BasicDAL ORM

Object Relational Mapper per ADO.NET

Versione 2.0

Data 05/11/2018

Sommario

1 Scopo del documento

Il presente documento ha come scopo quello di documentare l'ORM (Object Relationa Mapper) BasicDAL nelle sue parti e nel suo funzionamento.

2 Introduzione

BasicDAL è una implementazione di un Object Relational Mapper (ORM) e un Data Access Layer (DAL) basato sul pattern Active Record. Lo scopo principale di BasicDAL è quello di fornire una serie di oggetti applicativi idonei a implementare le interfacce necessarie a garantire le funzionalità di persistenza da e verso generici sistemi di basi dati di tipo RDBMS. BasicDAL è basato sull'interfaccia .NET ADO.NET.

2.1 Gli oggetti principali di BasicDAL

Come qualsiasi altra implementazione di un ORM e di un DAL operante in ambito RDBMS gli oggetti principali di BasicDAL sono:

- Configurazione
- Connessione e contesto di Transazione
- Entità/Tabella
- Colonne costituenti le entità/tabelle
- Selezione dei dati

La mappatura fra questi oggetti base e gli oggetti BasicDAL è la seguente

Oggetto generico ORM/DAL	Oggetto proprio BasicDAL
Configurazione	DbConfig
Connessione e contesto di Transazione	DbConfig
Entità/Tabella	DbObject
Colonne costituenti le entità/tabelle	DbColumn e collezioni di DbColumns
Selezione dei dati	DbFilter, DbFilters, DbFilterGroups

La relazione fra gli oggetti è la seguente:

DbConfig -> DbObject –> DbColumns -> DbColumn

- DbConfig contiene la configurazione, la connessione ed il contesto di transazione verso un sistema di Database. Un DbConfig può connettersi ad un solo tipo di Database per volta. Possono esistere molteplici DbConfig che puntano al medesimo database;

- DbObject si trova sempre in uno specifico contesto DbConfig. Un DbConfig può ospitare un numero indefiniti di DbObject. Un DbObject può essere istanziato molteplici volte all'interno dello stesso contesto DbConfig. Un DbObjet può mappare oggetti database di tipo Table, View, Stored Procedure o Funzioni che restituiscono valori scalari o recordsets;

- DbObject contiene una collection di DbColum (colonne). Ogni istanza di DbColum descrive un attributo su cui è mappato il corrispondente attributo/colonna di un oggetto database gestito;

- DbObject contiene un insieme di DbFilterGroups a loro volta definiti in termini di DbFilters. Un DbFilters è una collection di oggetti DbFilter. Un DbFilter è l'insieme minimale che compone una clausola di query (Clausola di Where in SQL) nella forma "<attributo><operatore ricerca><valore>";

- I singoli DbFilter sono messi in relazione logica con degli operatori logici AND, OR. Alla stessa maniera i DbFilterGroups sono messi in relazione logica con i medesimi operatori logici;

- La composizione degli DbFilterGroups detemina il valore della clausola di selezione dell'Active Record sottostante l'oggetto DbObject. Essendo basato su Active Record il DbObject per il quale è stato invocato il metodo di ".LoadAll()" (o il suo alias ".Open()") contiene un insieme iterabile di oggetti DbColums . L'iterazione avviene tramite i metodi di ".MoveFirst(), .MoveNext(), .MovePrevious(), .MoveLast();". Per ogni DbObject è possibile ottenere direttamente il sottostante DataSet e DataTable. E' possibile anche mappare una classe "POCO" con attributi che si chiamano come i singoli oggetti DbColumn ed ottenere una lista IList<"POCO">.

2.2 Sistemi RDBMS interfacciabili da BasicDAL

BasicDAL può interfacciare i seguenti sistemi RDBMS

```
Microsoft SQLServer
Microsoft OleDB generico
Oracle OracleDataAccess (versioni 9i -> 14g)
Microsoft ODBC generico
Microsoft ODBC specifico IBM Client Access AS/400
Microsoft OleDB specifico IBM Client Access AS/400
```

3 Uso di BasicDAL

L'assembly .NET è denominato BasicDAL.dll. E' un oggetto compilato in modalità AnyCPU e richiede la presenza del .NET Framework 4.0 o superiore.

Esistono diverse versioni dell'assembly in funzione del numero di Provider DB referenziati. La versione standard non contiene riferimenti al provider Oracle OracleDataAccess. La versione standard non richiede installazione di componenti specifici per l'accesso ai provider standard. Rimane la necessità di installare gli eventuali driver OleDB o ODBC. La versione estesa contiene i riferimenti a tutti i provider supportati e richiede quindi la loro installazione preventiva sul sistema, come nel caso del provider Oracle.

Per utilizzare le funzionalità di BasicDAL nei propri programmi è necessario referenziare l'assembly BasicDAL.dll nel proprio progetto.

3.1 Definizione delle entità DbObject

L'oggetto principale di interazione è l'oggetto DbObject. Esso mappa l'oggetto database in un oggetto applicativo basato sul pattern Active Record. Si tratta di una classe derivata da una classe base (DbObject) con proprietà predefinite e arricchita da un insieme di proprietà DbColumn e DbParameter che variano in funzione delle colonne e dei parametri degli oggetti database.

La definizione di un nuovo oggetto derivato da DbObject può essere di tipo "Early Binding" o "Late Binding". La definizione "Early Binding" permette di usare l'oggetto DbObject con le funzionalità di Intellisense che includono anche le proprietà di tipo DbColumn. E' tuttavia possibile usare una definizione "Late Binding" dove la collection DbColumns viene creata esternamente alla classe DbObject e collegata

successivamente. In tale modalità “Late Binding” si perde la funzione di Intellisense e la tipizzazione forte. E’ tuttavia importante notare che anche con una dichiarazione “Early Bindig” è possibile modificare a runtime TUTTE le proprietà degli oggetti DbColumn. Ciò rende possibile caricare le proprietà dei singoli oggetti DbColumn a Runtime da definizioni esterne al programma (ad esempio da XML o Json).

Es. “Early Binding” – Dichiarazione

```
public class db_comuni : BasicDAL.DbObject
{
    public BasicDAL.DbColumn progressivocomune = new BasicDAL.DbColumn("[progressivocomune]","Progressivo", DbType.Int32, true, 0);
    public BasicDAL.DbColumn siglaprovincia = new BasicDAL.DbColumn("[siglaprovincia]", "Sigla Provincia", DbType.String, false, "");
    public BasicDAL.DbColumn nomecomune = new BasicDAL.DbColumn("[nomecomune]","Nome Comune", DbType.String, false, "");
    public BasicDAL.DbColumn comlatgradi = new BasicDAL.DbColumn("[comLatGradi]", "Lat.Gradi",DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlatminuti = new BasicDAL.DbColumn("[comLatMinuti]", "Lat.Minuti",DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlatsecondi = new BasicDAL.DbColumn("[comLatSecondi]","Lat.Secondi", DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlatsecondidec = new BasicDAL.DbColumn("[comLatSecondiDec]","Lat.Secondi.Dec.", DbType.String, false, "");
    public BasicDAL.DbColumn comlonggradi = new BasicDAL.DbColumn("[comLongGradi]", "Long.Gradi",DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlongminuti = new BasicDAL.DbColumn("[comLongMinuti]","Long.Minuti", DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlongsecondi = new BasicDAL.DbColumn("[comLongSecondi]","Long.Secondi", DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlongsecondidec = new BasicDAL.DbColumn("[comLongSecondiDec]","Long.Secondi.Dec.", DbType.String, false, "");
    public BasicDAL.DbColumn prioritacomune = new BasicDAL.DbColumn("[prioritacomune]","Priorità", DbType.Byte, false, 0);

    public db_comuni()
    {
        this.DbObjectType = BasicDAL.DbObjectTypeEnum.Table;
        this.DbTableName = "comuni";
        this.FriendlyName = "Comuni";
        this.InterfaceMode = BasicDAL.InterfaceModeEnum.Public;
    }
}
```

Es. “Early Binding” – Uso con Intellisense

```
this.db_comuni.progressivocomune.Value = 10;
```

Es. “Late Binding”

```
this.db_comuni.GetDbColumn(“progressivocomune”).Value=10;
```

3.1.1 Il metodo costruttore dell’oggetto derivato da DbObject

Il codice nel metodo costruttore è essenziale e va definiti indipendentemente dal fatto che si usi una definizione delle DbColumns in modalità “Early Binding” o “Late Binding”.

```
    public db_comuni()
    {
        this.DbObjectType = BasicDAL.DbObjectTypeEnum.Table;
        this.DbTableName = "comuni";
        this.FriendlyName = "Comuni";
        this.InterfaceMode = BasicDAL.InterfaceModeEnum.Public;
    }
```

.*DbObjectType* indica la tipologica di oggetto database che viene mappato. Il valori sono enumerati in DbObjectTypeEnum. In sintessi essi possono essere Tabelle, Viste, Query SQL, Stored Procedure e Funzioni

.*DbTableName* indica il nome dell’oggetto database mappato.

.*FriendlyName* indica il nome “amichevole” con cui volete indicare nelle UI l’oggetto database.

.InterfaceMode indica alla libreria la modalità di dichiarazione dell’insieme di DbColumn (campi public, campi privati, proprietà ecc.). Permette alla libreria di ispezionare gli oggetti in funzione del metodo scelo per la loro creazione.

Queste sono le propriertà necessarie alla definizione valida di un oggetto DbObject. Le altre sono documentate nella sezione di documentazione dell’oggetto DbObject.

3.1.2 Interazione con i dati e gli elementi della UI.

La proprietà .DataBinding dell'oggetto DbObject determina il comportamento relativamente alle modalità di interazione fra programma e dati sottostanti il DbObject.

Ogni DbColumn ha la proprietà .Value che espone il valore letto dal database e che verrà eventualmente scritto.

```
public enum DataBoundControlsBehaviour
{
	WindowsFormsDataBinding = 2, // Usa il databind di Windows
	BasicDALDataBinding = 1, // Usa il databind native BasicDAL (E' il default)
	NoDataBinding = 0 // Non usa nessun databind se non la property Value
}
```

Quando è impostato il valore BasicDALDataBinding l'utente può definire, per ogni DBColumn una collection di BoundControl con cui associare delle proprietà di oggetti al valore .Value del DbColumn. In genere gli oggetti associati sono oggetti di UI.

```
this.db_comuni.DataBinding = BasicDAL.DataBoundControlsBehaviour.BasicDALDataBinding;
this.db_comuni.progressivocomune.BoundControls.Add(txtProgressivoComune);
this.db_comuni.nomecomune.BoundControls.Add(txtNomeComune);
this.db_comuni.progressivocomune.BoundControls.Add(txtProgressivoT, "text", BasicDAL.BindingBehaviour.Write);
this.db_comuni.nomecomune.BoundControls.Add(txtNomeComuneT, "text", BasicDAL.BindingBehaviour.Write);
this.db_comuni.prioritacomune.BoundControls.Add(txtPrioritaComune);
this.db_comuni.siglaprovincia.BoundControls.Add(cmbProvince, "selectedvalue");
```

L'enumerazione BasicDAL.BindingBehaviour contiene i valori che determinano la logica di Binding.

```
public enum BindingBehaviour
{
	ReadWrite = True,
	Write = False,
	ReadWriteAddNew = 2,
	WriteAddNew = 3
}
```

Read è la funzione di lettura del valore della proprietà dell'oggetto mappato nel BoundControl nella proprietà Value del DbColumn, sarebbe a dire da UI a Database.
Write è la funzione di scrittura del valore della proprietà Value del DbColum nella proprietà dell'oggetto mappato nel BoundControl, sarebbe a dire da Database a UI.

AddNew è equivalente a Read, limitatamente alla creazione di nuovi elementi.
Il valore di default di BindigBehaviour è ReadWriteAddNew.

3.2 Il collegamento del DbObject con l'oggetto Database: l'oggetto DbConfig.

Il DbObject definisce la struttura dell'oggetto database e le regole di mappatura ma non ha nessuna informazione di deve si trova il DB , del contesto di connessione e dell'eventuale contesto di transazione.

Questo compito è assolto dall'oggetto DbConfig. Per poter essere operativo un DbObject deve essere quindi associato ad un oggetto DbConfig.

```
public BasicDAL.DbConfig DbConfig_Italia= new BasicDAL.DbConfig();

DbConfig_Italia.Provider = BasicDAL.Providers.SqlServer;

DbConfig_Italia.ServerName = "mysqlserver";
DbConfig_Italia.DataBaseName = "DataBaseName";
DbConfig_Italia.UserName = "UserName";
DbConfig_Italia.Password = "Password";
DbConfig_Italia.AuthenticationMode = 0; //0 = autenticazione nativa database
DbConfig_Italia.Open();
```

3.3 Il caricamento dei dati nel DbObject

Una volta definito ed "aperto" il DbConfig è possibile inizializzare e popolare il DbObject usando il metodo .Init del DbObject nel seguente modo

```
this.db_comuni.Init(this.DbConfig);
this.db_comuni.LoadAll();
```

oppure

```
this.db_comuni.Init(this.DbConfig);
this.db_comuni.Open();
```

In questo modo si associa il DbObject al DbConfig e si caricano i dati nel DataTable sottostante il DbObject. Dato l'elevato numero di dati in termini di righe che l'esecuzione del comando di LoadAll/Open può implicare il numero massimo di righe (records) ritornato è impostabile per mezzo della proprietà .TopRecords del DbObject. Il default è 0 ovvero tutti i record ritornabili senza provocare una condizione di out-of-memory nel processo.

Di default non vengono applicati filtri nella query che popola i dati del DbObject. E' possibile passare insiemi di filtri usando le collezioni di oggetti BasicDAL.DbFilters e DbFiltersGroup del DbObject. Nell'esempio sottostante viene impostata una condizione di ricerca per i comuni aventi come sigla provincia il valore "TE" e dove nomecomune inizia per "C%"

```
BasicDAL.DbFilters DbFilters = new BasicDAL.DbFilters();
DbFilters.Add(db_comuni.siglaprovincia, BasicDAL.ComparisionOperator.Equal, "TE", BasicDAL.LogicOperator.AND);
DbFilters.Add(db_comuni.nomecomune , BasicDAL.ComparisionOperator.Like , "C%");
db_comuni.FiltersGroup.Clear();
db_comuni.FiltersGroup.Add(DbFilters); -- oppure -- db_comuni.FiltersGroup.AddFilters(DbFilters, BasicDAL.LogicOperator.AND );
```

E' possibile comporre insiemi di valutazione delle condizioni (livelli di parentesi) e mettere tali insiemi in relazione logica usando il metodo .AddFilters in luogo del metodo .Add, limitato al solo operatore logico di AND.

