

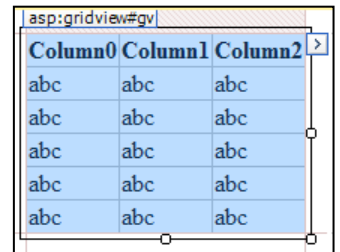
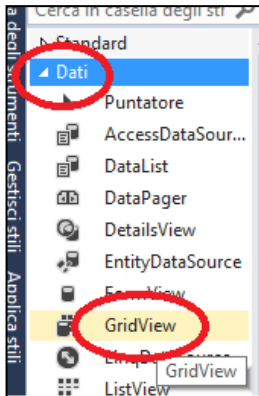
IL CONTROLLO WEB “GRIDVIEW” NELLE PAGINE ASPX

La classe GridView e la Visualizzazione di Tabelle di Dati in una Pagina ASPX

Il controllo web **GridView** permette di visualizzare con facilità una *Tabella di Dati* su una pagina web ASPX.

Nella *Casella degli Strumenti* di Visual Studio, il controllo **GridView** è presente nel **gruppo Dati** (figura a sinistra).

Una volta posizionato sulla *finestra di progettazione* della pagina ASPX, il controllo **GridView** appare come nella figura a destra.



E' necessario anzitutto accedere a un **DataBase** (o a un *File*), **caricare la tabella di dati** da visualizzare in un **oggetto DataTable** e successivamente **Associare l'oggetto DataTable al controllo GridView**, che provvede a visualizzare i dati sulla pagina ASPX.

Il **GridView** è analogo al controllo di form *DataGridView*, utilizzato nelle normali applicazioni Windows Form, ma al contrario di esso, **non ha una capacità autonoma di memorizzare i dati al proprio interno**: deve essere sempre associato ad una *sorgente dati*, tipicamente un oggetto *DataTable*.

Per **Associare un DataTable a un controllo GridView**, si usano la **proprietà DataSource** e il **metodo DataBind**.

In questo esempio, **gv** è il nome del controllo *GridView* presente sulla pagina ASPX e **dt** è il nome del *DataTable* in cui sono stati preventivamente caricati dati provenienti da un DataBase o da un File:

```
gv.DataSource = dt;
gv.DataBind ( );
```

IdSoggetto	Nominativo	Indirizzo	Telefono
2	Paperino	Piazza del Papero	987654
3	Pippo	Largo del Tonto	597416
1	Topolino	Via del Pollo	123456

Queste semplici operazioni sono sufficienti a visualizzare automaticamente l'intera tabella di dati sulla pagina.

Personalizzazione delle Colonne in un controllo GridView


Il controllo **GridView** compone automaticamente la tabella secondo dei criteri standard che potrebbero non soddisfare le esigenze del programmatore. E' allora possibile applicare una **Composizione Personalizzata delle Colonne**.

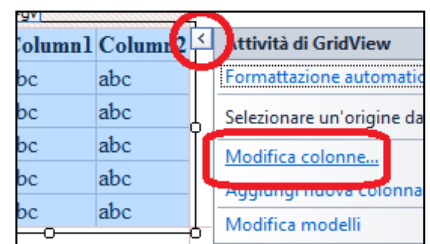
La **proprietà AutoGenerateColumns** (normalmente impostata a True) può essere posta a False per **Disattivare la Composizione Automatica delle Colonne** in un **GridView**.

Si noti come l'impostazione di **AutoGenerateColumns** v'è effettuata *prima di associare il GridView al DataTable*:

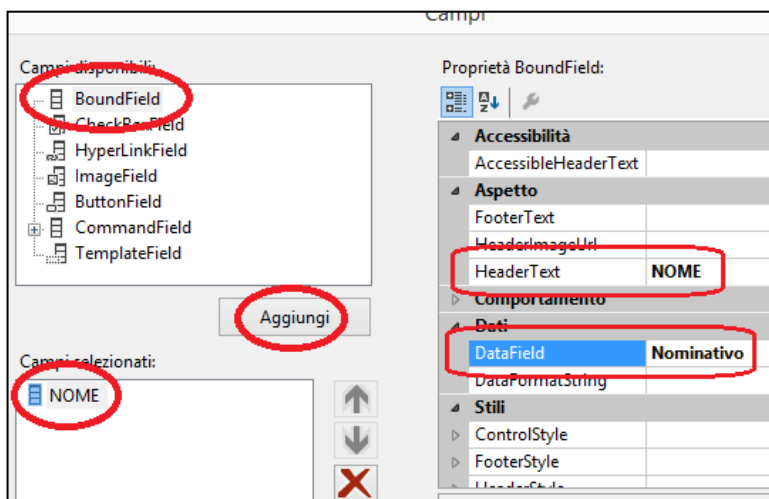
```
gv.AutoGenerateColumns = False;
gv.DataSource = dt;
gv.DataBind ( );
```

Se si disattiva la generazione automatica delle colonne, è necessario **definire manualmente la struttura delle colonne** del **GridView**.

