#', 'OrdCRM24 Gestione Offerte', 'ORDINE Presa Ordine', 'Outcoming Gestione Messaggi', 'StpOff StepOfferta', and 'TT Trouble Ticket'.

Il codice dello step dell'offerta (file **CStepOrdine.cs**) si presenta in Visual Studio con questo elenco di classi:

```
namespace StepOfferta
{
 /// <summary> ...
 public class StepOrdineLoader : Siseco.MultimediaSteps.CMultimediaAddinStep...

 /// <summary> ...
 public class StepOrdineUpdateToEsito : Siseco.Multimedia.IMultimediaWorkflowSaveData...

 /// <summary> ...
 public class StepOrdineHeader : GatControl...

 /// <summary> ...
 public class StepSezioniHeader : GatControl...

 /// <summary> ...
 public class StepOrdineInputMask : GatAjaxMultimediaCommonStepData...

 /// <summary> ...
 public class StepAjaxMethods : GatAjaxMultimediaCommonStepData...

 /// <summary> ...
 public class StepOrdine : Siseco.MultimediaSteps.CMultimediaExternalGuestPagePlus...

 /// <summary> ...
 public class StepSezioni : Siseco.MultimediaSteps.CMultimediaExternalGuestPagePlus...

 /// <summary> ...
 public class StepProdotti : Siseco.MultimediaSteps.CMultimediaExternalGuestPagePlus...

 /// <summary> ...
 public class StepCondizioniOrdine : Siseco.MultimediaSteps.CMultimediaExternalGuestPagePlus...

 /// <summary> ...
 public class StepRiepilogoOrdine : Siseco.MultimediaSteps.CMultimediaExternalGuestPagePlus
 {
 Costruttori

 Definizioni variabili

 #region Metodi Principali

 public override string AjaxRender(Siseco.SiJson.SiJsonObject Parameter)...

 public void OnPreRender()...

 public void OnInitControls()...

 public void OnSetControls()...
 }
}
```

che corrispondono ai sottostep in Multimedia:

- **Step Offerta**
- Offerte
- Sezioni
- **Articoli**
- Condizione Offerta
- Riepilogo Offerta

Di seguito verrà disposta una macro illustrazione di ogni step con l'esatta corrispondenza delle classi elencate sopra:

**Sottostep "Offerte":**

**Sottostep "Sezioni":**

**Sottostep "Articoli":**

Offerta n° 7876\_113823189 del 26/08/2014
Class StepOrdineHeader

Imponibile: 0,00 - Totale offerta: 0,00

PRODOTTI DISPONIBILI CARRELLO

Lista dei prodotti disponibili a listino

Gruppo(1) Gruppo(2) Gruppo(3) Gruppo(4) Filtri avanzati Azzera Filtri Cerca!

CODICE	DESCRIZIONE	DESCRIZIONE_VARIANTE	PREZZOGRETT	CODICEMACRO	CODICEMACRO2	CODICEMACRO3	CODICEMACRO4
001	Infu		0,0000				

(D) Record: 1

---

Dettaglio prodotto Aggiungi al carrello Inserisci da file...

Class StepOrdineInputMask

Prezzo (EUR - 5 decimal)  IVA ind.  Unità di misura  Quantità (2 decimal)  Costo unitario

Imponibile (EUR)  Cod. IVA  Totale Iivato (EUR)

Maggior. %  Sconti %

N. riga  Stato  Data evasione

Tipo riga  Articolo

Variante

B / I / U / ↶ / ↷ / ☰ / ☷ / 📄

Demco caratteristiche M...