Gli operatori di comparazione supportati sono

```
public enum ComparisionOperator
{
        Equals = 0, //=
        Equal = 0, //=
        GreaterThan = 1, //>
        LessThan = 2, //<
        GreaterThanOrEqualTo = 3, //>=
        LessThanOrEqualTo = 4, //<=
        NotEqualTo = 5, //<>
        Between = 6,
        In = 7,
        Like = 8,
        NotLike = 9,
```

```
        Exists = 10,
        None = 11,
        IsNull = 12,
        IsNotNull = 13,
        NotIn = 14,
        NotBetween = 15
}
```

Gli operatori logici sono

```
public enum LogicOperator
{
        None = 0,
        AND = 1,
        OR = 2,
        NOT = 3
}
```

3.3.1 Iterazione sull'insieme di record.

L'iterazione avviene tramite i metodo .MoveFirst, .MoveLast, .MoveNext, MovePrevious, .MoveTo dell'oggetto DbObject. Tali metodi spostano il cursore costituente l'Active Record. E' possibile alternativamente ottenere l'intero insieme di records come DataTable ADO.NET o come lista di oggetti POCO.

3.4 Aggiornamento dei dati

Per modificare i dati nel Database sottostante il DbObject è disponibile il metodo .Update del DbObject. Il metodo Update opera su un record la volta mentre il metodo UpdateBatch opera su più record la volta. La differenza sta nel momento dell'invocazione del metodo di aggiornamento. Update viene applicato solo sull'active record (quindi al solo record su cui è posizionato il cursore) mentre UpdateBatch viene applicato su tutti i records modificati.

3.5 Inserimento di nuovi dati

Per eseguire l'inserimento di nuovi dati occorre invocare il metodo .AddNew del DbObject, valorizzare i dati tramite la proprietà .Value dei DbColumn o tramite i controlli bindati e quindi invocare il metodo .Update o UpdateBatch. Se nelle DbColumn è presente un colonna di tipo Identity viene sempre invocato il metodo Update in luogo del metodo UpdateBatch per ritornare immediatamente il valore dell'Identity.

3.6 Annullamento delle modifiche

Per riportare i dati al valore originario presente nel database, anche dopo aver modificato il relativo .Value del DbColum è possibile alternativamente usare:

- La proprietà .ShadowValue del DbColum, assegnando il suo valore alla proprietà .Value;
- Il metodo .UndoChanges del DbObject. Questo metodo è però limitato all'Active record corrente.

3.7 Cancellazione dei dati

La cancellazione avviene invocando i seguenti metodi del DbObject:

- .Delete – Elimina l'active record corrente;
- .DeleteAll – Elimina TUTTI i records contenuti nel DbObject

L'operazione di cancellazione è reversibile solo se effettuata all'interno di una Transazione, sarebbe a dire invocando PRIMA il metodo .BeginTransaction del DbConfig a cui il DbObject è associato. Se non si conosce il nome dell'oggetto esso è comunque disponibile nel DbObject tramite la proprietà .DbConfig del DbObject. Il commit avviene invocando il metodo .CommitTransaction mentre il Rollback tramite l'invocazione del metodo .RollBack.

3.8 Gli Eventi del DbObject

Il DbObject genera i seguenti eventi

```
BoundCompleted()
```

Viene generato quando è completato il binding dei controlli associati ai DbColumn.

```
DataEventBefore (DataEventType EventType)
```

Viene generato prima di eseguire un'operazione fra quelle elencate in DataEventType. E' possibile cancellare l'evento impostando la variabile Cancel a true.

```
DataEventAfter (DataEventType EventType)
```

Viene generato dopo l'esecuzione di un'operazione fra quelle elencate in DataEventType. E' possibile cancellare l'evento impostando la variabile Cancel a true.

```
WriteCSVRow (int rownum, ref boolean Cancel)
```

EventType è enumerato nel seguente modo.

```
public enum DataEventType
{
        Disposing = 0,
        Initializing = 1,
        AddNew = 2,
        Update = 3,
        Delete = 4,
        Query = 5,
        Insert = 6,
        UndoChanges = 9,
        Binding = 10,
        BindingFromDataReader = 11,
        ControlsBinding = 12,
        MoveFirst = 100,
        MovePrevious = 101,
        MoveLast = 102,
        MoveNext = 103,
        MoveTo = 104,
        AddToDataSet = 200,
        DeleteAll = 400,
        DeleteFromDataSet = 401,
        DeleteFromDataTable = 402,
        UpdateFromDataSet = 300,
        UpdateFromDataTable = 302
}
```

3.9 La validazione dei DbColumn

Ogni DbColumn possiede una regola di validazione impostabile tramite il metodo .SetValidationRule. Le regole di validazione sono enumerate in ValidationTypes

```
public enum ValidationTypes
{
        None = 0,  // nessuna regola
        Required = 1, // il dato deve essere presente
        RegularExpression = 2, //deve soddisfare una espressione regolare
        ValidURI = 6, // deve essere una URI correttamente formata
        Range = 3,  // deve essere compreso in un range
        CompareList = 4, // deve essere compreso in una lista di valori
        Equal = 5, // deve essere uguale a
        ValidURL = 7, // deve essere una URL correttamente formata
        ValidEmail = 8 // deve essere un indirizzo email correttamente formato
}
```

Le regole si impostano in funzione del tipo di regola. Di seguito alcuni esempio di regole

```
this.db_campeggi.numerostelle.SetValidationRule(BasicDAL.ValidationTypes.Range, 1, 5);
this.db_campeggi.emailproprieta.SetValidationRule(BasicDAL.ValidationTypes.ValidEmail, "Deve essere una email valida.");
```

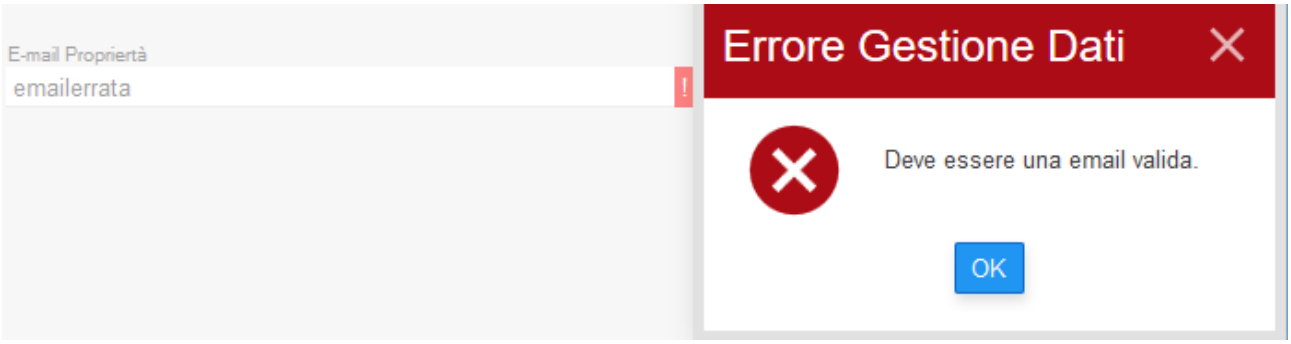
La loro validazione avvenire in diversi modi.

L'oggetto DbColumn possiede la funzione .Validate() che ritorna un booleano che indica se la validazione è stata superata. Se l'oggetto DbColumn ha dei BoundControl collegati è possibili invocare la funzione .ValidateBoundControls() che effettuerà la validazione del valore proprietà associate per i controlli che possono modificare la proprietà .Value del DbColumn (sarebbe a dire quelli con comportamento BindingBehaviour = Read e ReadAddNew.

invocando il metodo .EnsureValidationRules dell'oggetto DbObject.

```
this.db_campeggi.EnsureValidationRules ();
```

Per gli oggetti BoundControl in caso di mancata validazione viene visualizzato a fianco dell'oggetto un indicatore (rosso normalmente) che se oggetti di click con il mouse mostra il motivo della mancata validazione



3.10 La gestione degli errori e degli esiti delle operazioni in BasicDAL

Gli errori e gli esiti delle operazioni vengono gestiti in BasicDAL per mezzo di proprietà specifiche degli oggetti DbConfig e DbObject

Proprietà	Descrizione
.LastError	Stringa, contiene la descrizione dell'errore eventualmente generato dall'ultimo metodo/funzione invocato per l'oggetto. Ha valore nullo se non c'è stato errore.
.LastErrorCode	Intero, contiene il codice di errore eventualmente generato dall'ultimo metodo/funzione invocato per l'oggetto. Ha valore 0 se non c'è stato errore.
.LastErrorException	Exception .NET , contiene l'oggetto di Eccezione .NET eventualmente generato dall'ultimo metodo/funzione invocato per l'oggetto. Ha valore nullo se non c'è stato errore.
.ExecutionResult	L'oggetto ExecutionResult. L'uso di tale oggetto rimpiazza l'uso delle proprietà .LastError, mantenute per compatibilità con le versioni precedenti di BasicDAL.

Dopo ogni esecuzione di un metodo è possibile verificare il valore di queste proprietà e decidere come gestire l'eventuale errore.

3.10.1 Interazione con la situazione di errore.

BasicDAL cerca di gestire internamente le sue situazioni di errore ed eccezioni presentando nel caso una finestra di dialogo con l'indicazione dell'errore e delle sue cause. Tuttavia è possibile scegliere di gestire direttamente con proprio codice la situazione di errore senza che venga emessa una finestra di dialogo agendo sul valore della proprietà booleana .SuppressErrorsNotification del DbConfig o del DbObject. Il valore impostato dal DbConfig si propaga (se non definito) al DbObject associato al DbConfig.

. SuppressErrorsNotification = true; è utile quando l'applicazione è un servizio windows in modo da non interrompere con un dialogo modale l'esecuzione del codice.

Il meccanismo di interazione cambia a seconda della natura della UI utilizzata. E' possibile impostare in BasicDAL la tipologia di UI usando la proprietà .RuntimeUI degli oggetti DbConfig e DbObject e la proprietà RedirectErrorsNotificationTo `dell'oggetto DbConfig.`

```
public enum RuntimeUI
{
        Service = 0,
        WindowsForms = 1,
        Wisej = 2,
        Web = 3,
        WPF = 4
}
```

RedirectErrorsNotificationTo indica un oggetto che implementa il metodo .Show, come ad esempio

```
public class BasicDALMessageBox
{
        public void Show(string Text, string Caption = "Errore Gestione Dati")
        {
                MessageBox.Show(Text, Caption, MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
}
```

Impostando la proprietà RedirectErrorsNotificationTo nel seguente modo si ottiene l'uso del dialogo specifico della UI prescelta.

```
DbConfig_Italia.RedirectErrorsNotificationTo = BasicDALMessageBox;
```

4 I Controlli BasicDAL

5 BoundControl

```
[Serializable()]
```

```
public class BoundControl
```

L'Oggetto BoundControl lega un oggetto DbColumn ad una proprietà di oggetto di interfaccia utente (sia esso Windows Forms, Windows Presentation Foundation , Web Forms o Wisej).

Ogni oggetto DbObject possiede una collection vuota di BoundControls (vedi) che viene inizializzata contestualmente alla sua creazione. Il binding fra valore del DbColumn e della property del BoundControl viene eseguito solo se la proprietà "`.DataBinding`" `del DbObject è uguale a BasicDAL.DataBoundControlsBehaviour.BasicDALDataBinding`

5.1 Costruttori

	Name	Description
	BoundControl()	
	BoundControl(object control, string propertyname, BindingBehaviour)	Control è l'oggetto di UI del quale si vuole legare la proprietà indicata nel parametro propertyname con il comportamento indicato nell'enumerazione BindingBehavior (vedi Enumerazione BindingBehaviour)

5.2 Metodi

	Name	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	ToString	Returns a string that represents the current object. (inherited from Object).

6 BoundControls

```
[Serializable()]
```

```
public class BoundControls : CollectionBase
```

Ogni oggetto DbObject possiede una collection vuota di BoundControls (vedi) che viene inizializzata contestualmente alla sua creazione. Il binding fra valore del DbColumn e della property del BoundControl viene eseguito solo se la proprietà ".DataBinding" del DbObject è uguale a BasicDAL.DataBoundControlsBehaviour.BasicDALDataBinding

6.1 Proprietà

	Name	Description
	Capacity	Gets or sets the number of elements that the CollectionBase can contain. (inherited from CollectionBase).
	Count	Gets the number of elements contained in the CollectionBase instance. This property cannot be overridden. (inherited from CollectionBase).

6.2 Metodi

	Name	Description
	Add(object control, string propertyname, BindingBehaviour BindingBehaviour)	Control è l'oggetto di UI del quale si vuole legare la proprietà indicata nel parametro propertyname con il comportamento indicato nell'enumerazione BindingBehavior (vedi Enumerazione BindingBehaviour)
	Add(object control, string propertyname)	Control è l'oggetto di UI del quale si vuole legare la proprietà indicata nel parametro propertyname con il comportamento indicato nell'enumerazione BindingBehavior (vedi Enumerazione BindingBehaviour)
	Add(int, object, string, BindingBehaviour)	Control è l'oggetto di UI del quale si vuole legare la proprietà indicata nel parametro propertyname con il comportamento indicato nell'enumerazione BindingBehavior (vedi Enumerazione BindingBehaviour)
	Clear	Removes all objects from the CollectionBase instance. This method cannot be overridden. (inherited from CollectionBase).

	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetEnumerator	Returns an enumerator that iterates through the CollectionBase instance. (inherited from CollectionBase).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	OnClear	Performs additional custom processes when clearing the contents of the CollectionBase instance. (inherited from CollectionBase).
	OnClearComplete	Performs additional custom processes after clearing the contents of the CollectionBase instance. (inherited from CollectionBase).
	OnInsert(Int32, Object)	Performs additional custom processes before inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnInsertComplete(Int32, Object)	Performs additional custom processes after inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnRemove(Int32, Object)	Performs additional custom processes when removing an element from the CollectionBase instance. (inherited from CollectionBase).
	OnRemoveComplete(Int32, Object)	Performs additional custom processes after removing an element from the CollectionBase instance. (inherited from CollectionBase).