E' possibile **Personalizzare le Colonne di un GridView** in modo *interattivo*, direttamente dalla *Finestra di Progettazione*, scegliendo il **comando Modifica Colonne**, accessibile dal **pulsante** , situato nell'angolo in alto a destra del controllo **GridView**.



La conseguente **finestra Campi**, consente di definire liberamente le colonne da visualizzare, la loro tipologia (dati testuali, immagini, checkbox, pulsanti, link, ecc.) e le relative proprietà per la personalizzazione delle colonne stesse.



Per **Creare una Nuova Colonna**, sceglierne la **Tipologia** e fare click sul **pulsante Aggiungi**.

Le tipologie disponibili sono diverse: è possibile visualizzare colonne con *dati numerici o alfanumerici* (**BoundField**), o *booleani* (**CheckBoxField**) ma anche colonne di *hyperlink*, di *immagini*, di *pulsanti di comando* e anche colonne basate su un *controllo web liberamente definito* (**TemplateField**).

La **sezione Campi Selezionati** elenca tutte le colonne create: è possibile **Selezionare**, **Eliminare** e **modificare l'Ordine** delle colonne.

Selezionata una colonna, è possibile accedere alle **Proprietà della Colonna**.

La **proprietà HeaderText** determina l'**Intestazione** che appare in testa alla colonna selezionata, sulla pagina web. Questa proprietà imposta anche il nome che appare, per la colonna, nella *sezione Campi Selezionati*.

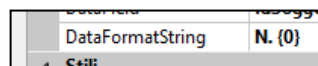
La **proprietà DataField**, di *fondamentale importanza*, stabilisce **da quale Colonna del DataTable devono essere prelevati i dati da visualizzare nella colonna del GridView** di cui si sta impostando la proprietà stessa.

- ✎ In pratica, la **proprietà DataField** "associa" alla **relativa colonna del GridView**, una **colonna del DataTable**. Senza questa associazione, il GridView non saprebbe "da dove prelevare" i dati da visualizzare e la colonna del GridView resterebbe in bianco. Si ricordi che il GridView viene sempre "agganciato" a un DataTable con la proprietà DataSource.
- ✎ Se la **proprietà DataField** indica una **colonna che NON ESISTE nel DataTable**, si verifica un errore in fase di generazione della pagina ASPX.
- ✎ Nell'esempio riportato nella figura, saranno i dati presenti nella **colonna Nominativo del DataTable** a essere visualizzati nella **colonna NOME del GridView**.

La **proprietà DataFormatString** consente di stabilire in che "forma" deve essere visualizzato il dato nella colonna: anzitutto è possibile **definire liberamente la stringa (cioè la sequenza di caratteri) da visualizzare** e, in essa, si indica **la posizione in cui si desidera posizionare il dato, con la notazione {0}**.

- ✎ Ad esempio, se una **colonna Id** del GridView, contiene la "numerazione" dei soggetti elencati, si può abbellire la visualizzazione della tabella, impostando la **proprietà DataFormatString** della **colonna Id** con la stringa: **N. {0}**

In tal modo il GridView visualizzerà la colonna sostituendo ogni volta al gruppo di caratteri {0}, il reale valore del campo, ottenendo così la visualizzazione **N. 1 N. 2 ... ecc.**



ID	NOME
N. 2	Paperino
N. 3	Pippo
N. 1	Topolino

Questa proprietà permette, fra l'altro, di **definire il Formato per i Dati Numerici** (*numeri interi, numeri con decimali, valute, percentuali, date, ore, ecc.*), usando la **notazione {0:x}**, dove "x" è una **singola lettera che specifica il formato numerico desiderato**.

