

FluentNHibernate Conventions III

2011년 12월 17일 토요일

오후 3:23

FluentNHibernate Conventions 의 마지막 시리즈입니다. 이번에는 아주 특수한 상황에 대한 대처입니다.

1. Oracle 에서는 버전에 따라 엔티티의 명칭의 길이가 제한됩니다. (최대 24자 ~ 30자입니다)
2. RDBMS에서는 클래스나 속성명과 같이 풀어쓰는 명칭보다는 약어가 더 의미있거나, 기존 DB가 약어로 된 경우가 많습니다.

위의 두 경우를 해결하기 위해서 몇 가지 해결책이 있겠지만, 가장 손쉬운 방법은 약어(Abbreviation)을 이용하여, 길이를 줄일 수 있고, RDBMS의 기존 약어를 클래스나 속성명으로 풀어 쓸 수 있게 하면 가장 좋습니다.

예를 들면 다음과 같이 Object 영역과 RDBMS 영역 간의 상호 매핑을 약어 사전을 이용한다면, 상당히 많은 코드 및 "Magic String" 을 없앨 수 있습니다.

1. Department (클래스) <=> DEPT (테이블)
2. Department.Id <=> DEPT_ID (컬럼명)
3. IX_Department_Company <=> IX_DEPT_CMP (인덱스나 Foreign Key 명)

사실 아주 간단하게 약어 사전 (Abbreviation Dictionary)에 정보를 지정하고, 그 정보를 바탕으로, 실제 매핑 시에 약어로 변환할 수 있도록 해주기만 합니다^^

자 이제 실제 구현을 어떻게 하나 봅시다.

```
namespace RCL.Data.NH.Fluents
{
    public interface IFluentConvention : IConvention
    {
        /// <summary>
        /// Convention Option
        /// </summary>
        ConventionOptions Options { get; set; }
        /// <summary>
        /// 컬럼명을 클래스명 + 속성명으로 표현되도록 하는 속성명의 컬렉션입니다.
        /// 예: Code 는 EntityName + Code (예: Company.Code 는 컬럼명이 CompanyCode | COMPANY_CODE 로 변환된다)
        /// </summary>
        IList<string> PropertyWithClassName { get; set; }
        /// <summary>
        /// 컬럼명으로 매핑시에 약어로 매핑해야 할 이름(단어) 매핑이다. (예: Department-Dept, Locale-Loc, Configuration-Conf 등)
        /// </summary>
        IDictionary<string, string> AbbrNameMap { get; }
        /// <summary>
        /// <paramref name="text"/>에 <see cref="AbbrNameMap"/>에 등록된 약어 변환 단어가 있다면 약어로 변환하여 반환합니다.
        /// </summary>
        /// <param name="text">원본 데이터</param>
        /// <returns></returns>
        string GetAbbrName(string text);
    }
}
```