Class StepProdotti

- Dati Lead
- Gestione Messaggi
- Email
- Sms
- Step Offerta
- Offerte
- Sezioni
- Articoli
- Condizione Offerta
- Riepilogo Offerta
- Esito

**Sottostep "Condizioni Offerta":**

Offerta n° 7876\_113823189 del 26/08/2014
Class StepOrdineHeader

Imponibile: 0,00 - Totale offerta: 0,00

**DATI ACQUIRENTE**

Cliente **6360** **PELLICORI & C. S.P.A.** Telefono **051/6512611-799155**

Ala c. a. d.  Cellulare **051/6512611-799155**

Indirizzo **VIA EMBILIA, 3147** Fax **051/6512519-596**

Città **40054** **OZZANO DELL'EMILIA**  **BO** eMail **roberta.carta@pelliconi.com**

Cons. Inf. **<selezione>** Documento  Part. **CONTATTI WEB**

**CONDIZIONI**

Modalità consegna  Vettore  Porto

Consegna  Installazione  Lingua **Italiano**

Nr. Coll **1** Aspetto beni

Note Esterne

B / I / U / ↶ / ↷ / ☰ / ☷ / 📄

**DESTINATARIO / SEDE OPERATIVA**

Anagrafica  Telefono

Ala c. a. d.  Num.

Indirizzo  Piv.

Cap  Città

**SPESI/RESI/ACCORTI**

Reso (2 decimal) **0,00** Acconto (2 decimal) **0,00**

Spese acc. (2 decimal) **0,00**

Class StepCondizioniOrdine

- Dati Lead
- Gestione Messaggi
- Email
- Sms
- Step Offerta
- Offerte
- Sezioni
- Articoli
- Condizione Offerta
- Riepilogo Offerta
- Esito

**Sottostep "Riepilogo Offerta":**

Offerta n° 7876\_113823189 del 26/08/2014  
Importo: 1.920,60 - Totale offerta: 2.323,93

## Class StepOrdineHeader

Descrizione	Allegato	Lingua
Offer - cot estimate	[Non definita]	<a href="#">Anteprima</a> <a href="#">Allega</a>
Offerta con kit	[Non definita]	<a href="#">Anteprima</a> <a href="#">Allega</a>
Offerta Parametrica - Preventivo	[Non definita]	<a href="#">Anteprima</a> <a href="#">Allega</a>
Stampa preventivo online	[Non definita]	<a href="#">Anteprima</a> <a href="#">Allega</a>
Stampa preventivo online_OLD	[Non definita]	<a href="#">Anteprima</a> <a href="#">Allega</a>

Spettabile PELLICONI & C. S.P.A.

VIA EMILIA, 3147

OZZANO DELL'EMILIA(BO), 40064

00502181209

00315140376

numero documento 7876\_113823189

Da consegnare PELLICONI & C. S.P.A.

VIA EMILIA, 3147

OZZANO DELL'EMILIA(BO), 40064

Data doc.	Valido fino al	Agente	TLX00
26/08/2014	26/08/2014		
Listino	Sconto 1%		3
Totale 1.920,60	Totale Ivato 2.323,93	Totale da Pagare 2.323,93	
Valuta EURO	Condizioni pag. Contanti	Sconto pag. % 0	
Mezzo trasport.	Vettore	Porto	
Spese trasport. 0,00	Spese acc. 0,00	Reso 0,00	
Acconto 0,00			

Stato	Codice Prodotto	Descrizione	Prezzo (EUR)	Iva	Prezzo (EUR)