- ✎ Ad esempio, se una **colonna Prezzo** del GridView, contiene un importo in Euro, è possibile impostare la **proprietà DataFormatString** della **colonna Prezzo** con la stringa: **{0:C}** ove la lettera "C" sta per "currency" ossia "valuta". Si ottiene € 2,00 € 3,00 ecc.
- ✎ Allo stesso modo è anche possibile specificare il **numero di decimali desiderati, indicandone il numero dopo la lettera: {0:C3}** (vedi figura più a destra).

Prezzo	Prezzo
€ 2,00	€ 2,000
€ 3,00	€ 3,000
€ 1,00	€ 1,000

Per conoscere l'elenco di tutte le opzioni per la formattazione dei valori numerici, si vedano gli ultimi paragrafi.

Personalizzazione dell'Aspetto Grafico di un controllo GridView

Il controllo GridView, come ogni altro elemento html in una pagina web, esso **eredita uno Stile dell'Elemento HTML che lo contiene**. Se non altrimenti specificato, diversi aspetti grafici del GridView sono determinati da questo.

Ad esempio, se il tag html del controllo GridView è "racchiuso" in un tag html DIV e ad esso è applicato uno stile che stabilisce il tipo (Arial) e le dimensioni (small) del carattere ...

```
<div style="font-family:Arial;font-size:small">
  <asp:GridView ID="gv" runat="server" ...
```

... elemento DIV con uno stile

... tag html del controllo GridView

... allora anche il GridView viene visualizzato con le indicazioni espresse nello stile del DIV.

In alternativa, il controllo GridView permette un **completo controllo del proprio aspetto grafico** grazie a una serie di **Proprietà che consentono di specificare lo Stile delle varie parti** che lo compongono.

AutoGenerate	False
BackColor	
BackImageUrl	
BorderColor	
BorderStyle	NotSet
BorderWidth	
Caption	
CaptionAlign	NotSet
CellPadding	-1
CellSpacing	0
Columns	(Raccolta)
CssClass	
DataKeyNames	

Anzitutto, come ogni altro controllo web ASPX, il GridView prevede le solite e note **proprietà grafiche generali**, comuni a tutti i controlli, che ne influenzano la rappresentazione (**BackColor, ForeColor, Font, ecc.**).

E' disponibile anche la **proprietà CssClass** che permette di specificare il **Nome di uno Stile** definito nella pagina o in un Foglio di stile (.css) allegato alla pagina, applicandone, con una sola impostazione, tutte le regole in esso contenute.



Come è noto i Fogli di Stile (CSS) in HTML, permettono di raggruppare le definizioni di molte caratteristiche grafiche: se "centralizzati" in un file CSS applicato a tutte le pagine di un sito web, consentono di uniformarne efficacemente l'aspetto grafico.

Generalmente, le **proprietà specifiche del controllo** "sovrascrivono" quelle dello stile specificato tramite la proprietà **CssClass** e queste ultime, "sovrascrivono" quelle ereditate dall'elemento html che contiene il controllo GridView.

Per controllare l'aspetto grafico delle sue varie componenti (*testata, righe, colonne, celle, ecc.*), sono previste delle proprietà specifiche che chiameremo **Proprietà di Stile**.



Le **Proprietà di Stile** disponibili sono molte, proprio perché il GridView è un oggetto complesso con diverse "componenti": *HeaderStyle, FooterStyle, RowStyle, SelectRowStyle, PagerStyle, ecc.* Come si vede esse sono facilmente riconoscibili perché *terminano con la parola "Style"*.

Le **Proprietà di Stile** sono articolate in diverse **sottoproprietà** per stabilire i singoli aspetti dell'aspetto grafico. Anch'esse prevedono una **sottoproprietà CssClass** per **specificare il Nome di uno Stile** da applicare in blocco, proprio come per la proprietà **CssClass** globale.



Nella figura a destra è visualizzata una porzione delle proprietà del controllo GridView: si può vedere la **Proprietà di Stile "HeaderStyle"** (stile della testata) che è "esplosa" nelle sue sottoproprietà (*BackColor, BorderColor, ecc.*). Fra esse è presente la **proprietà CssClass** che permette di **indicare direttamente il Nome di uno Stile** definito nella Pagina o in un file CSS.

GridLines	Both
HeaderStyle	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
CssClass	
Font	
ForeColor	
Height	
HorizontalAlign	NotSet
VerticalAlign	NotSet
Width	
Wrap	True
Height	74px


E' possibile **Controllare l'aspetto grafico di ogni Singola Colonna** del GridView, accedendo alle **Proprietà di Stile** presenti fra le proprietà delle colonne, nella già vista *finestra Campi*.