보시다시피 AbbrNameMap 에 약어 사전을 등록할 수 있도록 Dictionary 를 제공하고, 필요할 때 이 값을 이용하여 약어로 변환하기만 하면 됩니다.

```
/// <summary>
/// FluentNHibernate 의 Convention 기능을 활용하기 위한 기본 클래스입니다.
/// </summary>
public class FluentConventionBase : IFluentConvention, IClassConvention
```

```

{
    private IList<string> _propertyWithClassName = new List<string>();

    protected FluentConventionBase() : this(ConventionOptions.Default, null) {}
    protected FluentConventionBase(ConventionOptions options) : this(options, null) {}
    protected FluentConventionBase(ConventionOptions options, IEnumerable<string> propertyWithClassName)
        : this(options, propertyWithClassName, null) {}
    protected FluentConventionBase(ConventionOptions options,
                                    IEnumerable<string> propertyWithClassName,
                                    IDictionary<string, string> abbrNameMap)
    {
        Options = options;
        if(propertyWithClassName != null)
            _propertyWithClassName = new List<string>(propertyWithClassName);

        if(abbrNameMap != null)
        {
            MappingContext.AbbbrNameMap = abbrNameMap;
        }
    }

    /// <summary>
    /// Convention Option
    /// </summary>
    public ConventionOptions Options { get; set; }

    /// <summary>
    /// 속성명과 컬럼명을 같은 값으로 유지하고자 하는 속성명 (Description, ExAttr 같은 것)
    /// 그 외의 것은 컬럼명이 EntityName + '_' + PropertyName 으로 매핑됩니다. (예: Company.Name 은 COMPANY_NAME)
    /// </summary>
    public IList<string> PropertyWithClassName
    {
        get { return _propertyWithClassName ?? (_propertyWithClassName = new List<string>()); }
        set { _propertyWithClassName = value; }
    }

    /// <summary>
    /// 컬럼명으로 매핑시에 약어로 매핑해야 할 이름(단어) 매핑이다. (예: Department-Dept, Locale-Loc, Configuration-Conf)
    /// </summary>
    public IDictionary<string, string> AbbrNameMap
    {
        get { return MappingContext.AbbbrNameMap; }
    }

    /// <summary>
    /// Apply changes to the target
    /// </summary>
    public virtual void Apply(IClassInstance instance)
    {
        if(Options.DefaultLazy)
            instance.LazyLoad();

        if(Options.DynamicInsert)
            instance.DynamicInsert();

        if(Options.DynamicUpdate)
            instance.DynamicUpdate();
    }

    /// <summary>
    /// <paramref name="text"/>에 <see cref="AbbrNameMap"/>에 등록된 약어 변환 단어가 있다면 약어로 변환하여 반환합니다.
    /// </summary>
    /// <param name="text">원본 문자열</param>
    /// <returns></returns>
    public string GetAbbrName(string text)
    {
        return MappingContext.ToAbbrName(text);
    }
}

```

FluentConventionBase, GetAbbrName은 문자열 중에 약어사전에 해당하는 문자열을 약어로 변환해주는 함수입니다. MappingContext 클래스의 관련 코드는 다음과 같다.

```

/// <summary>
/// 컬럼명으로 매핑시에 약어로 매핑해야 할 이름(단어) 매핑이다. (예: Department-Dept, Locale-Loc, Configuration-Conf 등)
/// </summary>
public static IDictionary<string, string> AbbrNameMap
{

```

```

    get { return _abbrNameMap ?? (_abbrNameMap = new Dictionary<string, string>()); }
    set { _abbrNameMap = value; }
}
/// <summary>
/// <paramref name="text"/>에 <see cref="AbbrNameMap"/>에 등록된 약어 변환 단어가 있다면 약어로 변환하여 반환합니다.
/// </summary>
/// <param name="text">원본 데이터</param>
/// <returns></returns>
public static string ToAbbrName(this string text)
{
    if(text.IsNullOrWhiteSpace())
        return string.Empty;
    var builder = new StringBuilder(text);

    foreach(var from in AbbrNameMap.Keys)
        if(text.Contains(from))
            builder.Replace(from, AbbrNameMap[from]);
    return builder.ToString();
}

```

약어 사전을 기반으로 일반 문자열을 약어로 변환한 문자열을 반환합니다.

그럼 이제 실제 활용하는 예제를 봅시다.

우선 Unit Test 시에는 FluentNHibernateFixtureBase를 상속받아 TestFixtureSetUp 시에 다음과 같이 코드를 구현하면 됩니다.

```

protected virtual void OnTestFixtureSetUp()
{
    // FluentNHibernate Conventions
    // 참고 : http://wiki.fluentnhibernate.org/Convention\_shortcut
    // 이 것을 설정 해줘야 합니다. (기본값이 Pascal 이므로 Oracle 일때만 설정해주면 됩니다) Mapping 파일에
    // MappingContext.AsNamingText() 를 사용하세요
    var abbrNameMap = new Dictionary<string, string>
    {
        { "Department", "Dept" },
        { "Locale", "Loc" },
        { "Metadata", "Meta" }
    };
    var convention = new OracleNamingConvention(null,
                                                new List<string> { "Code", "Name" },
                                                abbrNameMap);
    InitializeNHibernateAndIoC(ContainerFilePath,
                               GetDatabaseEngine(),
                               GetDatabaseName(),
                               GetMappingInfo(),
                               GetNHibernateProperties(),
                               cfg =>
                               {
                                   cfg.SetListener(NHibernate.Event.ListenerType.PreInsert, new UpdateTimestampEventListener());
                                   cfg.SetListener(NHibernate.Event.ListenerType.PreUpdate, new UpdateTimestampEventListener());
                               },
                               convention);
    CurrentContext.CreateUnitOfWork();
}