	OnSet(Int32, Object, Object)	Performs additional custom processes before setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnSetComplete(Int32, Object, Object)	Performs additional custom processes after setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnValidate(Object)	Performs additional custom processes when validating a value. (inherited from CollectionBase).
	RemoveAt(Int32)	Removes the element at the specified index of the CollectionBase instance. This method is not overridable. (inherited from CollectionBase).
	ToString	Returns a string that represents the current object. (inherited from Object).

7 BusinessObjectAdapter(Of T)

```
public class BusinessObjectAdapter<T>
```

L'oggetto BusinessObjectAdapter<T> permette di convertire l'insieme di dati sottostanti il DbObject in una lista di oggetti di una arbitraria classe POCO e viceversa. La mappatura fra i due oggetti (DbObject e oggetto POCO di classe <T>) avviene fra gli oggetti DbColumn presenti nella collection DbColumns del DbObject aventi lo stesso nome delle proprietà nella classe POCO.

7.1 Metodi

	Name	Description
	FromDataTableToList(DataTable)	Converte il DataTable sottostante il DbObject in una lista di tipo <T>
	FromDbObjectToList(DbObject)	Converte i dati sottostanti il DbObject in una lista di tipo <T>
	FromListToDbObject(List<T>, ref DbObject)	Converte una lista di oggetti di tipo <T> in dati sottostanti l'oggetti DbObject
	ToList(DbObject)	Converte i dati sottostanti il DbObject in una lista di oggetti di classe <T>. Equivalente a FromDbObjectToList(DbObject)

8 ConnectedDbLookUps

```
public class ConnectedDbLookUps : CollectionBase
```

8.1 Proprietà

	Name	Description
	Capacity	Gets or sets the number of elements that the CollectionBase can contain. (inherited from CollectionBase).
	Count	Gets the number of elements contained in the CollectionBase instance. This property cannot be overridden. (inherited from CollectionBase).

8.2 Metodi

	Name	Description
	Add(DbLookUp)	
	Clear	Removes all objects from the CollectionBase instance. This method cannot be overridden. (inherited from CollectionBase).
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetEnumerator	Returns an enumerator that iterates through the CollectionBase instance. (inherited from CollectionBase).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	OnClear	Performs additional custom processes when clearing the contents of the CollectionBase instance. (inherited from CollectionBase).

	OnClearComplete	Performs additional custom processes after clearing the contents of the CollectionBase instance. (inherited from CollectionBase).
	OnInsert(Int32, Object)	Performs additional custom processes before inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnInsertComplete(Int32, Object)	Performs additional custom processes after inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnRemove(Int32, Object)	Performs additional custom processes when removing an element from the CollectionBase instance. (inherited from CollectionBase).
	OnRemoveComplete(Int32, Object)	Performs additional custom processes after removing an element from the CollectionBase instance. (inherited from CollectionBase).
	OnSet(Int32, Object, Object)	Performs additional custom processes before setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnSetComplete(Int32, Object, Object)	Performs additional custom processes after setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnValidate(Object)	Performs additional custom processes when validating a value. (inherited from CollectionBase).
	RemoveAt(Int32)	Removes the element at the specified index of the CollectionBase instance. This method is not overridable. (inherited from CollectionBase).
	ToString	Returns a string that represents the current object. (inherited from Object).

9 DbACE Class

```
[Serializable()]
```

```
public class DbACE
```

9.1 Metodi

	Name	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	ToString	Returns a string that represents the current object. (inherited from Object).

9.2 Campi

	Name	Description
	ACEApplyTo	
	ACL	
	AccessMask	
	User	

10 DbACL Class

```
[Serializable()]
```

```
public class DbACL : SerializableDictionary<string, DbACE>
```

10.1 Properties

	Name	Description
	Comparer	Gets the IEqualityComparer<T> that is used to determine equality of keys for the dictionary. (inherited from Dictionary<TKey, TValue>).
	Count	Gets the number of key/value pairs contained in the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).
	Item(TKey)	Gets or sets the value associated with the specified key. (inherited from Dictionary<TKey, TValue>).
	Keys	Gets a collection containing the keys in the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).
	Values	Gets a collection containing the values in the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).

10.2 Methods

	Name	Description
	Add(string, ACEAccessMask)	
	Clear	Removes all keys and values from the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).
	ContainsKey(TKey)	Determines whether the Dictionary<TKey, TValue> contains the specified key. (inherited from Dictionary<TKey, TValue>).
	ContainsValue(TValue)	Determines whether the Dictionary<TKey, TValue> contains a specific value. (inherited from Dictionary<TKey, TValue>).
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).

	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetEnumerator	Returns an enumerator that iterates through the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetObjectData(SerializationInfo, StreamingContext)	Implements the ISerializable interface and returns the data needed to serialize the Dictionary<TKey, TValue> instance. (inherited from Dictionary<TKey, TValue>).
	GetSchema()	This method is reserved and should not be used. When implementing the IXmlSerializable interface, you should return null (Nothing in Visual Basic) from this method, and instead, if specifying a custom schema is required, apply the XmlSchemaProviderAttribute to the class. (inherited from SerializableDictionary(Of TKey, TValue)).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	OnDeserialization(Object)	Implements the ISerializable interface and raises the deserialization event when the deserialization is complete. (inherited from Dictionary<TKey, TValue>).
	ReadXml(XmlReader)	Generates an object from its XML representation. (inherited from SerializableDictionary(Of TKey, TValue)).
	Remove(TKey)	Removes the value with the specified key from the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).
	ToString	Returns a string that represents the current object. (inherited from Object).

	TryGetValue(TKey,)	(inherited from Dictionary<TKey, TValue>).
	WriteXml(XmlWriter)	Converts an object into its XML representation. (inherited from SerializableDictionary(Of TKey, TValue)).

11 Oggetto DbColumn

L'oggetto DbColumn rappresenta la colonna di una tabella/vista/recordset, sarebbe a dire dell'insieme di dati sottostante un DbObject. Ogni DbObject possiede una collection di DbColumn denominata DbColumns.

11.1 Costruttori

	Name	Description
	DbColumn()	
	DbColumn(string, string, DbType, bool, int, bool, object, object)	
	DbColumn(string, string)	
	DbColumn(string, DbType, bool, object, int)	
	DbColumn(string, string, DbType, bool, object, int)	

11.2 Proprietà

	Name	Description
	Value	Imposta o restituisce il valore della colonna. Imposta sempre il valore nell'oggetto DataRow nel DataTable ADO.NET associato al DbObject.
	Value(bool)	Imposta o restituisce il valore della colonna impostando o meno il valore nell'oggetto DataRow nel DataTable ADO.NET associato al DbObject.
	ValueNoUpdateCurrentDataRow	Imposta o restituisce il valore della colonna. Non imposta il valore nell'oggetto DataRow nel DataTable ADO.NET associato al DbObject.

11.3 Metodi

	Name	Description

	`public bool CheckForDataChange(object OnlyBoundControls = false)`	Restituisce in valore booleano che indica una modifica dei dati avvenuta dopo il loro caricamento dalla sorgente dati. Il paramento OnlyBoundControls limita la modifica ai soli oggetti di UI collegati al DbColum.
	`public void DataBindings(ref object Control, string PropertyName, DataTable DataTable, bool Clear)`	Effettua il databing ad un controllo Control quando la modalità Databing è WindowsDataBinding.
	`public void DataBindings(ref object Control, string PropertyName, DataTable DataTable)`	Effettua il databing ad un controllo Control quando la modalità Databing è WindowsDataBinding.
	`public void DataBindingsClear(ref object Control)`	Azzera il databindig del controllo Control quando la modalità Databing è WindowsDataBinding.
	GetMaxValue()	Ottiene il valore massimo di quel DbColumn
	IsBoolean()	Indica che il tipo di dato è booleano.
	IsChanged()	Indica che il valore del DbColumn è cambiato da quando è stato letto dal DB sottostante.
	IsDate()	Indica che il tipo di dato è Date
	IsEdited()	Indica che almeno un BoundControl associato è stato editato dall'utente.
	IsNumeric()	Indica che il tipo di dato è numerico.

	IsString()	Indica che il tipo di dato è alfanumerico.
	IsTime()	Indica che il tipo di dato è Time
	IsXML()	Indica che il tipo di dato è XML
	`public void SetValidationRule(BasicDAL.ValidationTypes ValidationType, object MinValue, object MaxValue, string ValidationMessage = "")`	Imposta una regola di validazione per il DbColumn
	Validate()	Esegue tutte le regole di validazione definite per il DbColumn restituendo True se superate.
	ValidateBoundControls()	Esegue tutte le regole di validazione definite per il DbColumn restituendo True se superate, limitandosi ai controlli BoundControl

11.4 Estensioni

	Name	Description
	IsChanged(this DbColumn)	(Defined by BasicDalSharedCode).

11.5 Campi

	Name	Description
	ExecutionResult	Contiene l'esito dei metodi propri dell'oggetto DbColumn.

12 DbColumns Class

```
[Serializable()]
```

```
public class DbColumns : CollectionBase
```

La collection DbColumns contiene l'insieme di colonne presenti nell'oggetto Database mappato su DbObject. La collection è creata vuota al momento dell'invocazione del metodo costruttore del DbObject e viene riempita automaticamente con le occorrenze degli oggetti di DbColumn declinati come proprietà nella definizione dell'istanza del DbObject. Questo tipo di definizione è di tipo "Early Binding". La definizione "Early Binding" permette di usare l'oggetto DbObject con le funzionalità di Intellisense che includono anche le proprietà di tipo DbColumn. E' tuttavia possibile usare una definizione "Late Binding" dove la collection DbColumns viene creata esternamente alla classe DbObject e collegata successivamente. In tale modalità "Late Binding" si perde la funzione di Intellisense e la tipizzazione forte.
E' tuttavia importante notare che anche con una dichiarazione "Early Bindig" è possibile modificare a runtime TUTTE le proprietà degli oggetti DbColumn. Ciò rende possibile caricare le proprietà dei singoli oggetti DbColumn a Runtime da definizioni esterne al programma (ad esempio da XML o Json).

Es. "Early Binding" – Dichiarazione

```
public class db_comuni : BasicDAL.DbObject
{
    public BasicDAL.DbColumn progressivocomune = new BasicDAL.DbColumn("[progressivocomune]","Progressivo", DbType.Int32, true, 0);
    public BasicDAL.DbColumn siglaprovincia = new BasicDAL.DbColumn("[siglaprovincia]", "Sigla Provincia", DbType.String, false, "");
    public BasicDAL.DbColumn nomecomune = new BasicDAL.DbColumn("[nomecomune]","Nome Comune", DbType.String, false, "");
    public BasicDAL.DbColumn comlatgradi = new BasicDAL.DbColumn("[comLatGradi]", "Lat.Gradi",DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlatminuti = new BasicDAL.DbColumn("[comLatMinuti]", "Lat.Minuti",DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlatsecondi = new BasicDAL.DbColumn("[comLatSecondi]","Lat.Secondi", DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlatsecondidec = new BasicDAL.DbColumn("[comLatSecondiDec]","Lat.Secondi.Dec.", DbType.String, false, "");
    public BasicDAL.DbColumn comlonggradi = new BasicDAL.DbColumn("[comLongGradi]", "Long.Gradi",DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlongminuti = new BasicDAL.DbColumn("[comLongMinuti]","Long.Minuti", DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlongsecondi = new BasicDAL.DbColumn("[comLongSecondi]","Long.Secondi", DbType.Byte, false, 0);
    public BasicDAL.DbColumn comlongsecondidec = new BasicDAL.DbColumn("[comLongSecondiDec]","Long.Secondi.Dec.", DbType.String, false, "");
    public BasicDAL.DbColumn prioritacomune = new BasicDAL.DbColumn("[prioritacomune]","Priorità", DbType.Byte, false, 0);

    public db_comuni()
    {
        this.DbObjectType = BasicDAL.DbObjectTypeEnum.Table;
        this.DbTableName = "comuni";
        this.FriendlyName = "Comuni";
        this.InterfaceMode = BasicDAL.InterfaceModeEnum.Public;
    }
}
```

Es. "Early Binding" – Uso con Intellisense

```
this.db_comuni.progressivocomune.Value = 10;
```

Es. "Late Binding"

```
this.db_comuni.GetDbColumn("progressivocomune").Value=10;
```

12.1 Costruttori

	Name	Description
	DbColumns()	

12.2 Proprietà

	Name	Description
	Capacity	Gets or sets the number of elements that the CollectionBase can contain. (inherited from CollectionBase).
	Count	Gets the number of elements contained in the CollectionBase instance. This property cannot be overridden. (inherited from CollectionBase).

12.3 Metodi

	Name	Description
	Add(DbColumn)	Aggiunge un DbColumn alla collection
	Add(DbColumn, object, string)	
	Clear	Removes all objects from the CollectionBase instance. This method cannot be overridden. (inherited from CollectionBase).
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetEnumerator	Returns an enumerator that iterates through the CollectionBase instance. (inherited from CollectionBase).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	OnClear	Performs additional custom processes when clearing the contents of the CollectionBase instance. (inherited from CollectionBase).

	OnClearComplete	Performs additional custom processes after clearing the contents of the CollectionBase instance. (inherited from CollectionBase).
	OnInsert(Int32, Object)	Performs additional custom processes before inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnInsertComplete(Int32, Object)	Performs additional custom processes after inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnRemove(Int32, Object)	Performs additional custom processes when removing an element from the CollectionBase instance. (inherited from CollectionBase).
	OnRemoveComplete(Int32, Object)	Performs additional custom processes after removing an element from the CollectionBase instance. (inherited from CollectionBase).
	OnSet(Int32, Object, Object)	Performs additional custom processes before setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnSetComplete(Int32, Object, Object)	Performs additional custom processes after setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnValidate(Object)	Performs additional custom processes when validating a value. (inherited from CollectionBase).
	RemoveAt(Int32)	Removes the element at the specified index of the CollectionBase instance. This method is not overridable. (inherited from CollectionBase).
	ToString	Returns a string that represents the current object. (inherited from Object).

12.4 Estensioni

	Name	Description
	AsParallel()	Enables parallelization of a query. (Defined by ParallelEnumerable).
	AsQueryable()	Converts an IEnumerable to an IQueryable. (Defined by Queryable).

