

# Using AdoRepository

2009.01



# 목차

## 1. Using AdoRepository without IoC

1. Setup project
2. Setup configuration
3. Instancing IAdoRepository
4. Samples

## 2. Using AdoRepository with IoC

1. Project 구성
2. IoC Configuration 구성
3. IoC Testing
4. Instancing AdoRepository
5. Apply IoC to Web Application

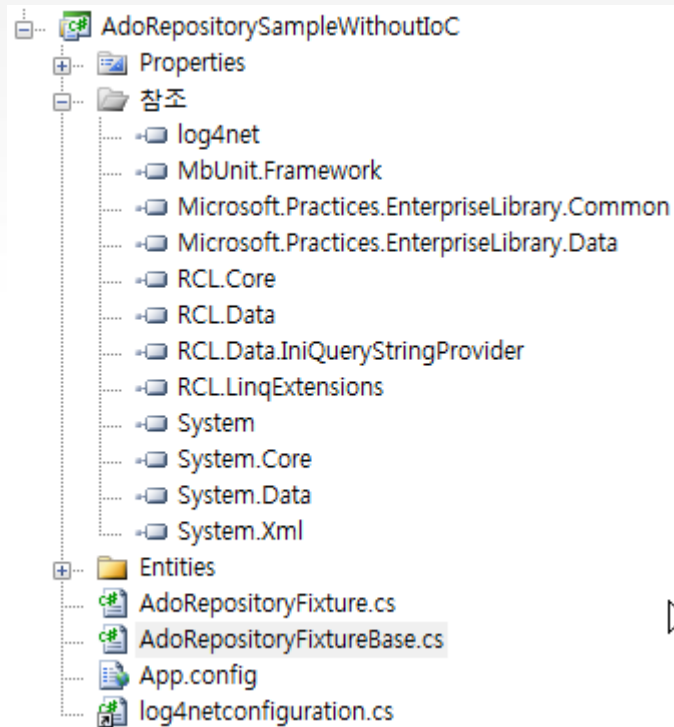


# 1. Using AdoRepository without IoC

# ▶▶▶ 1. Using AdoRepository without IoC

- ◆ IoC 를 이용하면 시스템 확장성 및 유연성이 증대되지만, 환경설정하기 위해 IoC에 대한 이해가 필수이다.
- ◆ 모든 개발자가 IoC에 대한 이해와 적용을 하면 최상이지만, 중급 이상의 개발자 정도에서 다루는 것이 효과적이다.
- ◆ IoC를 이용하지 않고, 전통적 방식으로 사용하기 위한 절차
  - 관련 Library를 Referencing 한다.
  - 시스템에서 사용할 Repository를 제공하는 클래스를 구현한다.
  - Repository 를 사용해 작업을 수행한다.

# 1.1 Setup Project without IoC



필요한 Library를 참조한다.

DAAB 사용 (필수)

Microsoft.Practices.EnterpriseLibrary.Common.dll

Microsoft.Practices.EnterpriseLibrary.Data.dll

AdoRepository 사용 (필수)

RCL.Data.dll

QueryStringProvider 사용 (Options)

RCL.Data.IniQueryStringProvider.dll

Nini.dll

Core Library (Options)

RCL.Core.dll

LINQ Extensions (Options)

RCL.LinqExtensions.dll

# 1.1 Setup configuration without IoC

환경설정은 DAAB만 사용할 때와 동일하다. DAAB용 configuration section 을 등록하고 dataConfiguration element 를 추가하면 된다.

```
2 <configuration>
3   <configSections>
4     <!-- Microsoft Enterprise Library Data Access Application Block 3.1 or higher -->
5     <section name="dataConfiguration"
6       type="Microsoft.Practices.EnterpriseLibrary.Data.Configuration.DatabaseSettings,
7         Microsoft.Practices.EnterpriseLibrary.Data,
8         Version=3.1.0.0,
9         Culture=neutral,
10        PublicKeyToken=b03f5f7f11d50a3a"/>
11
12     <!-- logging -->
13     <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler,log4net" />
14   </configSections>
15
16
17   <!-- DAAB 사용을 위해 -->
18   <dataConfiguration defaultDatabase="Northwind"/>
19   <connectionStrings>
20     <add name="Northwind"
21       connectionString="Data Source=(local)\SQLEXPRESS;Initial Catalog=Northwind;Integrated Security=True;Enlist=false"
22       providerName="System.Data.SqlClient"/>
23     <add name="Pubs"
24       connectionString="Data Source=(local)\SQLEXPRESS;Initial Catalog=Pubs;Integrated Security=True;Enlist=false"
25       providerName="System.Data.SqlClient"/>
26   </connectionStrings>
```