ButtonField	
CommandField	
TemplateField	
Aggiungi	
Campi selezionati:	
Articolo	
Prezzo	
HeaderText	Prezzo
Comportamento	
Dati	
DataField	Prezzo
DataFormatString	{0:C3}
Stili	
ControlStyle	
FooterStyle	
HeaderStyle	
ItemStyle	
ControlStyle	

Per ogni singola colonna definita sono disponibili le seguenti **Proprietà di Stile**:

- **ControlStyle** opera sul "Controllo" (ossia *TextBox, CheckBox, Immagine, Pulsante, ecc.*) della colonna;
- **FooterStyle e HeaderStyle** operano sulla "Testata" e sul "Piede" della colonna;
- **ItemStyle** opera sulla "cella" della colonna;


Fra le sottoproprietà più importanti di *ItemStyle*, oltre a quelle note, è utile evidenziare:

- la **proprietà Width** che consente di specificare la **Larghezza della Colonna**.
 Come per le *misure utilizzate negli stili HTML* anche qui è possibile esprimere la larghezza in **pixel** (es.: 200px), in **centimetri** (3cm), ecc.
- le **proprietà HorizontalAlign e VerticalAlign** per l'impostazione degli **Allineamenti** del dato all'interno della cella.
- la **proprietà Wrap** per consentire o no il ritorno a capo automatico del testo.
- la **proprietà Font** (e le sue sottoproprietà) per definire **Tipo, Dimensioni, ecc. del Carattere** da usare che, quindi, può essere *diversificato da colonna a colonna*.
- la solita **proprietà CssClass** per specificare il Nome di uno Stile e applicare alla cella della colonna.

ItemStyle	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
CssClass	
Font	
ForeColor	
Height	
HorizontalAlign	NotSet
VerticalAlign	NotSet
Width	
Wrap	True

Le *proprietà Width e HorizontalAlign* esistono anche come *proprietà "globali"* del GridView.


La **proprietà globale Width** consente di **Forzare la Larghezza dell'intero controllo GridView a uno specifico valore**: la larghezza delle singole colonne sarà "distribuita" sulla larghezza complessiva che è comunque "bloccata" al valore specificato.

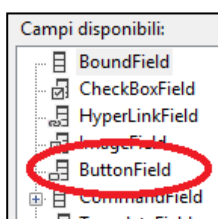
-  Se non si impostano valori né per la *proprietà Width globale* né per la *proprietà Width delle singole colonne*, la larghezza della tabella **viene "adattata" automaticamente al contenuto**: in tal modo la larghezza complessiva varia a seconda dei dati. A volte questa "variabilità" può essere dannosa ai fini del controllo del layout complessivo della pagina.

La **proprietà globale HorizontalAlign** permette di controllare l'**Allineamento Orizzontale dell'intero GridView** che, non essendo un elemento "testuale", non risente delle impostazioni di allineamento dei normali stili HTML.

Pulsanti di Comando nelle Colonne di un controllo GridView

E' molto ricorrente che un GridView serva a elencare delle istanze allo scopo di selezionarne una ed effettuare su di essa una o più specifiche operazioni.


-  Ad esempio, in una Rubrica Telefonica "on-line", il GridView, oltre a *visualizzare* i Nominativi presenti in rubrica, deve anche consentire di *selezionare* uno di essi allo scopo di *modificare* i suoi dati o di *eliminare* il nominativo dalla rubrica stessa.



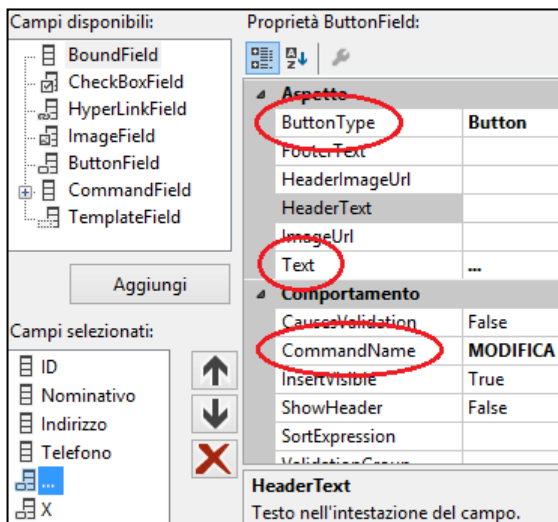
Per consentire l'**Esecuzione di una Operazione su una delle Righe** del GridView, è consigliabile utilizzare una **Colonna di tipo ButtonField**.

