



DesignedNet Object Model Reference

Software Application Layers

DesignedNet.Dal

- Data Access Layer
 - Derived from a single base class that defines the minimum interface
 - ExecCmd()
 - Contains base classes for each common data store
 - Sql Server Database
 - Access Database
 - OleDb Database
 - Active Directory
 - File System
 - Incorporates transparent caching for consumer
 - Automatically added data to cache stack when requested
 - Checks cache for data when requested
 - Invalidates/updates cache of data when required

DesignedNet.State

- StateItem supported state types:
 - DataRow
 - SqlDataReader
 - OleDbDataReader
 - XmlNode
 - XmlReader
- Provides state management for data columns or attributes in any supported state type
 - object GetField(string index)
 - Default indexer serves same functionality
 - A function is implemented for each supported CLR type
 - string GetString(string index)
 - bool GetBoolean(string index)
 - void SetField(string index, object value)
 - Default indexer serves same functionality
 - A function is implemented for each supported CLR type
 - void SetString(string index, string value)
 - void SetBoolean(string index, bool value)
- Provides functionality to return related state item or state item arrays
 - StateItem GetChild(string relationship)
 - Not supported for SqlDataReader or OleDbDataReader
 - DataRow returns the first DataRow in GetChildRows(relationship)
 - XmlNode returns the first child element where NodeName = relationship
 - XmlReader returns the first child element where NodeName = relationship

- StateItemArray GetChildren(string relationship)
 - Not supported for SqlDataReader or OleDbDataReader
 - DataRow returns DataRow array from GetChildRows(relationship)
 - XmlNode returns the element collection where NodeName = relationship
 - XmlReader returns the element collection where NodeName = relationship
- StateItemArray supported state types:
 - DataView
 - DataTable
 - DataRow[]
 - SqlDataReader
 - OleDbDataReader
 - XmlReader
 - XmlNodeList
- Provides cursor functionality to navigate multiple records managed by state type
 - int Count() Returns the count of records in array
 - bool Reset() Returns cursor to first record in array
 - bool MoveNext() Moves cursor to next record in array
 - bool MoveTo(int position) Moves cursor to specified position in array
 - StateItem Current() Returns state item for current position in array
- Implements IEnumerable interface to support enumeration on all state items in array

DesignedNet.Biz

- Business Logic Layer
 - Derived from a single base class that defines the minimum interface
 - New()
 - Read()
 - Load()
 - Validate()
 - Save()
 - References a state object to maintain its state
 - References a BizError collection to gather object errors
 - Provides typed properties to read or modify the value of any attribute
 - DateTime _dateCreated { get {} set {} }
 - Provides string properties to read the value of any attribute
 - string DateCreated { get {} }
 - Provides methods to attempt the conversion and formatting of any string into the typed property
 - bool SetDateCreated(string dateCreated)
 - Provides properties that return an entity object or entity collection for related data
 - Entity collection for any table referencing the primary key
 - Entity collection for any table referenced via a many to many table
 - Entity object for any table referenced by a foreign key
 - Provides a base collection class to data source binding or list enumeration
 - Implements the ICollection and IEnumerable interfaces
 - Provides methods to populate itself with lists of entity objects
 - Provides methods to filter and/or sort list and return a new collection



- An entity object can be returned from a collection via its primary key
- An entity object state can be inserted or removed from the collection
- Provides a base enumerator class to enumerate through a collection class
 - Implements the IEnumerator interface to enable base class enumeration
 - References base state object, casts to specific types to provide
 - Enumerator class deals with state layer directly to support MoveNext
 - The propriety implementation for each state type supported is required



Business Object Interface Definitions

DesignedNet.Biz.IBizBase

Properties

StateBase State { get {} }	Returns the state object for the business object
BizErrors Errors { get {} }	Returns the errors collection object for the business object
string EntityName { get {} }	Returns the entity name for the business object

Methods

bool New()	Creates a new empty state for the business object
bool Save()	Attempts to persist current object state to default data store
bool Save(bDeep)	Attempts to persist all state in current the entity state object
bool Validate()	Validates attributes of current object state before persisting
bool Read(StateBase state)	Instantiates the business object from the state specified