## 1.3 Instantiating IAdoRepository without IoC

```
private static IAdoRepository _defaultAdoRepository;

/// <summary>
/// DAAB의 기본 Database에 대한 IAdoRepository
/// </summary>
public static IAdoRepository DefaultAdoRepository
{
    get
    {
        return _defaultAdoRepository
            = _defaultAdoRepository
                ?? AdoRepositoryFactory.Instance.CreateRepository();
    }
}

private static IAdoRepository _northwindRepository;
/// <summary>
/// Northwind DB에 대한 IAdoRepository
/// </summary>
public static IAdoRepository NorthwindRepository
{
    get
    {
        return _northwindRepository =
            _northwindRepository
                ?? AdoRepositoryFactory.Instance.CreateRepository("Northwi");
    }
}
```

DAAB의 DatabaseFactory와 같은 기능을 수행하는  
AdoRepositoryFactory를 이용하여 Repository를 생성한다.

```
// Create Instance of IAdoRepository with default database name
AdoRepositoryFactory.CreateRepository();
```

```
// Create Instance of IAdoRepository with dbName
AdoRepositoryFactory.CreateRepository(string dbName);
```

환경설정에서 설정한 기본 DB에 대한 Repository 는 코드상에서  
DefaultAdoRepository 이고,  
명시적으로 Northwind DB를 사용하려면 NorthwindRepository 처럼 제공  
하면 된다.



## 1.4 Samples without IoC

```
public void ExecuteProcedure()
{
    IAdoParameter[] outputs
        = NorthwindRepository.ExecuteProcedure("CustOrderHist",
                                                base.CustomerTestParameter);

    if (log.IsDebugEnabled)
        outputs.ForEach(output => log.Debug(output.ObjectToString()));
}
```

CustOrderHist 라는 SP를 실행하고, Input이 아닌 Parameter 정보를 반환한다.

```
public void ExecuteReaderWithPaging(int pageIndex, int pageSize)
{
    using (DbCommand cmd = NorthwindRepository.GetCommand(OrderDetailSql))
    {
        var orderDetails
            = NorthwindRepository.ExecuteReader<OrderDetail>(TrimMapper, cmd, pageIndex, pageSize);

        Console.WriteLine(orderDetails);
    }
}

using (DataTable dt
    = NorthwindRepository.ExecuteDataTableByProcedure(
        "CustOrderHist",
        firstResult,
        maxResults,
        base.CustomerTestParameter))
{
    Assert.IsFalse(dt.HasErrors);
    Assert.GreaterThan(dt.Rows.Count, 0);
}
```

DB로부터 수행 결과를 IDataReader로 받고, 매핑을 통해 OrderDetail 의 인스턴스를 만들어 반환한다. Paging 기능을 이용할 수 있다.

Procedure "CustOrderHist" 를 실행해서 결과를 페이징을 수행하여 원하는 구간의 결과셋만 DataTable의 레코드로 추가한다.





- 2.1 SetUp Project
- 2.2 Setup Configuration (Application, IoC)
- 2.3 IoC Testing
- 2.4 Instancing AdoRepository with IoC
- 2.5 Apply IoC to Web Application

## 2. Using Adorepository with IoC

## 2. Using AdoRepository with IoC

- ◆ IoC를 이용해 다양한 로직이 추가된 IAdoRepository의 Concrete Class를 사용할 수 있다.
- ◆ IoC 와 Decorator pattern을 이용해, 부가적인 기능을 AdoRepository에 부여할 수 있고, 상황에 따라 코드 수정 없이 변경할 수 있다.
- ◆ IoC Container의 Component에 대한 Lifecycle을 조절 할 수 있다.