	Cast<TResult>()	Casts the elements of an IEnumerable to the specified type. (Defined by Enumerable).
	OfType<TResult>()	Filters the elements of an IEnumerable based on a specified type. (Defined by Enumerable).

13 L'Oggetto DbConfig

L'oggetto DbConfig contiene la configurazione in termini di parametri di connessione verso un sistema di database, lo stato della connessione ed il contesto di transazione della connessione. Nella sostanza esso ospita gli oggetti Connection e Transaction di ADO.NET. Ogni oggetto DbObject deve essere connesso da un DbConfig inizializzato per poter compiere una qualsiasi azione di lettura/scrittura.

13.1 Campi

	Name	Description
	DbTransaction	Contiene il contesto di transazione del DbConfig. E' a tutti gli effetti l'Oggetto Transaction di ADO.NET.
	ExecutionResult	Contiene l'esito dell'esecuzione dell'ultimo metodo invocato.
	ManagedDbObjects	E' la collection degli oggetti DbObject associati al DbConfig.
	ODBCConnectionStringBase1	Template di base con autenticazione integrata Windows di stringa di connessione ODBC
	ODBCConnectionStringBase2	Template di base con autenticazione nativa di stringa di connessione ODBC
	OracleConnectionStringBase1	Template di base con autenticazione integrata Windows di stringa di connessione ORACLE
	OracleConnectionStringBase2	Template di base con autenticazione nativa di stringa di connessione ORACLE
	RuntimeUI	Valore del tipo di UI a runtime. Enumerazione RuntimeUI
	SQLServerConnectionStringBase1	Template di base con autenticazione integrata Windows di stringa di connessione SQLServer
	SQLServerConnectionStringBase2	Template di base con autenticazione nativa di stringa di connessione SQLServer
	ValidatorLabelControl	E' l'oggetto usato per creare gli indicatori di fallita validazione legati che verranno usati dagli oggetti associati al DbConfig. Varia in funzione della tipologia di UI.

13.2 Proprietà

	Name	Description
	ACL	Contiene l'insieme delle ACE (Access Control Elements) applicabili al DbConfig. Si propagano (se non definite diversamente) ai DbObject associati al DbConfig.
	DbProviderFactory	E' l'oggetto ADO.NET DbProviderFactory
	ErrorState	Intero. Lo stato di errore
	HandleErrors	Booleano. Indica che gli errori generati internamente dal DbConfig debbono essere gestiti internamente o propagati al programma che istanzia DbConfig.
	LastError	Il testo dell''ultimo errore.
	LastErrorCode	Il codice numerico dell'ultimo errore
	LastErrorException	L'oggetto .NET Exception generato dall'ultimo errore/eccezione.
	LogErrors	Booleano. Indica la modalità di log delle condizioni di errore
	LogFile	Il nome del file che il DbConfig userà come file di log.
	NullDateValue	Stringa, il valore convenzionale che indica una data nulla. Default="01/01/0001 00:00:00"
	OnlyEntityInitialization	Booleano. Si propaga agli oggetti DbObject associati al DbConfig. In caso il valore sia true i metodi .Init ed .Open del DbObject non effettuano il controllo di coerenza dello schema fra elementi DbColumn e sottostante oggetto Database. Permette di aumentare la velocità di inizializzazione del DbObject ma può generare degli errori in caso di non coerenza dello schema.
	RedirectErrorsNotificationTo	E' l'oggetto di UI che verrà invocato per mostrare la finestra di dialogo degli errori interni di BasicDAL quando la proprietà SuppressErrorNotification è true. Si propaga a tutti i DbObject associati al DbConfig.
	SuppressErrorsNotification	Booleano. Se true indica che non viene mostrata nessuna finestra di dialogo in caso di errore interno BasicDAL.

13.3 Metodi

	Name	Description
	BeginTransaction(IsolationLevel)	Inizia una transazione che coinvolge tutti i DbObject associati al DbConfig.
	CaseSensitiveQuery(bool)	Booleano. Applicabile ai provider che usano comparatori case sensitive (Es. ORACLE). Se false la query sarà case insensitive.
	Clone()	Clona il DbConfig.
	Close()	Chiude connessione e contesto di transazione.
	CommitTransaction()	Esegue il Commit della Transazione.
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetConnectionString()	Ottiene la stringa di connessione ADO.NET usata per connettere la sorgente dati.
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetOracleVersion()	Ottiene la versione di Oracle.
	GetType	Gets the Type of the current instance. (inherited from Object).
	LoadACLFromXMLFile(string)	Carica le ACL da un file in formato XML
	LoadACLFromXMLString(string)	Carica le ACL da una stringa in formato XML
	LoadFromConnectionString(string, Providers)	
	ManagedDbObjectsInit()	
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).

	Open(string, bool)	
	Open(bool)	
	Oracle_CaseSensitiveQuery(bool)	Booleano. Specifico per Oracle. Se false la query sarà case insensitive.
	Oracle_NLS_LENGTH_SEMANTICS(int)	Indica il tipo dato base usato per rappresentare un carattere in Oracle : 0 = Char 1 = Byte
	Ping(int)	Effettua il test di connessione verso il database. Il parametro è il numero di secondi di timeout.
	ResetError()	Azzera la situazione di errore interno.
	RollBackTransaction()	Esegue il RollBack della transazione.
	ToString	Returns a string that represents the current object. (inherited from Object).
	TransactionPending()	Boolean: indica che c'è una transazione in corso.

13.4 Creazione dell'oggetto DbConfig

```
public BasicDAL.DbConfig DbConfig_Italia= new BasicDAL.DbConfig();

DbConfig_Italia.Provider = BasicDAL.Providers.SqlServer;
DbConfig_Italia.ServerName = "NOMEDELSERVER";
DbConfig_Italia.DataBaseName = "NOMEDELDB";
DbConfig_Italia.UserName = "UserName";
DbConfig_Italia.Password = "Password";
DbConfig_Italia.AuthenticationMode = 0 // 0=nativa DB  1=Integrata OS
DbConfig_Italia.Provider = BasicDAL.Providers.SqlServer;

DbConfig.Init(); //oppure DbConfig.Open();
```

14 DbFilter Class

```
public class DbFilter
```

14.1 Metodi

	Name	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	ToString	Returns a string that represents the current object. (inherited from Object).

15 DbFilters Class

```
public class DbFilters : CollectionBase
```

15.1 Proprietà

	Name	Description
	Capacity	Gets or sets the number of elements that the CollectionBase can contain. (inherited from CollectionBase).
	Count	Gets the number of elements contained in the CollectionBase instance. This property cannot be overridden. (inherited from CollectionBase).

15.2 Metodi

	Name	Description
	Add(DbColumn, ComparisionOperator, object, LogicOperator)	Aggiunge un elemento DbFilter creandolo con i parametri definiti nel metodo.
	Add(DbFilter)	Aggiunge un elemento DbFilter
	AddBoundControl(DbColumn, ComparisionOperator, object, string, LogicOperator)	Associa un BoundControl ed un filtro DbFilter mappando il valore di un DbColumn ad una proprietà del BoundControl.

	Clear	Removes all objects from the CollectionBase instance. This method cannot be overridden. (inherited from CollectionBase).
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetEnumerator	Returns an enumerator that iterates through the CollectionBase instance. (inherited from CollectionBase).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	OnClear	Performs additional custom processes when clearing the contents of the CollectionBase instance. (inherited from CollectionBase).
	OnClearComplete	Performs additional custom processes after clearing the contents of the CollectionBase instance. (inherited from CollectionBase).
	OnInsert(Int32, Object)	Performs additional custom processes before inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnInsertComplete(Int32, Object)	Performs additional custom processes after inserting a new element into the CollectionBase instance. (inherited from CollectionBase).

	OnRemove(Int32, Object)	Performs additional custom processes when removing an element from the CollectionBase instance. (inherited from CollectionBase).
	OnRemoveComplete(Int32, Object)	Performs additional custom processes after removing an element from the CollectionBase instance. (inherited from CollectionBase).
	OnSet(Int32, Object, Object)	Performs additional custom processes before setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnSetComplete(Int32, Object, Object)	Performs additional custom processes after setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnValidate(Object)	Performs additional custom processes when validating a value. (inherited from CollectionBase).
	RemoveAt(Int32)	Removes the element at the specified index of the CollectionBase instance. This method is not overridable. (inherited from CollectionBase).
	ToString	Returns a string that represents the current object. (inherited from Object).

15.3 Estensioni

	Name	Description
	AsParallel()	Enables parallelization of a query. (Defined by ParallelEnumerable).
	AsQueryable()	Converts an IEnumerable to an IQueryable. (Defined by Queryable).
	Cast<TResult>()	Casts the elements of an IEnumerable to the specified type. (Defined by Enumerable).
	OfType<TResult>()	Filters the elements of an IEnumerable based on a specified type. (Defined by Enumerable).

15.4 Campi

	Name	Description
	ExecutionResult	L'esito dell'ultima operazione.

16 L'oggetto DbObject

16.1 Propriertà

	Name	Description
	ACL	L'insieme delle regole di accesso (ACE) applicabili all'oggetto DbObject. Se non definito deriva quelle del DbConfig.
	BOF	Booleano. Indica che il cursore è al primo record.
	Command	E' l'oggetto ADO.NET Command associato al DbObject per le operazioni di Select
	DataChanged	Booleano. Indica che i dati sono stati modificati dal programma o dall'utente.
	DbProviderFactory	E' l'oggetto ADO.NET ProviderFactory
	DbTransaction	E' l'oggetto ADO.NET Transaction un cui si trova il DbObject
	DeleteCommand	E' l'oggetto ADO.NET Command associato al DbObject per le operazioni di Delete
	EOF	Booleano. Indica che il cursore è all'ultimo record.
	ExecuteCommand	E' l'oggetto ADO.NET Command associato al DbObject per le operazioni di Esecuzione
	FriendlyName	Stringa. E' il nome amichevole del DbObject da usare eventualmente nella UI.
	HandleErrors	Booleano. Se true il DbObject gestisce direttamente il dialogo con l'utente in caso di errore.
	InsertCommand	E' l'oggetto ADO.NET Command associato al DbObject per le operazioni di Insert.
	LastError	Stringa. Il testo dell'errore eventualmente generato dall'ultima chiamata.
	LastErrorCode	Intero. Il codice dell'errore eventualmente generato dall'ultima chiamata.
	LastErrorComplete	Stringa. Il testo dell'errore esteso eventualmente generato dall'ultima chiamata.

	LastErrorException	Eccezione .NET eventualmente generato dall'ultima chiamata.
	LogErrors	Booleano. Se true effettua il log delle operazioni.
	LogFile	Il file di Log
	NullDateValue	Il valore convenzionale per indicare una data nulla.
	RowCount	E' il conteggio dei record sottostanti il DbObject.
	SQLUpdateCache	Booleano. Indica che i comandi di Update non vengono rivalutati ad ogni esecuzione.
	StoreProcedureParameters	E' la collection dei parametri definiti per un DbObject che mappa una StoredProcedure.
	UpdateCommand	E' l'oggetto ADO.NET Command associato al DbObject per le operazioni di Update.

16.2 Metodi

	Name	Description
	AddNew(DataBoundControlsBehaviour, bool)	Inizia l'inserimento di un nuovo record.
	AddNew(bool)	Inizia l'inserimento di un nuovo record.
	AddStoredProcedureParameter(string, DbType, object)	Aggiunge un parametro alla collection dei parametri di una stored procedure assegnando l'eventuale valore.
	AddtoDataSet(ref DataSet, string, CommandType, ConnectionState)	
	BlankDbColumns(DataBoundControlsBehaviour, bool)	Azzera la proprietà .Value di tutti i DbColumn del DbObject.
	BuildSQLDelete(ref DbParameterCollection)	Genera il comando DELETE
	BuildSQLInsert(ref DbParameterCollection)	Genera il comando INSERT
	BuildSQLJoin()	Valuta la query SQL di Join
	CaseSensitiveQuery(bool)	La query SQL è case sensitive.

CheckBoundControlsForExistingFKeys(string, string)	Verifica che i valori dei BoundControl dei DbColumn aventi un lookup verso altri DbObject (chiavi esterne) esistano o meno in questi DbObjetc collegati.
CheckBoundControlsForExistingPKey()	Esegue una query nell'oggetto database mappato per verificare che esista una chiave primaria simile a quella presente nei BoundControl dei DbColumn. Utile per determinare preventivamente errori di chiave primaria violata.
CheckForDataChange()	Booleano. Verifica se almeno il .Value di un DbColumn del DbObject è stato cambiato.
CheckForExistingPKeys(ConnectionState)	Esegue una query nell'oggetto database mappato per verificare che esista una chiave primaria simile a quella presente nell'active record (record corrente). Utile per determinare preventivamente errori di chiave primaria violata.
ClearAllBoundControls()	Azzera i valori di tutti i controlli bindati
ClearData(bool)	Azzera i dati sottostanti il DbObject.
CreateDbParameter(string, DbType, ParameterDirection, int)	Crea un parametro associato ad uno statement SQL
DataBindingClear()	Azzere il databinding standard di Windows Forms
DataReaderGetValues(DbDataReader)	Ottiene i valori dell'active record usando un DataReader in luogo di un DataTable.
DataRowGetValues(DataRow)	Ottiene i valori dell'active record usando la riga corrente del DataTable sottostante il DbObject.

