

# Reactive Extensions (Rx) 入门(2) —— 安装 Reactive Extensions

 [blog.csdn.net/fangxing80/article/details/7581937](http://blog.csdn.net/fangxing80/article/details/7581937)

原文：[http://www.atmarkit.co.jp/fdotnet/introrx/introrx\\_01/introrx\\_01\\_02.html](http://www.atmarkit.co.jp/fdotnet/introrx/introrx_01/introrx_01_02.html)

作者：河合 宜文

## 安装方法

关于 Rx 的安装，可以通过 Reactive Extensions (Rx) 的主页 的直接下载安装，当然也可以利用 NuGet 导入 dll (<http://nuget.org/packages/Rx-Main>) [译注: NuGet 是VS集成的在线dll部署工具，非常方便]。Rx 不管是 Installer 还是 NuGet 安装的，都有 Stable(稳定版) 和 Experimental(实验版) 两种。

Stable版是API中方法名等不会再变更而Experimental版则是在Stable版中加入了一些实验性的方法等。例如：C# 5.0中新增特性await/async关键字、Expand方法等只在Experimental版中有。只在Experimental版中有的方法会以[ExperimentalAttribute]特性标识，在Visual Studio中通过metadata的表示可以区分出来。因此，在利用Experimental版的时候，这些加上[ExperimentalAttribute]特性标识的方法将来方法名，参数列表还是有可能发生变化的，需要留意。

另外，Experimental版在现在Developer Preview(开发者预览版)中的.NET Framework 4.5 和WinRT(Windows 8 运行环境)中已经包含了。C#5.0中特别是针对异步编程比较大的变化策略，已经在反映在Experimental版中。

## Assembly分类

Rx的Assembly按照如下功能划分。

程序集名	NuGet Package名 (*1)	功能
System.Reactive	Rx-Main	基本Core API
System.Reactive.Windows.Forms	Rx- WinForms	WindowsForm中使用的 ControlScheduler类 (*2) 和扩展方法
System.Reactive.Windows.Threading	Rx-WPF Rx-Silverlight	WPF / Silverlight中使用的 DispatcherScheduler类 (*2) 和扩展方法

System.Reactive.Providers	Rx-Providers	IObservable接口和其他类库(*3)
Microsoft.Reactive.Testing	Rx-Testing	UnitTest和Mockup

Rx的Assembly一览

Rx的Assembly文件、一般放在「C:\Program Files\Microsoft Reactive Extensions SDK」(32bit版), 而64bit版则是放在「C:\Program Files (x86)\Microsoft Reactive Extensions SDK」)

- \*1 一览中提到的Package是指Stable版, Experimental版的时候则带有「\_Experimental」比如: 「Rx\_Experimental-Main」
- \*2 Scheduler 是基于Rx的时间和线程管理的类。ControlScheduler类是对于WinForm及控件进行BeginInvoke调用, DispatcherScheduler类是对Dispatcher进行BeginInvoke。
- \*3 「IObservable」接口名取自「Queryable Observable」。表达式树可以生成一个Observable对象。IObservable接口和IObservable接口的关系就和IEnumerable接口和IQueryable接口关系相当。

作为Core必须引用 System.Reactive 程序集, 除此之外可以按需添加引用, 比如: 如果要使用WPF, 那么还要引用 System.Reactive.Windows.Threading。  
Windows Phone 7 SDK 里集成的标准的 Rx 程序集则和上面说的有些不同:

程序集名	功能
System.Observable	IObservable<T> / IObserved<T>接口
Microsoft.Phone.Reactive	Core API和DispatcherScheduler类

Windows Phone 7 SDK的程序集一览

Windows Phone 7 里也可以通过直接引用 Data Developer Center 里发布的程序集来使用 Rx, 但是注意一下两者的区别, Data Developer Center 版是为了以后, API 有些是不同的。这些差异比如: 命名空间不同, 类名也有些不同, 方法的参数列表也不同, 有增加的方法也有删除的方法。总的来说, Data Developer Center 版应该提供了高性能和更强大功能的。  
如果更加重视是否使用标准程序集且不需要额外引用程序集, 那么直接使用 Microsoft.Phone.Reactive; 如果希望能在 WPF, Sliverlight 中移植, 那么就选择 Data Developer Center 版本好了。

相关的库

Javascript 版本并不包含在Install 中，但可以从 Rx 主页最下方下载 ZIP 得到。库的主体是 Rx.js ， 除此之外还有 jQuery 和 prototype.js。

命名空间分类

Rx 主要是以扩展方法为中心的，如果没有正确的使用 using/Import 导入命名空间则无法使用。下面例举主要的命名空间以及代表的功能和类。

名前空間	功能点
System	Subscribe订阅 IObservable 接口使用的 Lambda表达式的扩展方法
System.Reactive	主要接口和一些基本类
System.Reactive.Concurrency	实现 IScheduler 接口的一些类
System.Reactive.Disposables	实现 IDisposable 接口的一些类
System.Reactive.Joins	利用 Observable.And / Observable.Then 方法进行Join操作的类
System.Reactive.Linq	Rx 主要的命名空间，Observable 类和 IObservable接口的一些扩展方法
System.Reactive.Subjects	实现 ISubject 接口(实现 Rx 语义的事件)的一些类
System.Reactive.Threading.Tasks	Task 相互转换的扩展方法



Rx 的命名空间和代表性的功能以及类

另外，Windows Phone 中，Microsoft.Phone.Reactive 程序集里的 Rx 全部在 Microsoft.Phone.Reactive 命名空间下。

