

# Early Adopter Release Notes

March 29th, 2010  
EntitySpaces, LLC  
The EntitySpaces Team



Persistence Layer and Business Objects for Microsoft .NET

## CONTENTS

Purpose .....	3
Requirements.....	4
Microsoft .NET 3.5 Framework .....	4
Breaking Changes .....	5
Custom Classes .....	5
esEntity.AddNew().....	5
esDataRowState .....	5
esEntity.GetColumn .....	6
Query.QueryReset().....	6
esEntityCollection.RowStateFilter .....	6
esEntityCollection.Sort .....	6
esEntityCollection.Filter .....	6
esEntityCollection.LowLevelBind() .....	6
Non Breaking Changes .....	7
Custom Base Classes.....	7
Generated Collection Classes.....	7
EmployeesCollection .....	7
esEmployeesCollection .....	7
esEntityCollection .....	8
Generated Entity Classes .....	8
Employees .....	8
esEmployees .....	9
esEntity .....	9
SQLite Support .....	10
Known Issues .....	10
Support.....	10
Blog Posts on EntitySpaces 2010.....	10

## PURPOSE

This release is being made available to you and your team. Please keep this release confidential, do not share this release with anyone. This release is also set to expire some distance in the future so do not attempt to go live on this release.

The EntitySpaces 2010 release has been totally rewritten from the EntitySpaces 2009 release. We no longer use DataTable's or DataRow's under the hood for your storage mechanism. The protected Table and Row properties no longer exist. We have also changed the class inheritance and now use .NET Generics in the Collection class inheritance tree. There are also other significant changes. We have worked very hard to keep the breaking changes to a minimum.

The purpose of this Early Adopter release is twofold. First, your participation in this release will help us add many more hours of real world use before our ES2010 public beta. Secondly, we want to hear your feedback on some of the new features such as our powerful WCF Proxy / Silverlight enhancements, our SQLite support and so on.

We are grateful that you have volunteered to participate in the EntitySpaces 2010 Early Adopter Release. It is our hope that you can convert existing applications over to ES2010 and begin to use it successfully. Please let us know about anything you discover that is different between ES2009 and ES2010 and is not mentioned in this document. We will be updating this document for the public ES2010 Beta when it is released.

**Finally, and important,** your EntitySpaces customer account has been added to a special role which will allow you to post in our **ES2010 Early Adopters** Forum located in our normal forums. Only early adopters can view or post in this forum. This forum will be deleted when the public beta for EntitySpaces 2010 is released.

## REQUIREMENTS

### Microsoft .NET 3.5 Framework

The EntitySpaces 2010 architecture requires the Microsoft .NET 3.5 Framework. It does not support running under .NET 2.0. Customers that wish to develop for the .NET 2.0 Framework will have to remain on EntitySpaces 2009.

## BREAKING CHANGES

### Custom Classes

While this is not really a breaking change, we decided to list this change here. When custom classes are generated in EntitySpaces 2010, they have constructors in them and all of the necessary namespaces needed for adding custom methods. The ability to place custom constructors in the custom classes has been an often-requested feature. You probably will not have to do anything to your custom classes. We just wanted you to be aware of these changes. The custom template will not overwrite existing files since custom classes contain your custom business logic. However, you can now manually add constructors to your existing custom classes if you want to place logic there.

#### *ES2009*

```
using EntitySpaces.Interfaces;
using EntitySpaces.Core;

namespace BusinessObjects
{
    public partial class Categories : esCategories
    {

    }
}
```

#### *ES2010*

```
using EntitySpaces.Core;
using EntitySpaces.Interfaces;
using EntitySpaces.DynamicQuery;

namespace BusinessObjects
{
    public partial class Categories : esCategories
    {
        public Categories()
        {

        }
    }
}
```

### esEntity.AddNew()

This method has been removed. It was optional in 2009 and has finally been removed. If you had overloaded or overridden this virtual method you can move that logic to the constructor in your Custom classes (see above).

### esDataRowState

Since we no longer use a DataRow to hold an entity's data, we no longer use the ADO.NET **DataRowState** enum. Instead, we use our own enum esDataRowState. Single entities have an esDataRowState of "Added" upon creation and IsDirty is true.

## esEntity.GetColumn

This method now returns null or Nothing and not DBNull.Value

## Query.QueryReset()

This method no longer exists, you never re-use existing queries, you simply instantiate new ones.

## esEntityCollection.RowStateFilter

This property no longer exists.

## esEntityCollection.Sort

This property no longer exists. You now use the esEntityCollection.Filter property to sort, filter, or perform both operations at the same time. See below.