DataRowGetValues(DataRow, DataBoundControlsBehaviour)	Ottiene i valori dell'active record usando la riga corrente del DataTable sottostante il DbObject.
Delete()	Elimina dell'active record dal DbObject e dal database sottostante.
Delete(DataBoundControlsBehaviour)	Elimina dell'active record dal DbObject e dal database sottostante.
DeleteAll(object)	Elimina tutti i dati dal DbObject e dal database sottostante.
DeleteAllFromFilters()	Non implementato.
DeleteFromDataSet(ref DataSet)	Elimina i dati nel database sottostante il DbObject a partire dalle modifice apportate ad un dataset
DeleteFromDataTable()	Elimina i dati nel database sottostante il DbObject a partire dalle modifiche apportate al datatable sottostante il DbObject.
DeleteFromDataTable(ref DataTable)	Elimina i dati nel database sottostante il DbObject a partire da un dataset
DisableBoundControls()	Disabilita tutti i controlli associati ai DbColumn del DbObject.
Dispose()	Esegue il dispose del DbObject
Dispose1()	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
DoQuery()	Esegue il caricamento dei dati del DbObject. E' un alias del metodo LoadAll
DumpCommandParameters()	Ottiene il dump dei parametri usati dal Command di esecuzione Query

DumpDeleteCommandParameters()	Ottiene il dump dei parametri usati dal Command di Delete
DumpExecuteCommandParameters()	Ottiene il dump dei parametri usati dal Command di esecuzione comandi
DumpInsertCommandParameters()	Ottiene il dump dei parametri usati dal Command di Insert
DumpRow()	Ottiene il dump dell'active record corrente.
DumpStoredProcedureCommandParameters()	Ottiene il dump dei parametri usati dal Command di usato dall'invocazione delle Stored Procedure
DumpUpdateCommandParameters()	Ottiene il dump dei parametri usati dal Command di Update
EnableBoundControls()	Abilita i controlli associati ai DbColumn del DbObject.
EnsureValidationRules()	Verifica le regole di validazione.
Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
ExecuteDataSet()	Esegue una query e carica i sottostanti DataSet e DataTable ADO.NET
ExecuteDataSet(string)	Esegue una query e carica i sottostanti DataSet e DataTable ADO.NET
ExecuteDataSet(string, CommandType)	Esegue una query e carica i sottostanti DataSet e DataTable ADO.NET
ExecuteDataSet(string, CommandType, ConnectionState)	Esegue una query e carica i sottostanti DataSet e DataTable ADO.NET
ExecuteDataSet(string, ConnectionState)	Esegue una query e carica i sottostanti DataSet e DataTable ADO.NET

ExecuteFunction()	Esegue una funzione definita nel Database ed associata al DbObject
ExecuteNonQuery(string)	Esegue un comando non query
ExecuteNonQuery(string, CommandType)	Esegue un comando non query
ExecuteNonQuery(string, CommandType, ConnectionState)	Esegue un comando non query
ExecuteNonQuery(string, ConnectionState)	Esegue un comando non query
ExecuteProcedure()	Esegue una Stored Procedure definita ne database ed associata al DbObject
ExecuteReader()	Esegue un DataReader per caricare i dati sottostanti il DbObject. Il set di record è sempre costituito da un singolo record con cursore solo in lettura in avanti.
ExecuteReader(string)	Esegue un DataReader in luogo di un DataAdapeter di un DataSet
ExecuteReader(string, CommandType, ConnectionState)	Esegue un DataReader in luogo di un DataAdapeter di un DataSet
ExecuteReader(string, ConnectionState)	Esegue un DataReader in luogo di un DataAdapeter di un DataSet
ExecuteScalar(string)	Esegue una funzione scalare definita nel Database ed associata al DbObject
ExecuteScalar(string, DbParameterCollection, CommandType, ConnectionState)	Esegue una funzione scalare definita nel Database ed associata al DbObject
ExecuteScalar(string, CommandType)	Esegue una funzione scalare definita nel Database ed associata al DbObject
ExecuteScalar(string, CommandType, ConnectionState)	Esegue una funzione scalare definita nel Database ed associata al DbObject

	ExecuteScalar(string, ConnectionState)	Esegue una funzione scalare definita nel Database ed associata al DbObject
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetByPrimaryKey(params object[])	Esegue una query mappando in maniera implicita il vettore di valori nelle colonne costituenti la chiave primaria del DbObject
	GetByQBEValues()	
	GetCurrentRowAsCSV(bool)	Ottiene la riga corrente come stringa formattata in CSV. Il parametro booleano se true indica che viene inclusa una riga con il nome delle colonne.
	GetCurrentRowAsXML(bool)	Ottiene la riga corrente come stringa formattata in XML. Il parametro booleano se true indica che viene incluso nell'XML il relativo schema.
	GetDataTable()	Ottiene il DataTable ADO.NET dei dati contenuti del DbObject.
	GetDataTableAsXML(bool)	Ottiene il DataTable ADO.NET corrente come stringa formattata in XML. Il parametro booleano se true indica che viene incluso nell'XML il relativo schema.
	GetDataTableFromDataReader(ref DataTable, string)	Ottiene il DataTable ADO.NET iterando sul DataReader .
	GetDataTableFromDataReader(ref DataTable, string, CommandType, ConnectionState)	Ottiene il DataTable ADO.NET iterando sul DataReader .
	GetDbColumn(string)	Ottiene un DbColumn a partire dal suo nome

GetDbColumns()	Ottiene la collection di tutti i DbColumn definiti nel DbObject.
GetDbParameters()	Ottiene la collection di tutti i parametri della query sottostante il DbObject.
GetDbSchema(ConnectionState)	Ottiene le Schema dell'oggetto database mappato dal DbObject.
GetEmptyDataTable()	Ottiene un DataTable vuoto mappato sull'oggetto database
GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
GetHashValue(byte[], HashProviders)	Ottiene il valore di Hash di un array di bytes.
GetHashValue(string, HashProviders)	Ottiene il valore di Hash di una stringa.
GetMaxDbValue(DbColumn)	Ottiene il valore massimo di un DbColumn nel database puntato dal DbObject
GetMaxDbValue(DbObject, DbColumn)	Ottiene il valore massimo del DbColumn di un DbObject (anche diverso da quello corrente).
GetPrimaryKeyDbColumns()	Ottiene la collection dei DbColumns che costituiscono la chiave primaria del DbObject
GetRows(DataSet)	Ottiene la collection delle Rows di un DataSet
GetTable(DataSet)	Ottiene la collection delle Table di un DataSet
GetType	Gets the Type of the current instance. (inherited from Object).
Init()	Inizializza l'oggetto DbObject associandolo ad un DbConfig

	Init(DbConfig, Providers)	Inizializza l'oggetto DbObject associandolo ad un DbConfig
	Init(string)	Inizializza l'oggetto DbObject associandolo ad un DbConfig
	Init(string, Providers)	Inizializza l'oggetto DbObject associandolo ad un DbConfig
	Init(Providers)	Inizializza l'oggetto DbObject associandolo ad un DbConfig
	InitBoundControls()	Inizializza i controlli Bindati ai DbColumn del DbObject
	Insert()	Esegue lo statement di Insert
	Insert(DataBoundControlsBehaviour)	Esegue lo statement di Insert
	LoadAll()	Carica i dati nel DbObject
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	MoveFirst()	Muove il cursore al primo record del set di dati sottostante il DbObject
	MoveLast()	Muove il cursore all'ultimo record del set di dati sottostante il DbObject
	MoveNext()	Muove il cursore al record successivo del set di dati sottostante il DbObject
	MovePrevious()	Muove il cursore al record precedente del set di dati sottostante il DbObject
	MoveTo(int)	Muove il cursore al record n del set di dati sottostante il DbObject
	Open(bool)	Inizializza il DbObject caricando o meno i dati sottostanti.

ReadFromBoundControls()	Aggiorna i dati dei DbColumn del DbObject con i valori dei controlli bindati ai DbColumn.
ReloadAll()	Ricarica i dati al valore originario senza rieseguire la query verso il database sottostante.
Remove()	Elimina una riga nel DataTable sottostante il DbObject.
Remove(DataBoundControlsBehaviour)	Elimina una riga nel DataTable sottostante il DbObject.
ResetDataTable()	Reimposta al valore originario i dati del DataTable sottostante
ResetError()	Azzera le condizioni di errore interne del DbObject.
ResetQBEAttributes()	Vedi oggetto QBEForm
ResetQBEvalues()	Vedi oggetto QBEForm
SaveAsCSV(string, string, bool, DataTable, DbColumns)	Salva I dati sottostanti il DbObject in un file CSV
SaveAsCSVMemoryStream(string, bool, DataTable, DbColumns)	Salva I dati sottostanti il DbObject in un oggetto .NET memory stream.
SaveAsXML(string, bool)	Salva I dati sottostanti il DbObject in un file XML
SaveDataReaderAsCSV(string, string, bool, DbDataReader, DbColumn[])	Salva I dati sottostanti un DataReader in un file CSV
SaveDataTableAsCSV(string, string, bool, DataTable, DbColumns)	Salva I dati sottostanti di un DataTable in un file CSV
SaveDataTableAsCSVMemoryStream(string, bool, DataTable, DbColumns)	Salva I dati sottostanti di un DataTable in un oggetto .NET memory stream.

SaveDataTableAsXML(string, bool)	Salva il DataTable sottostante il DbObject in XML
SequenceFrom(string, string)	Restituisce un valore in sequenza
SequenceFrom(string)	Restituisce un valore in sequenza
ToList<T>()	Converte i dati sottostanti il DbObject in una lista di oggetti di classe POCO <T>
ToObject<T>()	Converte i dati sottostanti il DbObject in un di oggetto di classe POCO <T>
ToString	Returns a string that represents the current object. (inherited from Object).
UndoChanges()	Annulla le modifiche apportate all'active record.
UnloadAll()	Pulisce l'insieme di dati sottostanti il DbObject.
Update()	Esegue il metodo di aggiornamento scrivendo i dati nel database sottostante. L'operazione non è annullabile se non inserita all'interno di un blocco di transazione.
Update(DataBoundControlsBehaviour)	Esegue il metodo di aggiornamento scrivendo i dati nel database sottostante. L'operazione non è annullabile se non inserita all'interno di un blocco di transazione.
UpdateBatch()	Esegue il metodo di aggiornamento scrivendo i dati nel database sottostante su blocchi di records in luogo di un record per volta. L'operazione non è annullabile se non inserita all'interno di un blocco di transazione.

UpdateBoundControls()	Aggiorna i valori dei BoundControl con il valore dei DbColumn associati.
UpdateBoundControls(DbColumn)	Aggiorna i valori dei BoundControl con il valore dei DbColumn associati.
UpdateFromDataSet()	Aggiorna i dati nel DbObject e nel database leggendoli dal DataSet interno.
UpdateFromDataSet(ref DataSet)	Aggiorna i dati nel DbObject e nel database leggendoli da un DataSet passato come parametro.
UpdateFromDataTable()	Aggiorna i dati nel DbObject e nel database leggendoli dal DataTable interno.
UpdateFromDataTable(ref DataTable)	Aggiorna i dati nel DbObject e nel database leggendoli da un DataTable passato come parametro.
ValidateBoundControls()	Esegue la validazione del valore delle proprietà dei BoundContro associati ai DbCOlumn
objRowsCount()	Il conteggio delle righe totali (senza filtro) dell'oggetto database mappato dal DbObject.

16.3 Eventi

	Name	Description
	BoundCompleted	Tutti i BoundControl sono stati valorizzati
	DataEventAfter	Prima dell'esecuzione di un metodo
	DataEventBefore	Dopo l'esecuzione di un metodo
	WriteCSVRow	Dopo la scrittura di una riga in un file CSV.

16.4 Strutture

	Name	Description
	Field	L'insieme dei campi dell'oggetto
	Join	L'insieme dei DbObject che compongono un Join
	Parameter	L'insieme dei parametri

16.5 Campi

	Name	Description
	AppendedSQL	Istruzione SQL da aggiungere ai comandi SQL Select, Delete, Insert, Update
	BatchRows	Numero di righe incluse in una singola operazione batch
	CachedConnection	DbConnection da usare come cache in luogo dell'aperura/chiusura delle connessioni interne. Obsoleta.
	CommandTimeOut	Intero . Secondi di Timeout di esecuzione comando
	CurrentDataRow	DataRow .NET corrente relativamente al DataTable sottostante il DbObject
	DbParameterNamePrefix	Stringa. Prefisso usato per indicare i parametri secondo le convenzioni del Provider
	DbParameterNamePrefixE	Stringa. Prefisso Esteso usato per indicare i parametri secondo le convenzioni del Provider
	DbProvider	Il tipo di provider Database
	DisableEvents	Booleano. Se true disabilita gli Eventi.
	ExecutionResult	L'oggetto contenente l'esito dell'esecuzione dell'ultimo metodo.
	RuntimeUI	Il tipo di UI a runtime.
	UpdateBatchEnabled	Booleano. Se true abilita comunque, se possibile, l'uso delle operazioni batch nelle operazioni di Update ed Insert. Non può essere usato per l'Insert in oggetti database che usano Identity.
	UseDataReader	Boolean. Se true forza il DbObject ad usare un DataReader in luogo di un DataAdapter ed un DataSet per reperire i dati nel

		Database. L'effetto è che i dati non vengono caricati TUTTI in memoria ma si ha un cursore in avanti.
	ValidatorLabelControl	E' l'oggetto di UI che verrà utilizzato per visualizzare lo stato di validazione di un DbColumn con BoundControls associati.
	WithOptions	Stringa. E' l'estensione "WithOptions" che viene aggiunta alle query SQLServer.

17 DbObjects Class

```
public class DbObjects : CollectionBase
```

18 DbOrderby Class

```
[Serializable()]
```

```
public class DbOrderby : ArrayList
```

19 DbParameter Class

```
[Serializable()]
```

```
public class DbParameter
```

19.1 Costruttori

	Name	Description
	DbParameter(string, DbType, ParameterDirection, int)	

19.2 Proprietà

	Name	Description
	DbParameter	
	DbType	
	Direction	
	ParameterName	
	Size	
	SourceColumn	
	SourceColumnNullMapping	
	SourceVersion	
	Value	

19.3 Metodi

	Name	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).

	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	ToString	Returns a string that represents the current object. (inherited from Object).

19.4 Campi

	Name	Description
	DbObject	

20 DbParameters Class

```
[Serializable()]
```

```
public class DbParameters : CollectionBase
```

20.1 Costruttori

	Name	Description
	DbParameters()	

20.2 Proprietà

	Name	Description
	Capacity	Gets or sets the number of elements that the CollectionBase can contain. (inherited from CollectionBase).
	Count	Gets the number of elements contained in the CollectionBase instance. This property cannot be overridden. (inherited from CollectionBase).

20.3 Metodi

	Name	Description
	Add(DbParameter)	
	Add(string, DbType, ParameterDirection)	
	Clear	Removes all objects from the CollectionBase instance. This method cannot be overridden. (inherited from CollectionBase).
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetEnumerator	Returns an enumerator that iterates through the CollectionBase instance. (inherited from CollectionBase).