## Class StepRiepilogoOrdine

Ogni Classe genera i controlli lato server e gestisce il codice JS lato client che viene iniettato nella pagina tramite il metodo `GetScriptStepFunctions()` di ciascuna classe. I controlli generati sono inizializzati come proprietà e vengono aggiunti alla pagina nell'override del metodo `AjaxRender()` di ciascuna classe, in fase di chiamata del metodo `OnPreRender()`. Nella classe `StepAjaxMethods` vi sono tutti i metodi AJAX che vengono chiamati dalle varie procedure dello step tra cui l'importante `AjaxUpdateOrdine()` che viene chiamato per il salvataggio dell'intera offerta. Nel file ***COrdini.cs*** vi è il metodo che genera l'inserimento dell'offerta nel database:

```
namespace StepOfferta
{
 public class COrdini
 {
 protected static string sSqlCols = " SELECT Stati.IDStato,Stati.DescrizioneStato AS DESCRIZIONE, CASE WHEN [Conteggia] = 1 THEN 'V' ELSE 'F' END AS CON,CASE W
 protected static string sOrder = " ORDER BY Stati.Tipo ASC, ABS(Stati.Obsoleto), Stati.Priorita ASC ";

 public static string SqlCmbStatiOrdine(...)
 protected static string GetStatiVis(string filtroIniziale)...)

 public static string sqlListaSottocontiCliente(int IDCliente)...)

 /// <summary>
 /// Controlla l'importo minimo, numero prodotti e prezzo per ordine di un cliente .
 /// </summary>
 /// <param name="ImportoMinimoOrdine">Importo minimo impostato</param>
 /// <param name="IDCliente">IDCliente</param>
 /// <param name="OrdiniTempTable">Tabella temporanea</param>
 /// <param name="IDElenco">IDElenco</param>
 /// <returns></returns>
 public static String CheckImportoMinimo(Double ImportoMinimoOrdine, int IDCliente, String OrdiniTempTable, int IDElenco)...)
 private static Boolean LoadTotalePrezzo(ref int num_prodotti, ref double prezzo, ref double prezzoivacompresa, String OrdiniTempTable, int IDCliente, int IDEL

 public static string CreaSQLInsertOfferta(int IDCliente, string Cliente, string ClienteCA, string Oggetto, int Tipo, string Anno, DateTime Data, string Numero
 Double TotImp, Double Tot, Double Totale, Double TotaleIvato, double ScontoPag, double Sconto, double Sconto2, double Reso
 int IDAgente, int IDOperatore, int IDProvenienza, int IDAgenzia, int IDSegnalatore, int IDPortafoglio, string Periodo, int
 string CodAspettoBeni, string NrColli, string CodIva, string Pagamento, string CodListino, string CodValuta, string CodLin
 string Consegna, DateTime ConsegnaEntro, DateTime ConfermaOrdine, string DataEvasione, string Magazzino, string TipoClient
 string Cliente_0, string ClienteSedeOp, string ClienteContattiSedeOp, string TelSedeOp, string IndirizzoSedeOp, string Num
 string Banca, string AccontoData, string AccontoNote, double AccontoImporto, string AccontoReferente, string NoteFattura,
 string IndirizzoCO, string NumeroCivicoCO, string CapCO, string CittaCO, string ProvinciaCO, string ConsensoInformativoCO,
 string CellICO, string EmailICO, string FaxICO, string IDcliInd, string IDClienteCA, string ClienteCA_0, string IDClienteC
 DateTime DataCambioListinoDoc, Double CambioListinoDoc, DateTime DataCambioAziendaDoc, Double CambioAziendaDoc, Double Tot

 public static string GetSottocontoCliente(string IDCliente)...)
 }
}
```