## 2.1 Setup Project with IoC

1. 프로젝트를 생성한다.
2. 필요 Library를 참조한다.

DAAB 사용 (필수)

Microsoft.Practices.EnterpriseLibrary.Common.dll

Microsoft.Practices.EnterpriseLibrary.Data.dll

IoC 사용 (필수)

Castle.Core.dll

Castle.MicroKernel.dll

Castle.Windsor.dll

AdoRepository 사용 (필수)

RCL.Data.dll

QueryStringProvider 사용 (Options)

RCL.Data.IniQueryStringProvider.dll

Nini.dll

Core Library (Options)

RCL.Core.dll

LINQ Extensions (Options)

RCL.LinqExtensions.dll

## 2.2 Setup configuration with IoC

1. App.Config (or Web.Config) 의 ConfigSection 에 아래와 같이 DAAB와 Castle 관련 section을 추가한다.

```
3 <configSections>
4   <!-- DAAB -->
5   <section name="dataConfiguration"
6       type="Microsoft.Practices.EnterpriseLibrary.Data.Configuration.DatabaseSettings,
7           Microsoft.Practices.EnterpriseLibrary.Data,
8           Version=3.1.0.0,
9           Culture=neutral,
10          PublicKeyToken=b03f5f7f11d50a3a"/>
11
12   <!-- Castle.Windsor for IoC -->
13   <section name="castle" type="Castle.Windsor.Configuration.AppDomain.CastleSectionHandler, Castle.Windsor" />
14
15   <section name="log4net"
16       type="log4net.Config.Log4NetConfigurationSectionHandler,log4net" />
```

2. Castle 관련 환경설정 파일 위치를 지정한다.

```
93 <!-- Castle Windsor IoC 를 사용합니다. -->
94 <castle>
95   <include uri="file://Windsor.config" />
96 </castle>
97
98 <!-- DAAB 사용을 위해 -->
99 <dataConfiguration defaultDatabase="Northwind"/>
100 <connectionStrings>
101   <add name="Northwind"
102       connectionString="Data Source=(local)\SQLEXPRESS;Initial Catalog=Northwind;Integrated Security=True;Enlist=True"
103       providerName="System.Data.SqlClient"/>
104   <add name="Pubs"
105       connectionString="Data Source=(local)\SQLEXPRESS;Initial Catalog=Pubs;Integrated Security=True;Enlist=True"
106       providerName="System.Data.SqlClient"/>
107 </connectionStrings>
```

## 2.2 Setup IoC configuration

```
11 <!-- Castle.Windsor Component configuration -->
12 <components>
13
14 <component id="NorthwindAdoRepository"
15     service="RCL.Data.Ado.IAdoRepository, RCL.Data"
16     type="RCL.Data.Ado.AdoRepositoryImpl, RCL.Data">
17     <parameters>
18         <!-- dbName을 설정하지 않으면 DAAB의 기본 DB 사용 -->
19         <!--<dbName>Northwind</dbName>-->
20         <QueryStringProvider>${IniAdoQueryStringProvider.Northwind}</QueryStringProvider>
21     </parameters>
22 </component>
23
24 <component id="PubsAdoRepository"
25     service="RCL.Data.Ado.IAdoRepository, RCL.Data"
26     type="RCL.Data.Ado.AdoRepositoryImpl, RCL.Data">
27     <parameters>
28         <dbName>Pubs</dbName>
29         <QueryStringProvider>${IniAdoQueryStringProvider.Pubs}</QueryStringProvider>
30     </parameters>
31 </component>
32
33 <component id="IniAdoQueryStringProvider.Northwind"
34     service="RCL.Data.IQueryStringProvider, RCL.Data"
35     type="RCL.Data.IniQueryStringProvider.IniAdoQueryStringProvider, RCL.Data.IniQueryStringProvider">
36     <parameters>
37         <queryFilePath>.\QueryFiles\Northwind.Ado.MsSql.ini</queryFilePath>
38     </parameters>
39 </component>
40
41 <component id="IniAdoQueryStringProvider.Pubs"
42     service="RCL.Data.IQueryStringProvider, RCL.Data"
43     type="RCL.Data.IniQueryStringProvider.IniAdoQueryStringProvider, RCL.Data.IniQueryStringProvider">
44     <parameters>
45         <queryFilePath>.\QueryFiles\Pubs.Ado.MsSql.ini</queryFilePath>
46     </parameters>
47 </component>
48 </components>
```

Northwind DB용 AdoRepository를 정의한다.  
QueryStringProvider를 사용한다면 parameters에 추가한다.