	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	OnClear	Performs additional custom processes when clearing the contents of the CollectionBase instance. (inherited from CollectionBase).
	OnClearComplete	Performs additional custom processes after clearing the contents of the CollectionBase instance. (inherited from CollectionBase).
	OnInsert(Int32, Object)	Performs additional custom processes before inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnInsertComplete(Int32, Object)	Performs additional custom processes after inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnRemove(Int32, Object)	Performs additional custom processes when removing an element from the CollectionBase instance. (inherited from CollectionBase).
	OnRemoveComplete(Int32, Object)	Performs additional custom processes after removing an element from the CollectionBase instance. (inherited from CollectionBase).
	OnSet(Int32, Object, Object)	Performs additional custom processes before setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnSetComplete(Int32, Object, Object)	Performs additional custom processes after setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnValidate(Object)	Performs additional custom processes when validating a value. (inherited from CollectionBase).

	RemoveAt(Int32)	Removes the element at the specified index of the CollectionBase instance. This method is not overridable. (inherited from CollectionBase).
	ToString	Returns a string that represents the current object. (inherited from Object).

20.4 Estensioni

	Name	Description
	AsParallel()	Enables parallelization of a query. (Defined by ParallelEnumerable).
	AsQueryable()	Converts an IEnumerable to an IQueryable. (Defined by Queryable).
	Cast<TResult>()	Casts the elements of an IEnumerable to the specified type. (Defined by Enumerable).
	OfType<TResult>()	Filters the elements of an IEnumerable based on a specified type. (Defined by Enumerable).

21 DbRelationship Class

```
public class DbRelationship
```

21.1 Metodi

	Name	Description
	Do()	
	DoRelationShipLeft(bool)	
	DoRelationShipRight()	
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	ToString	Returns a string that represents the current object. (inherited from Object).

22 DbRelationshipItem Class

```
public class DbRelationshipItem
```

22.1 Metodi

	Name	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	ToString	Returns a string that represents the current object. (inherited from Object).

23 DbRelationshipItems Class

```
public class DbRelationshipItems : CollectionBase
```

23.1 Costruttori

	Name	Description
	DbRelationshipItems()	

23.2 Proprietà

	Name	Description
	Capacity	Gets or sets the number of elements that the CollectionBase can contain. (inherited from CollectionBase).
	Count	Gets the number of elements contained in the CollectionBase instance. This property cannot be overridden. (inherited from CollectionBase).

23.3 Metodi

	Name	Description

	Add(DbColumn, DbColumn)	
	Add(DbRelationshipItem)	
	Clear	Removes all objects from the CollectionBase instance. This method cannot be overridden. (inherited from CollectionBase).
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetEnumerator	Returns an enumerator that iterates through the CollectionBase instance. (inherited from CollectionBase).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	OnClear	Performs additional custom processes when clearing the contents of the CollectionBase instance. (inherited from CollectionBase).
	OnClearComplete	Performs additional custom processes after clearing the contents of the CollectionBase instance. (inherited from CollectionBase).
	OnInsert(Int32, Object)	Performs additional custom processes before inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnInsertComplete(Int32, Object)	Performs additional custom processes after inserting a new element into the CollectionBase instance. (inherited from CollectionBase).
	OnRemove(Int32, Object)	Performs additional custom processes when removing an element from the CollectionBase instance. (inherited from CollectionBase).

	OnRemoveComplete(Int32, Object)	Performs additional custom processes after removing an element from the CollectionBase instance. (inherited from CollectionBase).
	OnSet(Int32, Object, Object)	Performs additional custom processes before setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnSetComplete(Int32, Object, Object)	Performs additional custom processes after setting a value in the CollectionBase instance. (inherited from CollectionBase).
	OnValidate(Object)	Performs additional custom processes when validating a value. (inherited from CollectionBase).
	RemoveAt(Int32)	Removes the element at the specified index of the CollectionBase instance. This method is not overridable. (inherited from CollectionBase).
	ToString	Returns a string that represents the current object. (inherited from Object).

24 DbUser Class

```
[Serializable()]
public class DbUser
```

24.1 Metodi

	Name	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).

	ToString	Returns a string that represents the current object. (inherited from Object).

24.2 Campi

	Name	Description
	Description	
	Enabled	
	Name	
	SID	
	SIDType	

25 ExecutionResult Class

```
public class ExecutionResult
```

25.1 Costruttori

	Name	Description
	ExecutionResult(string)	

25.2 Metodi

	Name	Description
	Failed	
	Success	

25.3 Proprietà

	Name	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetFromDb(DbConfig)	
	GetFromDb(DbObject)	
	GetFromDbConfig(DbConfig)	
	GetFromDbObject(DbObject)	
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	Reset()	
	Set(ExecutionResult)	

	ToInnerExecutionResult()	
	ToString	Returns a string that represents the current object. (inherited from Object).

25.4 Enumerazioni

	Name	Description
	eResultCode	

25.5 Campi

	Name	Description
	Context	
	ErrorCode	
	Exception	
	InnerExecutionResult	
	ResultCode	
	ResultMessage	
	Value	

26 InnerExecutionResult Class

```
public class InnerExecutionResult
```

26.1 Costruttori

	Name	Description
	InnerExecutionResult(string)	

26.2 Proprietà

	Name	Description
	Failed	
	Success	

26.3 Metodi

	Name	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	Reset()	
	ToString	Returns a string that represents the current object. (inherited from Object).

26.4 Enumerazioni

	Name	Description
	eResultCode	

26.5 Campi

	Name	Description
	Context	
	ErrorCode	
	Exception	
	ResultCode	
	ResultMessage	
	Value	

27 SerializableDictionary(Of TKey, TValue) Class

```
[XmlRoot("dictionary")]
public class SerializableDictionary<TKey, TValue> : Dictionary<TKey, TValue>,
   IXmlSerializable
```

27.1 Proprietà

	Name	Description
	Comparer	Gets the IEqualityComparer<T> that is used to determine equality of keys for the dictionary. (inherited from Dictionary<TKey, TValue>).
	Count	Gets the number of key/value pairs contained in the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).
	Item(TKey)	Gets or sets the value associated with the specified key. (inherited from Dictionary<TKey, TValue>).
	Keys	Gets a collection containing the keys in the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).
	Values	Gets a collection containing the values in the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).

27.2 Metodi

	Name	Description
	Add(TKey, TValue)	Adds the specified key and value to the dictionary. (inherited from Dictionary<TKey, TValue>).
	Clear	Removes all keys and values from the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).
	ContainsKey(TKey)	Determines whether the Dictionary<TKey, TValue> contains the specified key. (inherited from Dictionary<TKey, TValue>).
	ContainsValue(TValue)	Determines whether the Dictionary<TKey, TValue> contains a specific value. (inherited from Dictionary<TKey, TValue>).

	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetEnumerator	Returns an enumerator that iterates through the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetObjectData(SerializationInfo, StreamingContext)	Implements the ISerializable interface and returns the data needed to serialize the Dictionary<TKey, TValue> instance. (inherited from Dictionary<TKey, TValue>).
	GetSchema()	This method is reserved and should not be used. When implementing the IXmlSerializable interface, you should return null (Nothing in Visual Basic) from this method, and instead, if specifying a custom schema is required, apply the XmlSchemaProviderAttribute to the class.
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	OnDeserialization(Object)	Implements the ISerializable interface and raises the deserialization event when the deserialization is complete. (inherited from Dictionary<TKey, TValue>).
	ReadXml(XmlReader)	Generates an object from its XML representation.
	Remove(TKey)	Removes the value with the specified key from the Dictionary<TKey, TValue>. (inherited from Dictionary<TKey, TValue>).
	ToString	Returns a string that represents the current object. (inherited from Object).
	TryGetValue(TKey,)	(inherited from Dictionary<TKey, TValue>).
	WriteXml(XmlWriter)	Converts an object into its XML representation.

28 Utilities Class

```
public sealed class Utilities
```

Classe di utilità

28.1 Costruttori

	Name	Description
	Utilities()	

28.2 Metodi

	Name	Description
S	CallByName(object, string, CallType, params object[])	Esegue chiamate CallType in un qualsiasi oggetto .NET
S	Cast(object, DbType)	Converte un oggetto nel corrispondente tipo DbType
S	Cast(object, DbType, DateTimeResolution)	Converte un oggetto nel corrispondente tipo DbType usando eventuali arrotondamenti se di tipo DateTime.
S	ConvertParameterValue(DbColumn, object)	Converte un valore in un valore compatibile con il tipo mappato dal DbColumn.
S	ConvertTo<T>(List<T>)	Conversioni da oggetti a liste di oggetti
S	ConvertTo<T>(List<DataRow>)	Conversioni da oggetti a liste di oggetti
S	ConvertTo<T>(DataTable)	Conversioni da oggetti a liste di oggetti

S	CreateDBTableFromDBObject(DbObject, string, Providers)	
S	CreateItem<T>(DataRow)	
S	CreateTable<T>()	
S	DBNullHandler(ref object)	
	DataTableRemoveDuplicatesRows(DataTable, string)	Rimuove le righe duplicate in un DataTable
S	DataTableRemoveDuplicatesRows(DataTable)	Rimuove le righe duplicate in un DataTable
S	DataTableToArray<T>(DataTable)	Converte un DataTable in array di <T>
S	DataTableToList<T>(DataTable)	Converte un DataTable in una lista di <T>
S	DbObject_GetTotalROWSCOUNT(DbObject)	Ottiene il conteggio di tutte le righe dell'oggetto database mappato da un DbOBject
S	DecryptRijndael(string, string, string)	Decifra una stringa
S	DecryptTripleDES(string, string, bool)	Decifra una stringa
S	DeserializeObjectFromFile(string, ref object)	Deserializza un oggetto a partire da una stringa contenente la sua forma serializzata.
S	DeserializeObjectFromXMLString(string, ref object)	Deserializza un oggetto a partire da una stringa XML contenente la sua forma serializzata.

S	EAN13CheckDigit(string)	Calcola il Check digit EAN-13
S	EAN8CheckDigit(string)	Calcola il Check digit EAN-8
S	EncryptRijndael(string, string, string)	Cifra una sringa
S	EncryptTripleDES(string, string, bool)	Cifra una sringa
S	EnumDescription(Enum)	Ottiene la parte testuale di un valore enumerato
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
S	FileToFileStream(object)	Converte un oggetto File in un File Stream
S	FileToMemoryStream(object)	Converte un oggetto File in un MemoryStream
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
S	GetControlFullName(object, string)	
	GetDBValue(object)	
S	GetFileExtension(string)	
S	GetFileName(string)	

S	GetFileNameWithoutExtension(string)	
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
S	GetHashValue(byte[], HashProviders)	
S	GetHashValue(string, HashProviders)	
S	GetIListFromDataTable(DataTable)	
S	GetListFromTable(DataTable, ICopy, IList)	
S	GetRandomNumber(int, int)	
S	GetSQLCreateTable(string, DataTable)	
S	GetSafeFileName(string)	
S	GetStringLeftValue(string, string)	
S	GetStringRightValue(string, string)	
	GetType	Gets the Type of the current instance. (inherited from Object).
S	GetURLWithoutFileName(string)	
S	GetUniqueFileName(string)	

S	GetValueSubString(string, int, int, bool)	
S	HighlightChangedBoundControls(DbObject)	
S	IListToDataTable<T>(IList<T>)	
S	IntYYMMDD_ToDate(int, int)	
S	IntYYYYMMDD_ToDate(int)	
S	IsBase64String(string)	
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
S	NewGUID()	
S	QuoteValue(string)	
S	Reflection_GetValidFilename(string)	
S	RegisterEventHandler(object, object)	
S	SendMail(string, string, string, string, string, string, string, string, string, MailPriority, bool, string, string, string, string)	
S	SequenceFrom(string)	
S	SequenceFrom(string, string)	

S	SerializeObjectToFile(string, object)	
S	SerializeObjectToXMLString(object)	
S	SetParameter(ref object, DbColumn)	
S	SimpleDeserialize<T>(string)	
S	SimpleSerialize<T>(T)	
S	StringYYMMDD_ToDate(string, int)	
S	StringYYYYMMDD_ToDate(string)	
	ToString	Returns a string that represents the current object. (inherited from Object).
S	TruncateDateTime(DateTime, DateTimeResolution)	

28.3 Interfacce

	Name	Description
	ICopy	

28.4 Enumerazioni

	Name	Description
	CallType	

29 ValidationRange Class

```
public class ValidationRange
```

29.1 Metodi

	Name	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	ToString	Returns a string that represents the current object. (inherited from Object).

29.2 Campi

Name	Description
	MaxValue
	MinValue

30 Validators Class

```
public sealed class Validators
```

30.1 Costruttori

	Name	Description
	Validators()	

30.2 Metodi

	Name	Description
S	CompareList(object, object, bool)	

	CompareListToString(object)	
	Equal(object, object, bool)	
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetType	Gets the Type of the current instance. (inherited from Object).
	GreaterThan(object, object, bool)	
	GreaterThanOrEqual(object, object, bool)	
	LessThan(object, object, bool)	
	LessThanOrEqual(object, object, bool)	
	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
	NotEqual(object, object, bool)	
	Range(object, object, object, bool)	
	RegExp(object, string, bool)	
	Required(object)	
	ToString	Returns a string that represents the current object. (inherited from Object).
	ValidEmail(string, bool)	
	ValidURI(string, bool)	
	ValidURL(string, bool)	

31 BasicDalSharedCode Class

```
public sealed class BasicDalSharedCode
```

31.1 Metodi

	Name	Description
	Cast(object, DbType)	
	Cast(object, DbType, DateTime, DateTimeResolution)	
	ConvertParameterValue(DbColumn, object)	
	DBNullHandler(ref object)	
	DataTableToArray<T>(this DataTable)	
	Equals(Object)	Determines whether the specified Object is equal to the current Object. (inherited from Object).
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (inherited from Object).
	GetHashCode	Serves as a hash function for a particular type. (inherited from Object).
	GetIListFromDataTable(DataTable)	
	GetListFromTable(DataTable, ICopy, IList)	
	GetType	Gets the Type of the current instance. (inherited from Object).
	IListToDataTable<T>(this IList<T>)	
	IsBoolean(this DataColumn)	
	IsChanged(this DbColumn)	
	IsNumeric(this DataColumn)	
	IsString(this DataColumn)	

	MemberwiseClone	Creates a shallow copy of the current Object. (inherited from Object).
S	QuoteValue(string)	
S	SetParameter(ref object, DbColumn)	
	ToString	Returns a string that represents the current object. (inherited from Object).
S	TruncateDateTime(DateTime, DateTimeResolution)	

31.2 Interfacce

	Name	Description
S	ICopy	

32 ACEAccessMask Enumeration

	Member name	Description
	Delete	
	FullControl	
	Insert	
	None	
	ReadExecute	
	Write	