In ogni cella della colonna di tipo ButtonField **appare un pulsante distinto**: l'utente, effettuando un click su uno di essi indica esplicitamente "*su quale riga*" desidera effettuare l'operazione prevista dalla colonna.

-  E' possibile creare più colonne di tipo ButtonField per consentire diverse operazioni.

-  Ad esempio, nella Rubrica Telefonica "on-line", sono previste due colonne di tipo ButtonField (in figura sono cerchiare in rosso e verde): una per la Modifica dei dati (pulsanti ...) di un nominativo, l'altra per l'Eliminazione (pulsanti X).

ID	Nominativo	Indirizzo	Telefono		
2	Paperino	Piazza del Papero	987654	...	X
3	Pippo	Largo del Tonto	597416	...	X
1	Topolino	Via del Pollo	123456	...	X



La **proprietà ButtonType** della *colonna ButtonField* consente di scegliere il modo in cui far “apparire” il pulsante: le scelte sono **Button** (pulsante), **Image** (immagine) e **Link** (Collegamento).

La **proprietà Text** stabilisce il **Testo** che deve apparire sul pulsante.

La **proprietà CommandName** (fondamentale per il funzionamento della colonna di pulsanti) **attribuisce un Nome all'operazione** prevista la colonna di pulsanti.

il Nome può essere scelto liberamente ed è importante perché consente al codice di risalire su quale delle colonne ButtonField l'utente ha effettuato il click.

Come già detto, nel nostro esempio le *colonne ButtonField* sono due, una per *modificare* l'altra per *eliminare*. Esse si chiamano “...” e “X” e le rispettive *proprietà CommandName* sono impostate a “**MODIFICA**” e “**ELIMINA**”.

Un *click* su uno dei pulsanti di una *colonna ButtonField* genera l'**evento RowCommand** del controllo GridView. Questo provoca un *PostBack*, consentendo al server di effettuare l'operazione richiesta.

```
private void gv_RowCommand (object sender, GridViewCommandEventArgs e )
```

Come per tutti gli eventi, anche per l'*evento RowCommand*, viene passato un **parametro “e”** contenente *informazioni aggiuntive* sull'evento stesso.

La **proprietà e.CommandArgument** contiene l'**Indice della Riga** su cui è stato effettuato il click.

La **proprietà e.CommandName** contiene il **Nome dell'Operazione della Colonna** su cui è stato effettuato il click.

E' conveniente che nel GridView sia presente come prima colonna, visibile o no, **una colonna con le chiavi primarie** delle istanze elencate. Tramite l'indice di riga ottenuto con *e.CommandArgument* si può quindi *accedere alla riga del GridView che è stata cliccata e leggere la chiave primaria* da utilizzare per individuare correttamente il record su cui operare.

```
int RigaSelezionata = Convert.ToInt32 (e.CommandArgument) ... recupero l'indice della riga cliccata
```

```
int Id = Convert.ToInt32 (gv.Rows[RigaSelezionata].Cells[0].Text)... accedo alla riga e, dalla prima cella, recupero l'Id
```

La **proprietà e.CommandName** ci permette di distinguere quale azione effettuare:

```
if ( e.CommandName == “MODIFICA” ) ... verifico se il Nome dell'Operazione della colonna cliccata è “MODIFICA”
    ... azioni che modificano i dati ... ... modifico il record con chiave primaria pari a Id
else
    if ( e.CommandName == “ELIMINA” ) ... verifico se il Nome dell'Operazione della colonna cliccata è “ELIMINA”
        ... azioni che eliminano il record ... ... elimino il record con chiave primaria pari a Id
```

Non si dimentichi che il modo più efficace per **identificare univocamente un record** in una tabella di un DataBase (o di un File Dati) è sempre la **Chiave Primaria**. Utilizzare direttamente la “posizione” in una griglia visualizzata temporaneamente (come quella fornita da *e.CommandArgument*) per decidere su quale record agire, non è una buona idea. Ecco uno dei motivi...

Supponiamo che al momento del caricamento della pagina nel browser la posizione del record su cui agire era, ad esempio, la *posizione 5*. Nel periodo di tempo *fra il caricamento della pagina e la generazione dell'evento*, un altro utente potrebbe aver *cancellato un record in una posizione precedente* ... la *posizione 5*, ora, non indica più il record che è stato selezionato, bensì quello successivo! Se l'operazione fosse un'eliminazione, cancelleremmo il record sbagliato!