Mentre l'aggiornamento è gestito all'interno del metodo `AjaxUpdateOrdine()` della classe `StepAjaxMethods`:

```

 }
 else
 {
 try
 {
 Siseco.GatEnv.Environment.Database.BeginTransaction();

 Ricalcolo dei totali prima di salvare ...

 //Memorizzo il vecchio stato dell'offerta
 int mIDStatoOld = Siseco.GatEnv.Environment.Database.ExecuteScalar<int>("SELECT Stato FROM Ordini WHERE IDOrdine = " + mIDStatoOld + ";");

 sql = "UPDATE Ordini ";
 sql += "SET AnnoOrdine = " + mAnno + ", ";
 sql += "DataOrdine = " + CSqlParser.CSql(mData, CSqlParser.CSqlDateTimeEnum.Date) + ", ";
 sql += "IDITipo = " + mTipo + ", ";
 sql += "Stato = " + mStato + ", ";
 if (mPeriodo == String.Empty)
 sql += "PeriodoChiusura = NULL, ";
 else
 sql += "PeriodoChiusura = " + CSqlParser.CSql(mPeriodo) + ", ";
 if (mPercChiusura == -1)
 sql += "PercentualeChiusura = NULL, ";
 else
 sql += "PercentualeChiusura = " + mPercChiusura + ", ";
 sql += "Cod_Mag = " + CSqlParser.CSql(mMagazzino) + ", ";
 sql += "OggettoTrattativa = " + CSqlParser.CSql(mOggetto) + ", ";
 sql += "IDProvenienza = " + CSqlParser.CSql(mIDProvenienza) + ", ";
 sql += "Note = " + CSqlParser.CSql(mNote) + ", ";
 sql += "TotDoc = " + CSqlParser.CSql(mTotaleIvato) + ", ";
 //totivaDoc è il totale dell'imposta
 sql += "TotivaDoc = " + CSqlParser.CSql(mTotaleIvato - mTotale) + ", ";
 //totimpDoc è il totale dell'ordine senza IVA
 sql += "TotimpDoc = " + CSqlParser.CSql(mTotale) + ", ";
 sql += "Pagamento = " + CSqlParser.CSql(Pagamento) + ", ";
 sql += "scontopagamento = " + CSqlParser.CSql(ScontoPag) + ", ";
 sql += "codvaluta = " + CSqlParser.CSql(CodValuta) + ", ";
 sql += "sconto = " + CSqlParser.CSql(sconto) + ", ";
 sql += "sconto2 = " + CSqlParser.CSql(sconto2) + ", ";
 sql += "maggiorazione = " + CSqlParser.CSql(maggiorazioneOfferta) + ", ";
 sql += "codlingua = " + CSqlParser.CSql(CodLingua) + ", ";
 sql += "porto = " + CSqlParser.CSql(CodPorto) + ", ";
 sql += "codvettore = " + CSqlParser.CSql(CodVettore) + ", ";
 sql += "ConsegnaEntro = " + CSqlParser.CSql(ConsegnaEntro, CSqlParser.CSqlDateTimeEnum.Date) + ", ";
 sql += "DataConferma = " + ((ConfermaOrdine == DateTime.MinValue) ? "NULL" : CSqlParser.CSql(ConfermaOrdine, CSqlParser.CSqlDateTimeEnum.Date)) + ", ";
 sql += "codlistino = " + CSqlParser.CSql(CodListino) + ", ";
 sql += "spedizione = " + CSqlParser.CSql(Spedizione) + ", ";
 }
 }
}

```

### ATTENZIONE:

La classe [StepOrdineLoader](#) non è personalizzabile perché serve alla Gestione Campagne in quanto vi sono le informazioni necessarie per la corretta costruzione dello step.