33 ACEApplyTo Enumeration

	Member name	Description
	DbColumn	
	DbConfig	
	DbObject	

34 ACEAccessMask Enumeration

	Member name	Description
	Delete	
	FullControl	
	Insert	
	None	
	ReadExecute	
	Write	

35 BindingBehaviour Enumeration

	Member name	Description
	ReadWrite	
	ReadWriteAddNew	
	Write	
	WriteAddNew	

36 ComparisionOperator Enumeration

	Member name	Description
	Between	

	Equal	
	Equals	
	Exists	
	GreaterThan	
	GreaterThanOrEqualTo	
	In	
	IsNotNull	
	IsNull	
	LessThan	
	LessThanOrEqualTo	
	Like	
	None	
	NotBetween	
	NotEqualTo	
	NotIn	
	NotLike	

37 ConnectionState Enumeration

	Member name	Description
	CloseOnExit	
	KeepOpen	

38 CryptoProviders Enumeration

	Member name	Description
	MD5	
	TripleDES	

39 DataBoundControlsBehaviour Enumeration

	Member name	Description
	BasicDALDataBinding	
	NoDataBinding	
	WindowsFormsDataBinding	

40 DataEventType Enumeration

	Member name	Description
	AddNew	
	AddToDataSet	
	Binding	
	BindingFromDataReader	
	ControlsBinding	
	Delete	
	DeleteAll	
	DeleteFromDataSet	
	DeleteFromDataTable	
	Disposing	
	Initializing	
	Insert	
	MoveFirst	
	MoveLast	
	MoveNext	
	MovePrevious	
	MoveTo	
	Query	
	UndoChanges	

	Update	
	UpdateFromDataSet	
	UpdateFromDataTable	

41 DataTypeFamily Enumeration

	Member name	Description
	Binary	
	Boolean	
	DateTime	
	Numeric	
	String	
	Time	
	XML	

42 DateTimeResolution Enumeration

	Member name	Description
	Day	
	Hour	
	Millisecond	
	Minute	
	Month	
	Second	
	Tick	
	Year	

43 DbColumnProperties Enumeration

	Member name	Description
	AllowDBNull	
	BaseCatalogName	
	BaseColumnName	
	BaseSchemaName	
	BaseServerName	
	BaseTableName	
	ColumnName	
	ColumnOrdinal	
	ColumnSize	
	DataType	
	DataTypeName	
	IsAliased	
	IsAutoIncrement	
	IsColumnSet	
	IsExpression	
	IsHidden	
	IsIdentity	
	IsKey	
	IsLong	
	IsReadOnly	
	IsRowVersion	
	IsUnique	
	NonVersionedProviderType	
	NumericPrecision	

	NumericScale	
	ProviderSpecificDataType	
	ProviderType	
	UdtAssemblyQualifiedName	
	XmlSchemaCollectionDatabase	
	XmlSchemaCollectionName	
	XmlSchemaCollectionOwningSchema	

44 DbObjectTypeEnum Enumeration

	Member name	Description
	InMemoryJoin	
	Join	
	SQLQuery	
	ScalarFunction	
	StoredProcedure	
	Table	
	TableFunction	
	View	

45 ForeingKey Enumeration

	Member name	Description
	False	
	True	
	TrueNullable	

46 HashProviders Enumeration

	Member name	Description
	MD5	
	SHA256	
	SHA384	
	SHA512	

47 InterfaceModeEnum Enumeration

	Member name	Description
	Private	
	Property	
	Public	

48 JoinType Enumeration

	Member name	Description
	FullOuterJoin	
	InnerJoin	
	LeftOuterJoin	
	RightOuterJoin	

49 LogicOperator Enumeration

	Member name	Description
	AND	
	NOT	
	None	
	OR	

50 OrderBySequence Enumeration

Member name	Description
	Ascendig
	Descending

51 OrderBySequence Enumeration

	Member name	Description
	Ascendig	
	Descending	

52 ParameterStyles Enumeration

	Member name	Description
	Qualified	
	Simple	

53 Providers Enumeration

	Member name	Description
	ConfigDefined	
	ODBC	
	ODBC_DB2	
	OleDb	
	OleDb_DB2	
	OracleDataAccess	
	PostgreSQL	
	Sharepoint	
	SqlServer	
	Undefined	

54 Required Enumeration

	Member name	Description
	False	
	True	
	TrueNullable	

55 RuntimeUI Enumeration

	Member name	Description
	Service	
	WPF	
	Web	
	WindowsForms	
	Wisej	

56 SIDTypes Enumeration

	Member name	Description
	Group	
	User	

57 ValidationDataType Enumeration

	Member name	Description
	DateTime	
	Default	
	Numeric	
	String	

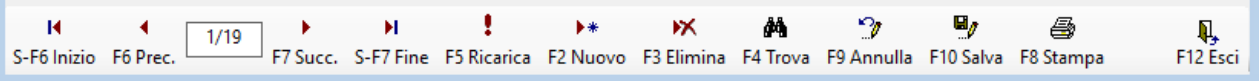
58 ValidationTypes Enumeration

	Member name	Description
	CompareList	
	Equal	
	None	
	Range	
	RegularExpression	
	Required	
	ValidEmail	
	ValidURI	
	ValidURL	

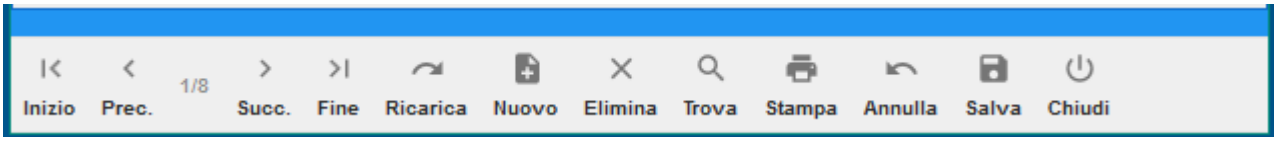
59 DataNavigator

L'oggetto DataNavigator è disponibile in due implementazioni specifiche per Windows Forms e Wisej.

Versione Windows Forms



Versione Wisej



Lo scopo dell'oggetto è quelli di concertare la navigazione e le funzionalità di gestione del DbObject e dei controlli UI associati ad esso (BoundControls) centralizzando la gestione dell'iterazione e degli eventi anche quando i dati sono associati da delle DataGridView.

59.1 Esempio d'uso

L'oggetto è un oggetto di UI utente. Va quindi inserito una sezione degli Strumenti posizionabili sulla superficie di disegno. Una volta posizionato sulla superficie esso è pienamente usabile a Design Time e Runtime.

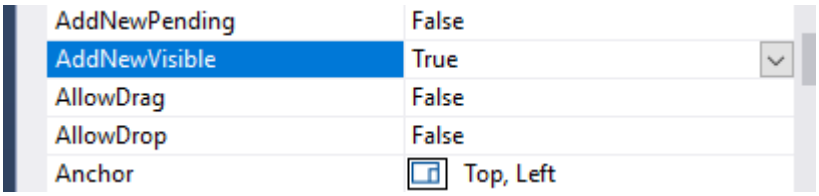
59.1.1 Dichiarazione e gestione degli eventi.

Ogni pulsante possiede le proprietà di visibilità, abilitazione, testo ed emette degli eventi delegati al DataNavigator.

La sua inizializzazione consiste nell'associazione fra il datanavigator ed il DbObject.

```
this.dataNavigator1.DbObject = this.db_comuni;
```

Le funzioni a disposizione dell'utente possono essere limitate abilitando a design-time



Oppure a run-time (es: `this.dataNavigator1.AddNewVisible = false`)

In configurazione di default questa è l'unica configurazione necessaria dato che tutte le operazioni di sono gestite internamente dal DataNavigator.

E' possibile tuttavia gestire ogni singolo evento manualmente impostando a false la proprietà

```
this.dataNavigator1.ManageNavigation = false
```

La gestione degli eventi può essere fatta associandoli tramite il designer oppure via codice come nell'esempio seguente

```
this.dataNavigator1.eAddNew += new BasicDALWisejControls.DataNavigator.eAddNewEventHandler(this.dataNavigator1_eAddNew);
this.dataNavigator1.ePrint += new BasicDALWisejControls.DataNavigator.ePrintEventHandler(this.dataNavigator1_ePrint);
this.dataNavigator1.eDelete += new BasicDALWisejControls.DataNavigator.eDeleteEventHandler(this.dataNavigator1_eDelete);
this.dataNavigator1.eRefresh += new BasicDALWisejControls.DataNavigator.eRefreshEventHandler(this.dataNavigator1_eRefresh);
this.dataNavigator1.eClose += new BasicDALWisejControls.DataNavigator.eCloseEventHandler(this.dataNavigator1_eClose);
```

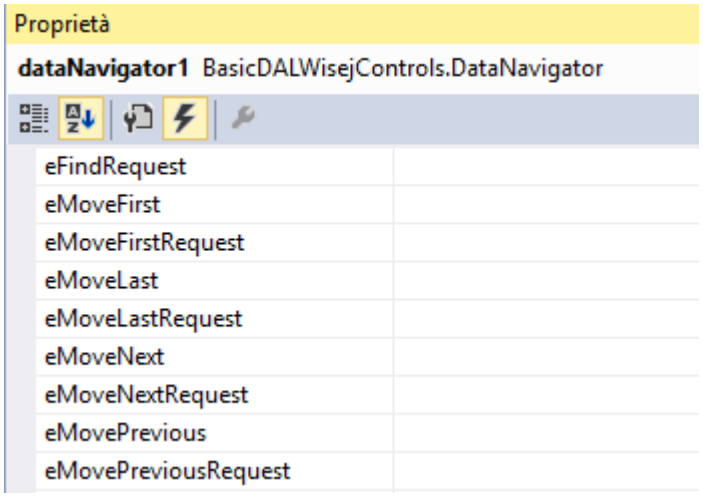
```
        this.dataNavigator1.eFind += new BasicDALWisejControls.DataNavigator.eFindEventHandler(this.dataNavigator1_eFind);
        this.dataNavigator1.eSave += new BasicDALWisejControls.DataNavigator.eSaveEventHandler(this.dataNavigator1_eSave);
        this.dataNavigator1.eMovePrevious += new BasicDALWisejControls.DataNavigator.eMovePreviousEventHandler(this.dataNavigator1_eMovePrevious);
        this.dataNavigator1.eMoveFirst += new BasicDALWisejControls.DataNavigator.eMoveFirstEventHandler(this.dataNavigator1_eMoveFirst);
        this.dataNavigator1.eMoveLast += new BasicDALWisejControls.DataNavigator.eMoveLastEventHandler(this.dataNavigator1_eMoveLast);
        this.dataNavigator1.eMoveNext += new BasicDALWisejControls.DataNavigator.eMoveNextEventHandler(this.dataNavigator1_eMoveNext);
        this.dataNavigator1.eUndo += new BasicDALWisejControls.DataNavigator.eUndoEventHandler(this.dataNavigator1_eUndo);
```

Gli eventi normalmente contengono il codice basico per la loro gestione.

```
private void dataNavigator1_eAddNew()
    {
        if (BasicDALWisejControls.Utilities.ManageDataNavigatorEvents(this.dataNavigator1, BasicDALWisejControls.DataNavigator.EventType.AddNew ) == DialogResult.No)
            return;

        if (this.dataNavigator1.DataGridActive == false)
        {
            this.dataNavigator1.DbObject.AddNew();
        }
        else
        {
            this.dataNavigator1.DataGrid_AddNew();
        }
    }
```

Anche quando è attiva la modalità di gestione interna delle operazioni è possibile tuttavia effettuare un controllo anteriormente all'esecuzione dell'operazione. A tale scopo esistono una serie di eventi con postfisso "Request". Questi eventi hanno il parametro Cancel che se impostato a true determina la cancellazione dell'evento principale.



Es.

```
private void dataNavigator1_eSaveRequest_1(ref bool Cancel)
    {

        if (datiok==false)
        {
            Cancel = true;
        }
    }
```

59.1.2 Uso del DataNavigator in associazione con una DataGridView

Il DataNavigator può essere usato anche in congiunzione con una DataGridView. In tal caso cambia lievemente la gestione. Oltre ad DbObject deve essere associata anche la DataGridView. La DataGridView associata deriverà il suo DataSource dal DataTable sottostante il DbObject associato al DataNavigator.

```
this.dataNavigator1.DataGrid = this.myDataGridView;
this.dataNavigator1.DataGridActive = true;
```

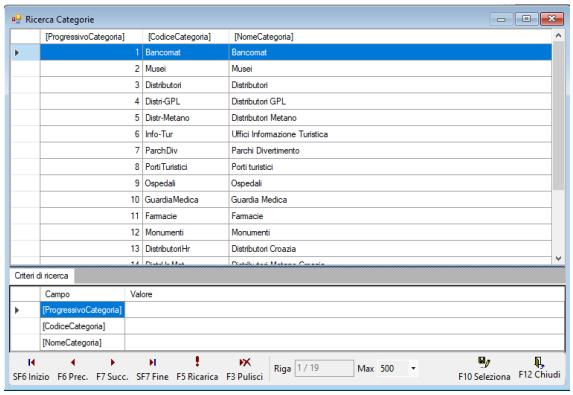
Esistono poi tutta una serie di metodi che implementano le funzioni applicative nel contesto della DataGridView. Tali metodi sono prefissati con "DataGrid_".

Tutte le impostazioni sono accessibile SOLO a run-time e non a design-time.