## esEntityCollection.Filter

This property is no longer a “string” type. Instead, it is of type IQueryable<T>. This method no longer relies on ADO.NET’s quirky syntax and instead uses the new Microsoft .NET 3.5 syntax. See the following examples:

- *Sorting by the Employee’s first name*

```
coll.Filter = coll.AsQueryable().OrderByDescending(d => d.FirstName);
```

- *Filtering for all Employees missing their FirstName*

```
coll.Filter = coll.AsQueryable().Where(d => d.FirstName == null);
```

- *Both Filtering and Sorting*

```
coll.Filter = coll.AsQueryable().Where(d => d.FirstName == null).OrderByDescending(d => d.LastName);
```

## esEntityCollection.LowLevelBind()

This method no longer exists. It is no longer necessary.

## NON BREAKING CHANGES

### Custom Base Classes

The shipping Custom Base Classes for EntitySpaces 2010 have changed slightly to reflect the changes in the architecture. If you are using the optional, advanced “Custom Base Class” feature of EntitySpaces, and have custom code in those files, then you should manually replicate the changes to them in your project. You will find them in the **C:\Program Files\EntitySpaces 2010\CodeGeneration\CustomBase** folder (just as in EntitySpaces 2009). If you have not added custom code, then you can just copy the new versions over the old.

### Generated Collection Classes

The Collection classes have changed the most. However, this should not really affect your current code base. We preserved the API as much as possible.

---

#### EmployeesCollection

- ES2009

```
public partial class EmployeesCollection : esEmployeesCollection
```

- ES2010

```
public partial class EmployeesCollection : esEmployeesCollection, IEnumerable<Employees>
```

---

#### esEmployeesCollection

- ES2009

```
abstract public class esEmployeesCollection : esEntityCollection
```

- ES2010

```
abstract public class esEmployeesCollection : esEntityCollection<Employees>
```

## esEntityCollection

- ES2009

```
public abstract class esEntityCollection : esEntityData,
    IEntityCollection,
    IComponent,
    IDisposable,
    IBindingList,
    IList,
    ICollection,
    IEnumerable,
    ITypedList,
    IListSource,
    ICommittable
```

- ES2010

```
public abstract class esEntityCollection<T> : esEntityCollectionBase,
    ICommittable,
    IComponent,
    IDisposable,
    IList<T>,
    ICollection<T>,
    IEnumerable<T>,
    ITypedList,
    IBindingList,
    IList,
    ICollection,
    IEnumerable,
    ICancelAddNew,
    IRaiseItemChangedEvents
    where T : EntitySpaces.Core.esEntity, new()
```

## Generated Entity Classes

### Employees

- ES2009 (no change)

```
public partial class Employees : esEmployees
```

- ES2010 (no change)

```
public partial class Employees : esEmployees
```



---

## esEmployees

- ES2009

abstract public class **esEmployees** : esEntity, INotifyPropertyChanged

- ES2010

abstract public class **esEmployees** : esEntity

---

## esEntity

- ES2009

public abstract class **esEntity** : esEntityData, IEntity, IEditableObject, IDataErrorInfo, ICommittable

- ES2010

public class **esEntity** : IEditableObject, IEntity, ICommittable, INotifyPropertyChanged, IDataErrorInfo

## SQLITE SUPPORT

We have added support for SQLite. You should be able to connect and generate your EntitySpaces classes as normal. The provider is named **EntitySpaces.SQLiteProvider.dll** and is bound to the 1.0.65.0 version. The SQLite ADO.NET provider can be found here: <http://sqlite.phxsoftware.com/>

```
connectionString="Data Source=C:\Databases\Northwind.db3;Version=3;"
```

## KNOWN ISSUES

- The IDataErrorInfo implementation on the Collection classes is not complete
- The VB.NET Silverlight Sample has some issues (probably due to porting) but will be fixed upon the next release.
- Our Windows.Forms sample in this release will be completely rewritten, we just shipped our test program(s).

## SUPPORT

As stated earlier, please only post questions regarding this Early Adopter Release in the new Early Adopter Forum area. We will be monitoring that forum very closely.

## BLOG POSTS ON ENTITYSPACES 2010

**[EntitySpaces 2010 Sneak Preview \(I\)](#)**

**[EntitySpaces 2010 Sneak Preview \(II\)](#)**

**[EntitySpaces 2010 Sneak Preview \(III\)](#)**

**[EntitySpaces 2010 Sneak Preview \(IV\)](#)**

**[EntitySpaces 2010 Supports the SQL Server 2008 types SqlGeometry, SqlGeography, more ...](#)**

**[EntitySpaces 2010 – More Template Friendly](#)**

**[EntitySpaces 2010 SQLite Support](#)**

**[EntitySpaces 2010 – Killer Silverlight Binding](#)**

**[EntitySpaces 2010 Two Part Series on Silverlight and WCF](#)**

**[JSON Serialization and EntitySpaces](#)**

**[EntitySpaces 2010 Better Handling of Schemas](#)**

**[EntitySpaces 2010 Release Schedule](#)**