LINQ-to-SQL и контенты QP7.Framework

Решаемые задачи

Работа с контентами упрощается за счет того, что:

- Доступ к полям становится типизированным, с поддержкой *IntelliSense*.
- **LINQ-to-SQL** предоставляет большое количество встроенных операций для работы с наборами данных. Единицами данных в этих наборах в нашем случае будут являться статьи.
- Становится доступным поддержка декларативного связывания контентов с контролами (по DataSourceID) с помощью стандартного класса LinqDataSource.
- Добавляется возможность редактирования статей с помощью LINQ-классов. Что вместе с предыдущим пунктом дает возможность использования, например, редактируемого Grid для контентов QP7.
- Связанные таблицы становятся доступными как свойства (включая связи многие-ко-многим).

Предварительная настройка

- В конфигурационный файл Backend нужно добавить параметр **SqlMetalPath**, определяющий путь к стандартной утилите генерации SQL-классов. Данная утилита устанавливатся вместе с Visual Studio 2008. Путь к ней может выглядеть так: `C:\Program Files\Microsoft SDKs\Windows\v6.0A\bin\sqlmetal.exe`. Если на целевом компьютере не установлена Visual Studio, утилиту можно просто скачать.
- Необходимо дать право доступа *Modify* пользователю, под которым запущен Application Pool (обычно это NETWORK SERVICE), на папку *App_Code* для возможности генерации файлов классов и папку *App_Data* для генерации промежуточных файлов отображений.
- В процессе работы утилита **SQLMetal** может выдавать ошибки, которые можно посмотреть в файле *sqlmetal.log* в папке *App_Data*. Ошибки обычно связаны с конфликтами имен.
- Если на сайте не использовался LINQ-to-SQL, то необходимо добавить в *web.config* ссылки на соответствующие сборки:

```
<system.web>
    <compilation>
        <assemblies>
...
            <add assembly="System.Data.Linq, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=B77A5C561934E089" />
            <add assembly="System.Data.DataSetExtensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=B77A5C561934E089" />
            <add assembly="System.Core, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
        </assemblies>
...
    </compilation>
```

```
</system.web>
...
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs" warningLevel="4"
type="Microsoft.CSharp.CSharpCodeProvider, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
        <providerOption name="CompilerVersion" value="v3.5" />
        <providerOption name="WarnAsError" value="false" />
      </compiler>
    </compilers>
  </system.codedom>
```

Настройка сборки контентов в классы LINQ-to-SQL

Сборка контентов QP7 в классы *LINQ-to-SQL* может осуществляться в двух режимах:

1. С использованием БД. При этом будут использованы специальные настройки LINQ-to-SQL, которые задаются в свойствах сайта, контентов и полей.
2. С использованием пользовательского файла отображения. При этом файл может быть как составлен разработчиком вручную, так и сгенерирован бэкендом по умолчанию.

Переключение режимов осуществляется опцией **Использовать прямое отображение из базы данных** (**Use direct mapping from database**) на уровне сайта. Сама сборка запускается кнопкой **Собрать Контенты** (**Assemble Contents**) на странице **Контенты** (**Contents**). При этом в папке *App_Code* происходит полная перегенерация всех LINQ-классов текущего сайта. Следует иметь в виду, что данный процесс с большой долей вероятности приведет к перегрузке домена приложения, поэтому его стоит с осторожностью применять на продукционных сайтах.

Кроме этого, на уровне сайта существует опция **Импортировать файл отображения в базу данных**(**Import mapping file to database**), которая позволяет импортировать существующий пользовательский файл отображения в базу данных для последующей настройки через бэкенд QP7 и генерации файла отображения в автоматическом режиме. При успешном выполнении импорта данная опция автоматически сбрасывается, а опция **Использовать прямое отображение из базы данных** (**Use direct mapping from database**) автоматически устанавливается.

Опция **Импортировать файл отображения в базу данных**(**Import mapping file to database**) также может быть полезна для первичного заполнения настроек LINQ-to-SQL для существующего сайта с большим количеством контентов. Для этого сначала генерируется пользовательский файл отображения по умолчанию (при отключенной опции **Использовать прямое отображение из базы данных** (**Use direct mapping from database**), а затем проводится его импорт.

Генерация LINQ-to-SQL классов и вспомогательных файлов осуществляется в папки *App_Data* и *App_Code* сайта. Расположение этих папок вычисляется из расположения папки *bin*, которая настраивается в свойствах сайта. При этом предполагается, что папки *App_Data* и *App_Code* находятся на том же уровне в дереве папок, что и *bin*.

Использование сгенерированных LINQ-to-SQL классов

Пример добавления статьи

```
    NewsArticle art = new NewsArticle();
        art.Title = "some title";
    art.Text = "some text";
    LinqHelper.Context.NewsArticles.InsertOnSubmit(art);
    LinqHelper.Context.SubmitChanges();
```