Pubs DB용 AdoRepository를 정의한다.  
Database connectionString name을 명시적으로 parameter에 정의한다.

Northwind DB용 IQueryStringProvider Component를 등록한다.  
실제 파일경로를 정의해 주어야 한다. (Options)

Pubs DB용 IQueryStringProvider Component를 등록한다.  
실제 파일경로를 정의해 주어야 한다. (Options)

## 2.3 Testing IoC

IoC 사용 시 주의할 점 중에 하나가 IoC를 초기화를 수행해야 한다는 점이다.

Web Application 같은 경우 HttpApplication을 상속 받아 Application Start Event에서 IoC 를 초기화를 수행하면 된다.

```
8 public abstract class DataTestFixtureBase
9 {
10     << log4net >>
16
17     [FixtureSetUp]
18     public void ClassSetUp()
19     {
20         if (IoC.IsInitialized == false)
21             IoC.Initialize(new RwContainer(new XmlInterpreter()));
22     }
23
24     [FixtureTearDown]
25     public void ClassCleanUp()
26     {
27         IoC.Initialize(null);
28     }
29 }
```

Application 시작시 IoC 환경설정 분석을 위해, 명시적으로 초기화를 수행한다.

단순히 IoC.Initialize() 만 호출해도 된다.

## 2.3 Testing IoC

```
[Test]
public void IoCTest()
{
    WindsorContainer container = new WindsorContainer();
    container.AddComponent("AdoRepository", typeof(RCL.Data.Ado.IAdoRepository),
        typeof(RCL.Data.Ado.AdoRepositoryImpl));
}

[Test]
public void ResolveAdoRepository()
{
    IAdoRepository repository = IoC.Resolve<IAdoRepository>();
    Assert.IsNotNull(repository);

    Console.WriteLine(repository.GetType().FullName);
}
```

수동으로 AdoRepository 를 등록해 본다.

아주 특별한 상황에서는 이와 같이 특정 Class를 사용할 때  
도 있지만, 사용할 기회가 없을 듯...

IoC Container로부터 IAdoRepository 서비스를 제공하는  
인스턴스를 반환받는다.

결과는 아래 ConsoleOutput에 있듯이  
AdoRepositoryImpl 이다.

☐ **ResolveAdoRepository** ■  
Duration: 0.04s, Assertions: 1.

ConsoleOutput  
RCL.Data.Ado.AdoRepositoryImpl



## »»» 2.4 Instanting AdoRepository with IoC

```
public static IAdoRepository DefaultAdoRepository
{
    get { return IoC.TryResolve<IAdoRepository>(AdoRepositoryFactory.Instance.CreateRepository()); }
}

public static IAdoRepository NorthwindAdoRepository
{
    get { return IoC.Resolve<IAdoRepository>("NorthwindAdoRepository"); }
}
```

환경설정에 정의된 기본 Database를 사용하는 DefaultAdoRepository는 IoC.TryResolve 를 수행해서 제공한다.

특정 DB에 대해, 특정 Component를 사용하려면 IoC.Resolve<IAdoRepository>(string componentId) 를 사용하여 구현할 수 있다.

## 2.5 Apply IoC to Web Application

- ◆ IoC 사용시에는 Web Application 시작 시에 사용자 코드가 실행되기 전에 초기화가 되어 있어야 한다.
- ◆ 초기화를 PageBase에서 수행하는 것은 의미 없다. Application 에서 한번만 수행하면 되기 때문이다.
- ◆ IoCHttpApplication 상속 받기
  - Application 시작 시 초기화여부를 검사하여 초기화를 한번만 수행한다. 구현이 쉽다.
  - DNN 등 이미 다른 HttpApplication을 상속 받은 App는 적용할 수 없다.
- ◆ IoCHttpModule을 사용하기
  - HttpModule에 등록하여, 모듈이 초기화 될 때 IoC 초기화를 수행한다.
  - 설정작업이 조금 어렵지만, 확장성이 뛰어나다.