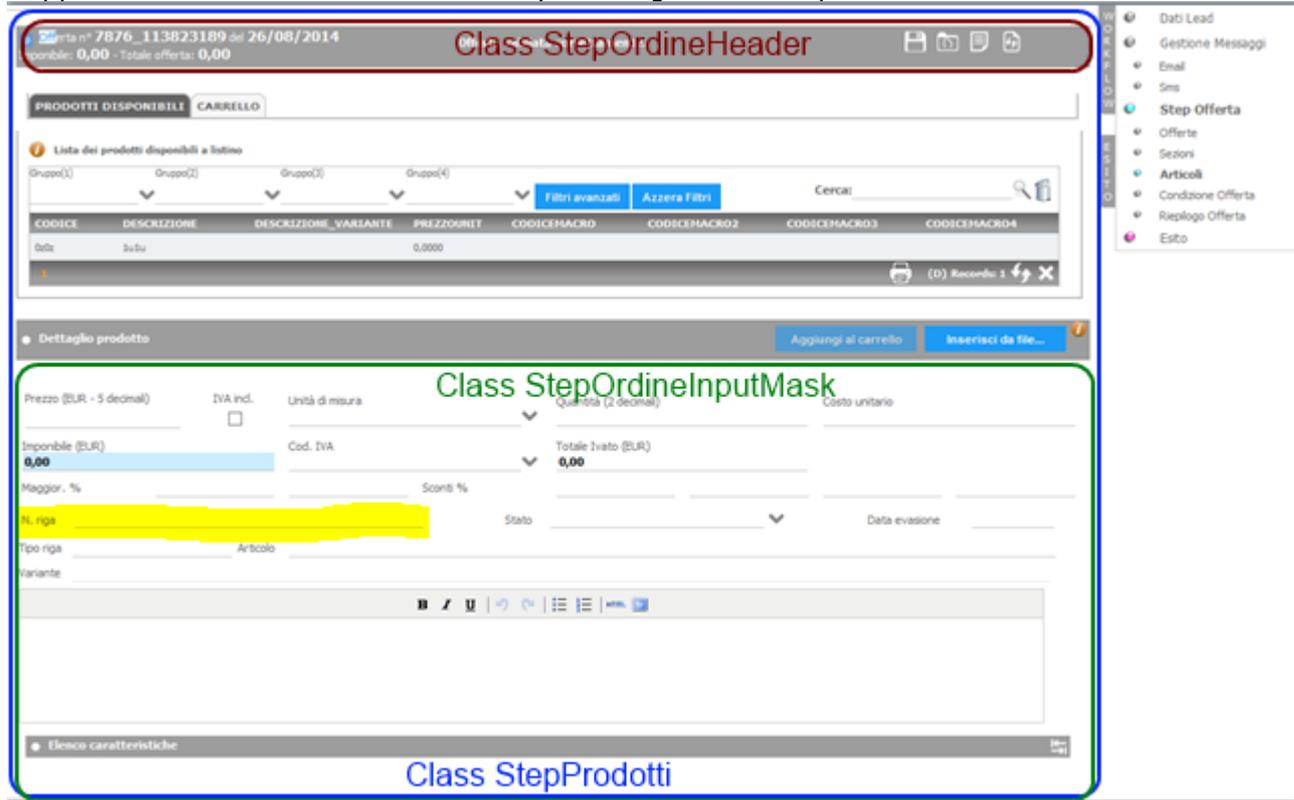
La classe [StepOrdineUpdateToEsito](#) non è personalizzabile, è necessario che sia compilata così come si presenta per la corretta integrazione con la CRM Multimedia.

## Come nascondere o spostare i controlli utente di uno step

E' consigliabile personalizzare lo step in modo tale che ci sia un margine di errore di basso profilo a causa della complessa gestione della Multimedia. Si consiglia pertanto di nascondere i

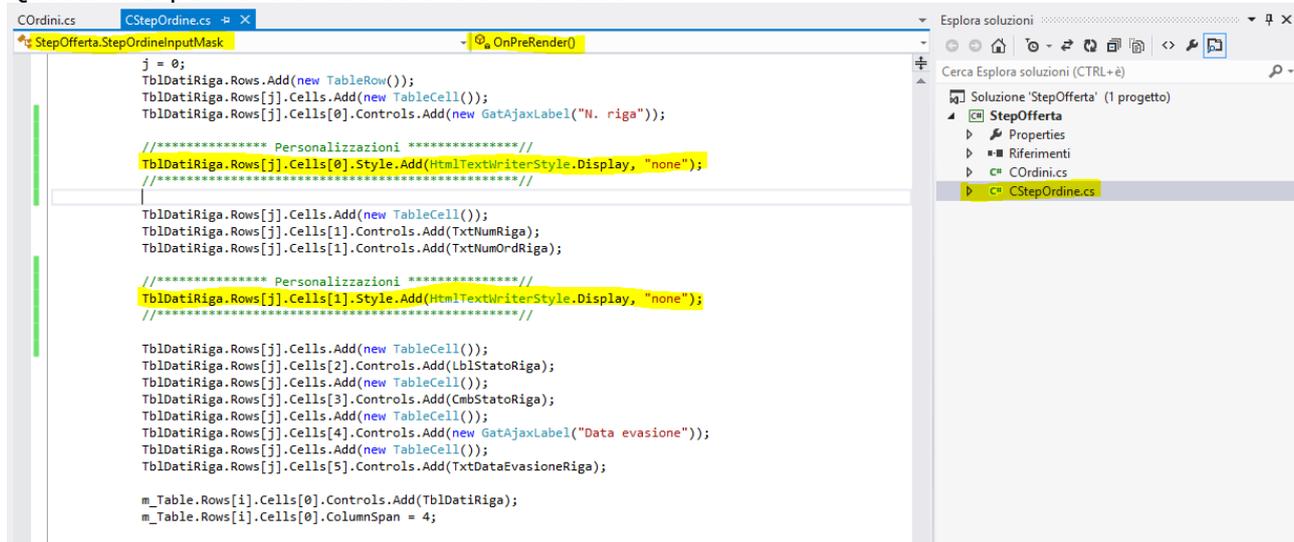
controlli presenti nello step, senza eliminarli, evitando di "rincorrere" tutti i punti in cui vengono gestiti dalla classe.