Caratteri di Formato per la Formattazione di Dati Numerici in un GridView

Carattere di formato	Descrizione	Esempio
C o c	I valori numerici sono visualizzati in formato valuta. È possibile specificare il numero di posizioni decimali.	{0:C} 123.456 -> \$123.46 {0:C3} 123.456 -> \$123.456
D o d	Visualizza i valori interi in formato decimale. È possibile specificare il numero di cifre: Anche se il tipo viene definito "decimale", i numeri vengono formattati come interi.	{0:D} 1234 -> 1234 {0:D6} 1234 -> 001234
E o e	I valori numerici sono visualizzati in formato scientifico (esponenziale). È possibile specificare le posizioni decimali.	{0:E} 1052.0329112756 -> 1.052033E+003 {0:E2} -1052.0329112756 -> -1.05e+003
F o f	I valori numerici sono visualizzati in formato fisso. È possibile specificare il numero di posizioni decimali.	{0:F} 1234.567 -> 1234.57 {0:F3} 1234.567 -> 1234.567
G o g	Visualizza i valori numerici in un formato generico (la più compatta tra la notazione a virgola fissa e quella scientifica). È possibile specificare il numero di cifre significative.	{0:G} -123.456 -> -123.456 {0:G2} -123.456 -> -120
N o n	Visualizza i valori numerici in formato numerico (includendo i separatori di gruppo e il segno negativo facoltativo). È possibile specificare il numero di posizioni decimali.	{0:N} 1234.567 -> 1,234.57 {0:N4} 1234.567 -> 1,234.5670
P o p	Visualizza i valori numerici in formato percentuale. È possibile specificare il numero di posizioni decimali.	{0:P} 1 -> 100.00% {0:P1} .5 -> 50.0%
R o r	Visualizza Single, Double, o valori BigInteger in formato round trip BigInteger.	{0:R} 123456789.12345678 -> 123456789.12345678
X o x	Visualizza i valori interi in formato esadecimale. È possibile specificare il numero di cifre:	{0:X} 255 -> FF {0:x4} 255 -> 00ff

Caratteri di Formato per la Formattazione di Date/Ore in un GridView

Carattere di formato	Descrizione	Esempio
d	Schema di data breve.	{0:d} 6/15/2009 1:45:30 PM -> 6/15/2009
D	Schema di data estesa.	{0:D} 6/15/2009 1:45:30 PM -> Monday, June 15, 2009
f	Schema di data/ora completa (ora breve).	{0:f} 6/15/2009 1:45:30 PM -> Monday, June 15, 2009 1:45 PM
F	Schema di data/ora completa (ora estesa).	{0:F} 6/15/2009 1:45:30 PM -> Monday, June 15, 2009 1:45:30 PM
g	Schema di data/ora generale (ora breve).	{0:g} 6/15/2009 1:45:30 PM -> 6/15/2009 1:45 PM
G	Schema di data/ora generale (ora estesa).	{0:G} 6/15/2009 1:45:30 PM -> 6/15/2009 1:45:30 PM
M o m	Schema di mese/giorno.	{0:M} 6/15/2009 1:45:30 PM -> June 15
O o o	Schema di data/ora di round trip.	{0:o} 6/15/2009 1:45:30 PM -> 2009-06-15T13:45:30.0900000
R o r	Modello RFC1123	{0:R} 6/15/2009 1:45:30 PM -> Mon, 15 Jun 2009 20:45:30 GMT
s	Schema di data/ora ordinabile.	{0:s} 6/15/2009 1:45:30 PM -> 2009-06-15T13:45:30
t	Schema di ora breve.	{0:t} 6/15/2009 1:45:30 PM -> 1:45 PM
T	Schema di ora estesa.	{0:T} 6/15/2009 1:45:30 PM -> 1:45:30 PM
u	Schema di data/ora ordinabile universale.	{0:u} 6/15/2009 1:45:30 PM -> 2009-06-15 20:45:30Z
U	Schema di data/ora completa universale.	{0:U} 6/15/2009 1:45:30 PM -> Monday, June 15, 2009 8:45:30 PM
Y o y	Schema di mese e anno.	{0:Y} 6/15/2009 1:45:30 PM -> June, 2009