```

실제 프로그램에서는 IoC configuration 을 이용하여 다음과 설정하면 됩니다.

```

<!-- 참고: http://docs.castleproject.org/Windsor.XML-Inline-Parameters.ashx?HL=xml,windsor,dictionary -->
<component id="FluentConvention.PascalNaming"
    service="RCL.Data.NH.Flucts.IFluentConvention, RCL.Data"
    type="RCL.Data.NH.Flucts.PascalNamingConvention, RCL.Data">
    <parameters>
        <Options>${FluentNHibernate.ConventionOptions}</Options>
        <PropertyWithClassName>
            <list>
                <item>Code</item>
                <item>Name</item>
            </list>
        </PropertyWithClassName>
        <AbbrNameMap>
            <dictionary>
                <entry key="Department">Dept</entry>
                <entry key="Locale">Loc</entry>
                <entry key="Metadata">Meta</entry>
            </dictionary>
        </AbbrNameMap>
    </parameters>
</component>

```

```

        </AbbrNameMap>
    </parameters>
</component>

```

자 이제 환경 설정은 모두 끝났습니다.

자동으로 클래스명과 속성명을 테이블명과 컬럼명으로 변환해줍니다. 다만 특정적으로 매핑 시에 지정해야 할 테이블명 또는 컬럼명의 경우에는 Mapping 파일에서 지정해주어야 합니다. 자 이제 Mapping 만 잘 만들면 됩니다. 지금까지 FluentNHibernate 의 기능 중에 하나인 Convention에 대해 설명했는데, 매핑 파일 작성을 HBM으로 한다고 생각한다면, 생뚱맞은거죠^^ FluentNHibernate의 코드로 매핑을 구현하는 ClassMap을 작성할 때, 어떻게 해야 하는지 봅시다.

```

public class FDepartmentMap : ClassMap<FDepartment>
{
    public FDepartmentMap()
    {
        Id(x => x.Id).GeneratedBy.Native();
        Map(x => x.Code);
        Map(x => x.Name);
        References(x => x.Parent)
            .Access.Property()
            .Cascade.SaveUpdate()
            .Fetch.Select()
            .LazyLoad(Laziness.Proxy);
        HasMany(x => x.Children)
            .Access.CamelCaseField(Prefix.Underscore)
            .Cascade.AllDeleteOrphan()
            .Inverse()
            .LazyLoad()
            .AsSet(SortType.Natural);
        Component<TreeNodePosition>(x => x.NodePosition,
            p =>
            {
                p.Map(x => x.Order).Column("TreeOrder".ToAbbrName().AsNamingText());
                p.Map(x => x.Level).Column("TreeLevel".ToAbbrName().AsNamingText());
            });
        HasManyToMany(x => x.Users)
            .Table("FDepartmentMember".ToAbbrName().AsNamingText())
            .Cascade.AllDeleteOrphan()
            .LazyLoad()
            .AsSet();
    }
}

```

Fdepartment 클래스에 대한 매핑파일을 예로 든다면 다음과 같은 결과가 나올 것입니다.

```

create table NHibernate.dbo.F_DEPT (
    F_DEPT_ID INT IDENTITY NOT NULL,
    F_DEPT_CODE NVARCHAR(255) null,
    F_DEPT_NAME NVARCHAR(255) null,
    PARENT_ID INT null,
    TREE_ORDER INT null,
    TREE_LEVEL INT null,
    primary key (F_DEPT_ID)
)

```

우선 Oracle Naming 방식을 지정해서 나온 결과 이므로, 일반적으로 Fdepartment 는 F_DEPARTMENT 가 되지만 약어 사전에 등록된 단어라서 Fdepartment => Fdept => F_DEPT 가 됩니다.

나머지 컬럼들도 같습니다.

자 이제 매핑 파일이 HBM XML 파일보다 우월하면서, 환경설정에 따라 다양한 기능을 제공하면서도, 코드의 반복을 최대한 없애고, 유지보수를 위해서도 일관된 변환 기능을 제공하는 FluentNHibernate Convention 기능에 대해 알아보았습니다.

더 많은 내용을 알고 싶으시면 <http://wiki.fluenthibernate.org/> 에서 Conventions 섹션을 참고하세요.