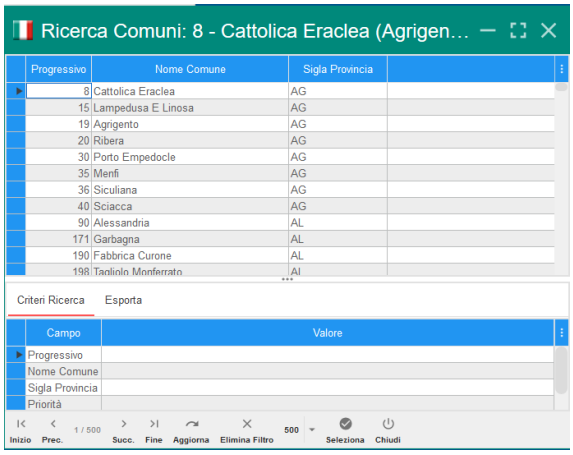
60 L'Oggetto QBEForm

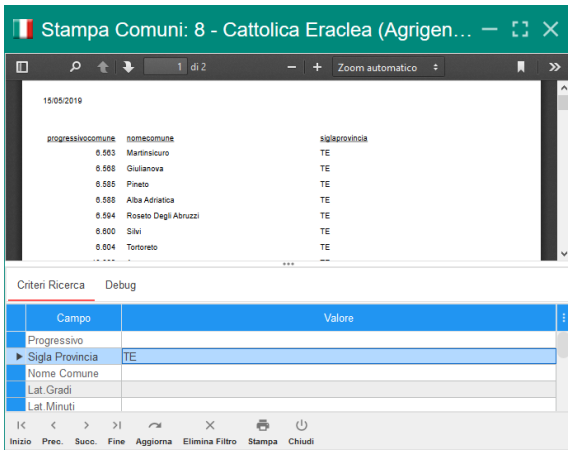
L'oggetto QBEForm implementa un sistema di ricerca, navigazione, selezione dei dati e di visualizzazione di reports di stampa. Le attuali implementazioni sono per Windows Forms e Wisej. L'oggetto non è disponibile a design-time ma solo a run-time.

Windows Forms



Wisej





60.1 L'instanziazione del QBEForm

L'oggetto non disponibile a design-time ma solo a run-time. Va quindi referenziato l'assembly contenente l'oggetto a seconda della UI scelta (BasicDALControls.dll per Windows Forms e BasicDALWisejControls.dll per Wisej).

```
BasicDALWisejControls.QBEForm QBEForm = new BasicDALWisejControls.QBEForm();
```

60.1.1 L'uso come selezione di dati (QbeMode.Query)

Impostando la proprietà QBEForm.Mode = BasicDALWisejControls.QbeMode.Query si attiva la modalità di Query. In questa modalità QBEForm permette di esplorare i dati un DbObject (QBEForm.DbObject), di selezionarli e ritornare i dati o la query che ha prodotto i dati ad un altro DbObject (Il QBEForm.QueryDbObject) impostando il valore della proprietà QBEResultMode. E' possibile definire quali DbColumn (QBEColumn) verranno visualizzati e/o usati come oggetti di filtro interattivo (Query By Example). I valori possibili di QBEResultMode sono

```
public enum QBEResultMode
{
          BoundControls = 0,  // Mappa DbColumn del QBEForm.DbObject sul QBEForm.QueryDbObject
          AllRows = 2,   //imposta I filtri del QBEForm.QueryDbObject con i medesimi filtri del QBEForm.DbObject
          SingleRow = 1, //imposta I filtri del QBEForm.QueryDbObject con i filtri del primo record del QBEForm.DbObject
          SelectedRows = 3, //imposta I filtri del QBEForm.QueryDbObject con i filtri dei record selezionati sulla Griglia
          DataTable = 4,  //torna il DataTable
          DBObject = 5  //imposta QBEForm.QueryDbObject con QBEForm.DbObject
}


private void QBE_db_comuni()
{
        db_comuni QBEDbObject = new db_comuni();
        QBEDbObject.Init(this.DbConfig);

        BasicDALWisejControls.QBEForm QBEForm = new BasicDALWisejControls.QBEForm();
        QBEForm.Mode = BasicDALWisejControls.QbeMode.Query;
        QBEForm.ResultMode = BasicDALWisejControls.QBEResultMode.SelectedRows;// ' torna tutte le righe selezionate
        QBEForm.CallerForm = this;
        QBEForm.Text = "Ricerca " + this.Text;
        QBEForm.DbObject = QBEDbObject;
        QBEForm.QueryDbObject = db_comuni;
        QBEForm.AutoLoadAll = true;
        QBEForm.QBEColumns.Add(QBEDbObject.progressivocomune);
        QBEForm.QBEColumns.Add(QBEDbObject.nomecomune);
        QBEForm.QBEColumns.Add(QBEDbObject.siglaprovincia );
        QBEForm.QBEColumns.Add(QBEDbObject.prioritacomune );

        QBEForm.DbColumnsMapping.Add(QBEDbObject.progressivocomune, db_comuni.progressivocomune);
        QBEForm.ShowQBE(this);
}
```

.CallerForm indica la finestra a cui rendere il focus dopo la chiusura del QBEForm.
.Text è il titolo della finestra QBEForm.
.AutoLoadAll = true indica che verrà effettuata una query al caricamento della QBEForm senza che l'utente prema il bottone "Aggiorna".

60.1.2 La collection QBEColumns

La Collection QBEColumns permette di definire quali DbColumn verranno mostrati nella griglia dei risultati e quali eventualmente usare nella lista dei parametri di Query By Example. Nella forma completa il metodo di Add è il seguente:

```
.Add(BasicDAL.DbColumn DbColumn, string FriendlyName = "", string DisplayFormat = "", object QBEValue = null, bool UseInQBE = true, bool
DisplayInQBEResult = true)
```

Parametro	Descrizione
DbColumn	E' un oggetto DbColumn definito del QBEForm.DbObject
FriendlyName	Opzionale, è il nome con cui viene mostrato all'utente il nome del DbColumn. Se omesso è il FriendlyName del DbColumn.

DisplayFormat	Opzionale, è la formattazione usata per mostrare il valore del DbColumn.
QBEValue	Opzionale, è il valore di default per la ricerca.
UseInQBE	Opzionale, se vero indica che il DbColumn sarà fra i parametri di QBE per filtrare i dati.
DisplayInQBEResult	Opzionale, se vero indica che il DbColumn sarà una delle colonne nella griglia dei risultati.

60.1.3 La collection DbColumnsMapping

La collection DbColumsMapping effettua una mappatura 1-1 fra le DbColumn del QBEForm.DbObject (l'oggetto che fa la query sul DB) ed il QBEForm.QueryDbObject (l'oggetto che vuole usare il risultato della Query). Lo scopo della mappatura è quello di permettere la creazione di un insieme di criteri di filtro applicabili al QueryDbObject a partire da quelli definiti nel QBEForm. Questo è necessario quando la modalità `QBEForm.ResultMode` è impostata su AllRows, SingleRow o SelectedRows.

```
public DbColumnMapping Add(BasicDAL.DbColumn DbObjectDbColumn, BasicDAL.DbColumn QueryDbObjectDbColumn)
```

60.1.4 L'uso come visualizzatore di reports (QbeMode.Report)

```
private void Report_db_comuni()
    {
        db_comuni  QBEDbObject = new db_comuni();
        QBEDbObject.Init(this.DbConfig);
        BasicDALWisejControls.QBEForm QBEForm = new BasicDALWisejControls.QBEForm();
        QBEForm.Mode  = BasicDALWisejControls.QbeMode.Report;
        QBEForm.AddReport("Elenco Comuni", "ElencoComuni.rpt", "Elenco dei comuni");
        QBEForm.DefaultReport = QBEForm.Reports["Elenco Comuni"];
        QBEForm.SessionStoreMode = CXMLSession.SessionStore.StoreModes.MemoryMappedFile;
        QBEForm.Text = "Stampa " + this.Text;
        QBEForm.ReportsPath = this.AppConfig.ReportsPath;
        QBEForm.ReportViewerMode = this.AppConfig.ReportViewerMode;
        QBEForm.CrystalReportViewerURL = BasicDAL.Utilities.GetURLWithoutFileName(Application.Url) + QBEForm.CrystalReportViewerPage;
        QBEForm.DbObject = QBEDbObject;
        QBEForm.ShowQBE(this);
    }
```

61 L'oggetto DbLookUp

L'oggetto DbLookUp permette di associare un meccanismo di look-up e binding di valori a degli oggetti di UI a partire dal valore di una proprietà di un oggetto associato tramite BoundControl ad un DbColumn.

L'uso tipico del DbLookup è la validazione dei dati immessi dall'utente e la conseguente valorizzazione di oggetti di UI. Un esempio di UI

Alla digitazione di un valore nella textBox "Prog.Camp." se il dato digitato è riconosciuto come Progressivo Campeggio valido le textbox "Nome Campeggio" e "Comune" vengono valorizzate.

61.1 Uso dell'oggetto DbLookUp

L'oggetto viene dichiarato nel seguente modo

```
private BasicDAL.DbLookUp dblcampeggio = new BasicDAL.DbLookUp();
```

L'oggetto deve quindi essere associato ad un DbObject che verrà utilizzato per le operazioni di lookup dei dati

```
this.dblcampeggio.DbObject = this.db_campeggi_qbedettagli;
```

Per poter effettuare i lookup sul DbObject è possibile usare la collection dei Filters. Si tratta di una collection di DbFilters che ottengono dinamicamente il valore di filtro a partire da una proprietà di un oggetto di UI associato al filtro come BoundControl. La sintassi è la seguente

```
this.dblcampeggio.Filters.AddBoundControl(this.db_campeggi_qbedettagli.progressivocampeggio,
BasicDAL.ComparisionOperator.Equal, this.txtProgressivoCampeggio, "text", BasicDAL.LogicOperator.None);
```

oppure associando direttamente un DbFilters esistente o creato direttamente nel metodo Add.

```
this.dblcampeggio.Filters.Add(DbFilter);


this.dblcampeggio.Filters.Add (this.db_campeggi_qbedettagli.progressivocampeggio, BasicDAL.ComparisionOperator.Equal, Value,
BasicDAL.LogicOperator.None);
```

La valorizzazione degli elementi di UI avviene associando uno o più BoundControl al DbLookUp

```
this.dblcampeggio.BoundControls.Add(this.db_campeggi_qbedettagli.nomecampeggio, this.txtNomeCampeggio, "text");
this.dblcampeggio.BoundControls.Add(this.db_campeggi_qbedettagli.nomecomune, this.txtLocalitaCampeggio, "text");
```

E' possible invocare il lookup usando il metodo

```
this.dblcampeggio.LookUpByFilters();
```

Il metodo .LookUpByFilters() può essere invocato ad esempio nell'evento di

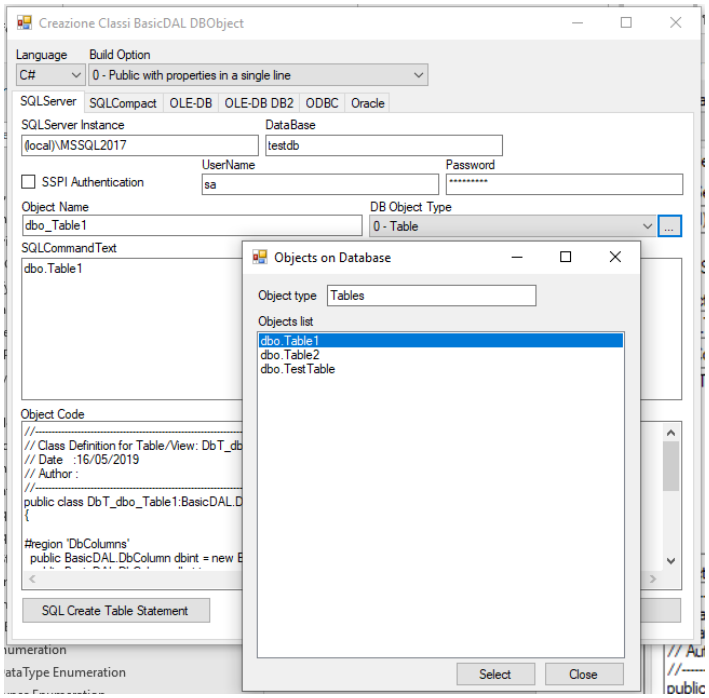
```
private void txtProgressivoCampeggio_TextChanged(object sender, EventArgs e)
    {
        this.dblcampeggio.LookUpByFilters();
    }
```

61.2 Validazione del DbLookUp

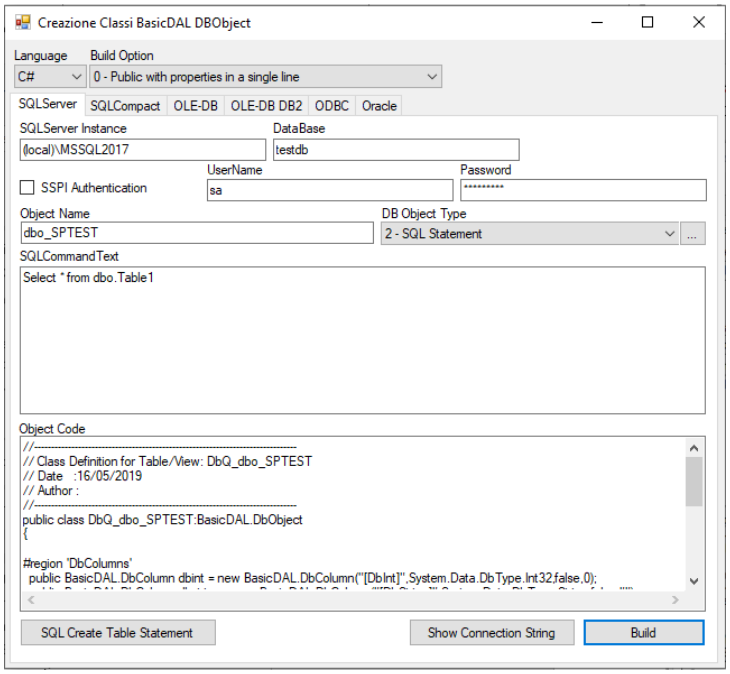
La validazione del lookup si ottiene invocando il metodo .Validate(valore) o il metodo .ValidateByFilters(). Entrambe i metodi restituiscono true o false a seconda che il valore (nel caso di Validate()) o i valori dei Filtri collegati siano esistenti o meno del DbObject collegato al DbLookup. La loro esecuzione, oltre a all'esecuzione del metodo .LookUpByFilters() imposta a true o false la proprietà .Validated().

62 BasicDAL ClassBuilder

La creazione delle classi DbObject può essere automatizzata con il programma di utility BasicDAL ClassBuilder. Lo scopo del programma è quello di creare classi DbObject in C# e VB.NET usando le varie tecnologie di connessione database disponibili in ADO.NET.



E' possibile ottenere lo schema di un database suddiviso per le tipologie di oggetti (tabelle, View, Stored Procedure, Funzioni), selezionare l'oggetto per il quale si vuole creare il DbObject e quindi generare il codice in VB.NET o C#. E' possibile anche definire una query SQL



Una volta generato il codice è possibile selezionarlo nella text-box "Object Code".