

之前陆陆续续的实现了LexicalAnalysis,DynamicDataSet以及ExpressionAnalysis的功能，今天就来讲讲如何利用它们来为项目添加一个脚本功能。

有时我们需要在项目中动态的执行一些简单的条件判断，比如在写一个自动测试的框架的时候，我们必须支持用户可以手动的写一些Check脚本。比如说

我们完成了一个测试用例，测试完成后我们从数据库中取到了很多的这个测试用例相关的表的数据，其中有的数据是在这个测试用例中创建的，也有的数据

是在这个测试中被修改的，还有的数据是被删除的，因此为了验证这个测试用例的结果我们必须写很多小的断言，如果能支持脚本那写这些断言就很简单了。

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Diagnostics.Contracts;
using Zxf.ExpressionBuilder;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            dynamic dynamicDataSet = new DynamicDataSet(PrepareDataSet());

            var dynamicLambdaExpFunc1 = BuildFunc("Data.Test.Message[0].ToString().StartsWith(\"Hello\")", dynam
            var result1 = dynamicLambdaExpFunc1.Invoke(dynamicDataSet);
            Contract.Assert((bool)result1 == true);

            var dynamicLambdaExpFunc2 = BuildFunc("Data.Test.Message[0].ToString().ToUpper()", dynamicDataSet);
            var result2 = dynamicLambdaExpFunc2.Invoke(dynamicDataSet);
            Contract.Assert((string)result2 == "Hello, World!".ToUpper());

            var dynamicLambdaExpFunc3 = BuildFunc("Data.Test.Count == 1", dynamicDataSet);
            var result3 = dynamicLambdaExpFunc3.Invoke(dynamicDataSet);
            Contract.Assert((bool)result3 == true);
        }

        private static Func<DynamicDataSet, object> BuildFunc(string code, DynamicDataSet parameter)
        {
            List<LexicalBlock> lexicalBlocks = new LexicalAnalysis().Analysis(new string[] { code });

            Dictionary<string, Type> parameterTypes = new Dictionary<string, Type>();
            parameterTypes.Add("Data", typeof(DynamicDataSet));
            return ExpressionAnalysis.Analysis<Func<DynamicDataSet, object>>(lexicalBlocks, parameterTypes);
        }

        private static DataSet PrepareDataSet()
        {
            DataTable dataTable = new DataTable("Test");
            dataTable.Columns.Add("Message", typeof(string));
            DataRow dataRow = dataTable.NewRow();
            dataRow[0] = "Hello, World!";
            dataTable.Rows.Add(dataRow);
            DataSet dataSet = new DataSet();
            dataSet.Tables.Add(dataTable);
            return dataSet;
        }
    }
}

```