Supponiamo di voler nascondere il campo "N. riga" nello step "Articoli":



Apriamo quindi la soluzione e posizioniamoci sul metodo **OnPreRender()** della classe **StepOrdineInputMask** nel file **CStepOrdine.cs**, in cui vengono aggiunti nella pagina tutti i controlli utente della classe.

Questa è la porzione di codice che ci interessa:



I controlli vengono aggiunti in una tabella (in questo caso **TblDatiRiga**). L'obiettivo è quello di nascondere il campo **N. riga** che è costituito dall'etichetta (Label) e da 2 controlli di tipo TextBox. Come si vede dall'immagine sopra, il controllo è stato nascosto a livello di cella, aggiungendo l'attributo "Display:none" alle celle all'interno delle quali è stato aggiunto il controllo.

Il risultato del codice sopra è il seguente:

● Dettaglio prodotto Aggiungi al carrello Inserisci da file...

Prezzo (EUR - 5 decimali)  IVA incl.  Unità di misura  Quantità (2 decimali)  Costo unitario

Imponibile (EUR) **0,00** Cod. IVA  Totale Ivato (EUR) **0,00**

Maggior. %  Sconti %

Stato  Data evasione

Tipo riga  Articolo

Variante

**B I U** ↶ ↷ ↺ ↻ ☰ ☲ ☱ HTML

● Elenco caratteristiche H

Supponiamo ora di voler spostare il campo **Tipo riga** nella riga sopra, a fianco del campo **Data evasione**:

**NON PERSONALIZZATO**

```

j = 0;
TblDatiRiga.Rows.Add(new TableRow());
TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[0].Controls.Add(new GatAjaxLabel("N. riga"));

//***** Personalizzazioni *****//
TblDatiRiga.Rows[j].Cells[0].Style.Add(HtmlTextWriterStyle.Display, "none");
//***** Personalizzazioni *****//

TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[1].Controls.Add(TxtNumRiga);
TblDatiRiga.Rows[j].Cells[1].Controls.Add(TxtNumOrdRiga);

//***** Personalizzazioni *****//
TblDatiRiga.Rows[j].Cells[1].Style.Add(HtmlTextWriterStyle.Display, "none");
//***** Personalizzazioni *****//

TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[2].Controls.Add(LblStatoRiga);
TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[3].Controls.Add(CmbStatoRiga);
TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[4].Controls.Add(new GatAjaxLabel("Data evasione"));
TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[5].Controls.Add(TxtDataEvasioneRiga);

m_Table.Rows[i].Cells[0].Controls.Add(TblDatiRiga);
m_Table.Rows[i].Cells[0].ColumnSpan = 4;

i++;

m_Table.Rows.Add(new TableRow());
m_Table.Rows[i].Cells.Add(new TableCell());
m_Table.Rows[i].Cells[0].Width = Unit.Percentage(5);
m_Table.Rows[i].Cells[0].Controls.Add(new GatAjaxLabel(T("Tipo riga")));
m_Table.Rows[i].Cells.Add(new TableCell());
m_Table.Rows[i].Cells[1].Width = Unit.Percentage(5);
m_Table.Rows[i].Cells[1].Controls.Add(TxtTipoProdotto);
m_Table.Rows[i].Cells.Add(new TableCell());
m_Table.Rows[i].Cells[2].Width = Unit.Percentage(5);
m_Table.Rows[i].Cells[2].Controls.Add(new GatAjaxLabel(T("Articolo")));
m_Table.Rows[i].Cells.Add(new TableCell());
m_Table.Rows[i].Cells[3].Width = Unit.Percentage(85);
m_Table.Rows[i].Cells[3].Controls.Add(TxtCodice);
m_Table.Rows[i].Cells[3].Controls.Add(txtProgrVart);
m_Table.Rows[i].Cells[3].Controls.Add(HidIDDettagliOrdini);
m_Table.Rows[i].Cells[3].Controls.Add(TxtDescrizione);
i++;

```

**PERSONALIZZATO**

```

j = 0;
TblDatiRiga.Rows.Add(new TableRow());
TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[0].Controls.Add(new GatAjaxLabel("N. riga"));

//***** Personalizzazioni *****//
TblDatiRiga.Rows[j].Cells[0].Style.Add(HtmlTextWriterStyle.Display, "none");
//***** Personalizzazioni *****//

TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[1].Controls.Add(TxtNumRiga);
TblDatiRiga.Rows[j].Cells[1].Controls.Add(TxtNumOrdRiga);

//***** Personalizzazioni *****//
TblDatiRiga.Rows[j].Cells[1].Style.Add(HtmlTextWriterStyle.Display, "none");
//***** Personalizzazioni *****//

TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[2].Controls.Add(LblStatoRiga);
TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[3].Controls.Add(CmbStatoRiga);
TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[4].Controls.Add(new GatAjaxLabel("Data evasione"));
TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[5].Controls.Add(TxtDataEvasioneRiga);

//***** Personalizzazioni *****//
TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[6].Width = Unit.Percentage(5);
TblDatiRiga.Rows[j].Cells[6].Controls.Add(new GatAjaxLabel(T("Tipo riga")));
TblDatiRiga.Rows[j].Cells.Add(new TableCell());
TblDatiRiga.Rows[j].Cells[7].Width = Unit.Percentage(5);
TblDatiRiga.Rows[j].Cells[7].Controls.Add(TxtTipoProdotto);
//***** Personalizzazioni *****//

m_Table.Rows[i].Cells[0].Controls.Add(TblDatiRiga);
m_Table.Rows[i].Cells[0].ColumnSpan = 4;

i++;

m_Table.Rows.Add(new TableRow());
m_Table.Rows[i].Cells.Add(new TableCell());
m_Table.Rows[i].Cells[0].Width = Unit.Percentage(5);
m_Table.Rows[i].Cells[0].Controls.Add(new GatAjaxLabel(T("Articolo")));
m_Table.Rows[i].Cells.Add(new TableCell());
m_Table.Rows[i].Cells[1].Width = Unit.Percentage(85);
m_Table.Rows[i].Cells[1].Controls.Add(TxtCodice);
m_Table.Rows[i].Cells[1].Controls.Add(txtProgrVart);
m_Table.Rows[i].Cells[1].Controls.Add(HidIDDettagliOrdini);

```

Come si deduce dall'immagine, le due celle che corrispondono rispettivamente alla Label "Tipo riga" e al rispettivo controllo TextBox, sono state aggiunte alla riga della tabella superiore, modificando opportunamente tutti gli indici della collection Cells. Il risultato del codice sopra è il seguente:

Prezzo (YEN - 2 decimali)	IVA incl. <input type="checkbox"/>	Unità di misura	▼	Quantità (2 decimali)	Costo unitario
Imponibile (YEN)		Cod. IVA	▼	Totale Ivato (YEN)	
<b>0,00</b>				<b>0,00</b>	
Maggior. %		Sconti %			
Stato	▼	Data evasione			Tipo riga
Articolo					
Variante					

---

## 3.3 Debug e test

La soluzione che viene distribuita contiene nella cartella Bin tutti gli assembly (DLL) impresoft necessari per poter personalizzare il prodotto utilizzando l'SDK.

Una volta scritto il codice, compilare la soluzione in modalità DEBUG utilizzando l'apposito selettore di Visual Studio .Net. In caso di assenza di errori operare nel seguente modo:

- Registrare la DLL creata nel database del CRM come indicato precedentemente (si consiglia di salvarsi gli script di registrazione per poterli poi riutilizzare per la fase di rilascio in produzione)
- Inserire nel codice sorgente i breakpoint desiderati
- Copiare la DLL e il file PDB generati come risultato della compilazione (normalmente localizzato nella directory BIN\DEBUG della soluzione) nel percorso indicato negli script di registrazione del plugin (ad esempio in C:\CRM\Addins) e nella directory BIN del CRM: potrebbe essere necessario riavviare il processo di IIS nel caso che le DLL risultassero in uso dal processo (è sufficiente lanciare il comando *IISRESET* dalla finestra dei comandi amministrativi di Windows)
- Far partire il browser (ad esempio Firefox) e lanciare il CRM
- Da Visual Studio .Net, attaccare il processo di IIS (w3wp.exe)
- Utilizzare il CRM navigando sulla personalizzazione implementata (form/griglia oppure step personalizzato nella Multimedia)
- Utilizzare le normali procedure di Debug

---

## 3.4 Rilascio in produzione di SDK

Quando si è raggiunto un livello qualitativo desiderato del codice realizzato, sarà possibile effettuare il rilascio in produzione del lavoro realizzato; dovendo lavorare direttamente sul database di produzione è consigliato operare in assenza di attività da parte dei normali utilizzatori del CRM e di avere la possibilità di riavviare il server IIS di produzione. Seguire i seguenti step per il rilascio in produzione:

- Modificare opportunamente gli script di registrazione delle DLL utilizzati per la fase di debug/test per puntare ai percorsi di produzione e registrare le DLL sul database di produzione
- Compilare il codice sorgente impostando il selettore di Visual Studio .Net nella modalità RELEASE
- Copiare la DLL generata come risultato della compilazione (normalmente localizzato nella directory BIN\RELEASE della soluzione) nel percorso indicato negli script di registrazione del plugin (ad esempio in C:\CRM\Addins) e nella directory BIN del CRM: potrebbe essere necessario riavviare il processo di IIS nel caso che le DLL risultassero in uso dal processo (è sufficiente lanciare il comando *IISRESET* dalla finestra dei comandi amministrativi di Windows)

## 4. CONCLUSIONI

Come si è potuto evincere dagli esempi forniti esistono diverse profondità di personalizzazione che si possono ottenere seguendo la strada del Simple# oppure dell'SDK.

A scapito di una forte conoscenza del prodotto e sicuramente di una "difficoltà" iniziale dovuta alla complessità della personalizzazione, l'SDK consente di personalizzare aspetti del CRM che non sono possibili con il Simple# essendo l'SDK molto più potente perché agisce direttamente a livello di codice di sviluppo; viceversa anche se in maniera più limitata, il Simple# consente con molta facilità di personalizzare il CRM rispettando però semantiche e logiche di business predefinite.

La scelta di un strada piuttosto che un'altra è il risultato di una riflessione comprendente aspetti di natura tecnica, usabilità per l'utente finale e sicuramente economico/commerciale.