## 2.5 Apply IoC to Web Application

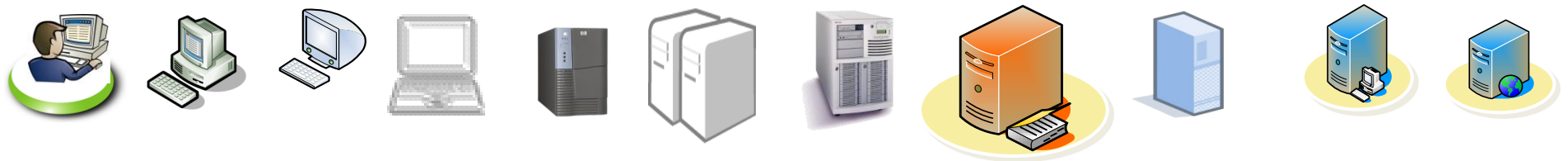
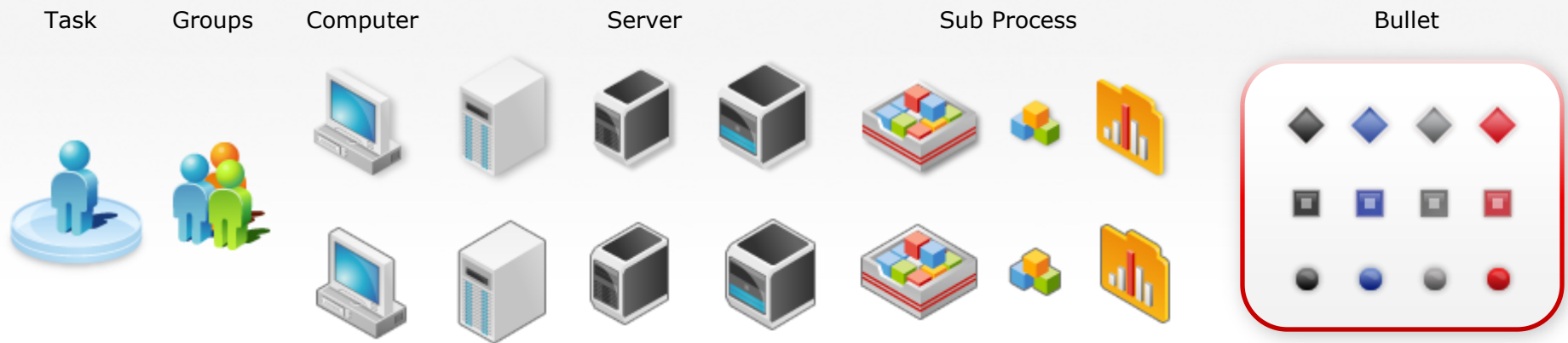
global.asax 에서 IoCApplication을 상속 받기

```
1  <%@ Application Language="C#" Inherits="RCL.CastleIoC.Web.IoCApplication" %>
2  <script RunAt="server">
3
4      // IoC 관련 처리는 IoCApplication 에서 수행한다.
5      public void Application_Start(object sender, EventArgs e)
6      {
7          // 제일 먼저 해줘야 로그를 사용할 수 있다.
8          log4net.Config.XmlConfigurator.Configure();
9      }
10
11     public void Application_End(object sender, EventArgs e)
12     {
13         log4net.LogManager.Shutdown();
14     }
15 </script>
16
```

system.web/httpModules에 IoCModule 등록하기

```
96 <httpModules>
97   <add name="ScriptModule" type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0,
98   <!-- Page 처리 성능 측정을 위한 모듈 -->
99   <add name="PagePerformanceModule" type="RCL.Web.HttpModules.PagePerformanceModule, RCL.Web"/>
100  <!-- IoC 초기화를 위한 모듈-->
101  <add name="IoCModule" type="RCL.CastleIoC.Web.IoCModule, RCL.CastleIoC"/>
102 </httpModules>
103 </system.web>
```

## 2. ICON SET



# 감사합니다