DesignedNet.Biz.IBizEntityItem : IBizBase

Properties

IBizItem All { get {} }	Returns the reference to the source business list object
object ID { get {} }	Returns an object which holds this objects unique ID value
* Typed Attributes	Gets or sets the CLR typed value for the named attribute
* Entity Attributes	Gets or sets the IBizEntityItem object for the object or list
* String Attributes	Returns the formatted string value for the named attribute

Methods

bool Load(object ID)	Loads the object specified from the default data source
* Set Methods	Attempts to convert the string value for the named attribute

DesignedNet.Biz.IBizEntityList : IBizBase, ICollection

Properties

int Count { get {} }	Returns the count of the number of entities in the list
----------------------	---

Methods

bool Add(IBizItem)	Adds a business entity item to the collection
bool Exists(object ID)	Returns true if the object referenced by the ID exists
bool Remove(object ID)	Removes the object referenced by the ID from the collection
IBizItem AddNew()	Adds to the collection and returns an empty business entity
IBizItem Item[object ID]	Attempts to located the object in the list identified by the ID
IBizItemList Filter(BizFilter filter)	Returns a sorted and filtered entity list fitting the filter criteria



Business Object Entity Examples

BizEntity : IBizEntityItem

Typed Attributes: Returns or updates the CLR typed value for the underlying data attribute maintained in the state entity object. Errors during a null conversion are possible if the attribute is referenced and the underlying attribute value is null with the exception of primary and foreign keys which -1 will represent a null value.

```
int _pkEntity { get {} set {} }      Gets or sets the primary key for the entity business object
int _fkChildEntity { get {} set {} } Gets or sets the foreign key to reference a child entity
string _name { get {} set {} }      Gets or sets the string value for the Name attribute
DateTime _created { get {} set {} } Gets or sets the DateTime value for the Created attribute
double _amount { get {} set {} }    Gets or sets the double value for the Amount attribute
```

Entity Attributes: Returns or updates the business object that represents the ID for the referenced attribute. Child attributes directly reference a single object identified by the foreign key (Many to one: M2O). Related entity objects reference this object by its primary key and that child entity object will have a related entity object list for the parent objects that reference it (One to many: O2M). A referenced entity list refers to a collection of objects that is defined by a many to many relationship (Many to many: M2M). If the many to many table contains any more than the two foreign keys, and optionally a primary key, then this relationship will be created as a related entity list with each related entity having a child entity that refers to the other item referenced by the foreign key. The state of the current business object is first checked to see if the state is already contained within. If the object is based off of a DataSet then it is checked for a relationship to obtain a reference to the ChildRows[] collection. If the state is based off of Xml then the child elements are checked for the expected data. DataReaders are ignored because they always contain two dimensional data. If the state is not found within the parent object's state and a foreign key value is specified the entity object is read from the default data store and will attempt to use the same state type that the parent object is based on. If the same state type is supported and the state type is not read only then the data is added to the parent's state object and any required relationship information is appended. If no value is specified or the data is not found in the data store then an empty object is returned with a default state.

```
BizChildEntity ChildEntity { get {} set {} }      Gets or sets entity object (M2O)
BizRelatedEntityList RelatedEntities { get {} set {} } Gets or sets entity list (O2M)
BizReferencedEntityList ReferencedEntities { get {} set {} } Gets or sets entity list (M2M)
```

String Attributes: Returns the string representation for the underlying attribute value. Nulls are returned as a zero length string. These properties should be used for data binding or display.

```
string Name { get {} }      Returns the formatted string value for the _name attribute
```



string Created { get {} } Returns the formatted string value for the _created attribute
string Amount { get {} } Returns the formatted string value for the _amount attribute

Set Methods: Each method attempts to parse and format the string value into the underlying attribute's data type. The success of the conversion attempt is returned by the method as a BizError object that can be checked by the consumer for the presence of an error. If the conversion is not successful then the original value is retained. A zero length string can be passed in to set the underling value to null if required.

bool SetCreated(string created) Attempts to assign the string value to the _created attribute
bool SetAmount(string amount) Attempts to assign the string value to the _amount attribute

