

عنوان:	آموزش #1 Prism
نویسنده:	مسعود پاکدل
تاریخ:	۸:۱۵ ۱۳۹۲/۰۴/۰۱
آدرس:	www.dotnettips.info
گروه‌ها:	MVVM, Silverlight, WPF, prism

امروزه تقریباً تمام کسانی که پروژه‌های WPF یا Silverlight رو توسعه می‌دهند با مدل برنامه نویسی MVVM آشنایی دارند. فریم ورک‌های مختلفی برای توسعه پروژه‌ها به صورت MVVM وجود دارد. نظیر:

MVVM Light
Prism
Caliburn
Cinch
WAF
Catel
Onyx
MVVM helpers

...و

هر کدام از فریم ورک‌های بالا مزایا، معایب و طرفداران خاص خودشون رو دارند (^) ولی به جرات می‌تونیم Prism رو به عنوان قوی‌ترین فریم ورک برای پیاده سازی پروژه‌های بزرگ و قوی و ماژولار با تکنولوژی WPF یا Silverlight بنامیم. در این پست به معرفی و بررسی مفاهیم اولیه Prism خواهیم پرداخت و در پست‌های دیگر به پیاده سازی عملی همراه با مثال می‌پردازیم.

*اگر به هر دلیلی مایل به یادگیری و استفاده از Prism نیستید، بهتون پیشنهاد می‌کنم از WAF استفاده کنید.

پیش نیازها:

برای یادگیری PRISM ابتدا باید با مفاهیم زیر در WPF یا Silverlight آشنایی داشته باشید. (فرض بر این است که به UserControl و Xaml و Dependency Properties، تسلط کامل دارید)

Data binding

Resources

Commands

Behaviors

چرا Prism ؟

Prism به صورت کامل از Modular Programming برای پروژه‌های WPF و Silverlight پشتیبانی می‌کند*

از Prism هم می‌توانیم در پروژه‌های WPF استفاده کنیم و هم Silverlight.

Prism به صورت کامل از الگوی MVVM برای پیاده سازی پروژه‌ها پشتیبانی می‌کند.

پیاده سازی مفاهیمی نظیر Composite Command و Command Behavior و Asynchronous Interacion به راحتی در Prism امکان پذیر است.

مفاهیم تزریق وابستگی به صورت توکار در Prism فراهم است که برای پیاده سازی این مفاهیم به طور پیش فرض امکان استفاده از UnityContainer و MEF در Prism تدارک دیده شده است.

پیاده سازی Region navigation در Prism به راحتی امکان پذیر است.

به وسیله امکان Event Aggregation به راحتی می توانیم بین ماژول های مختلف ارتباط برقرار کنیم.

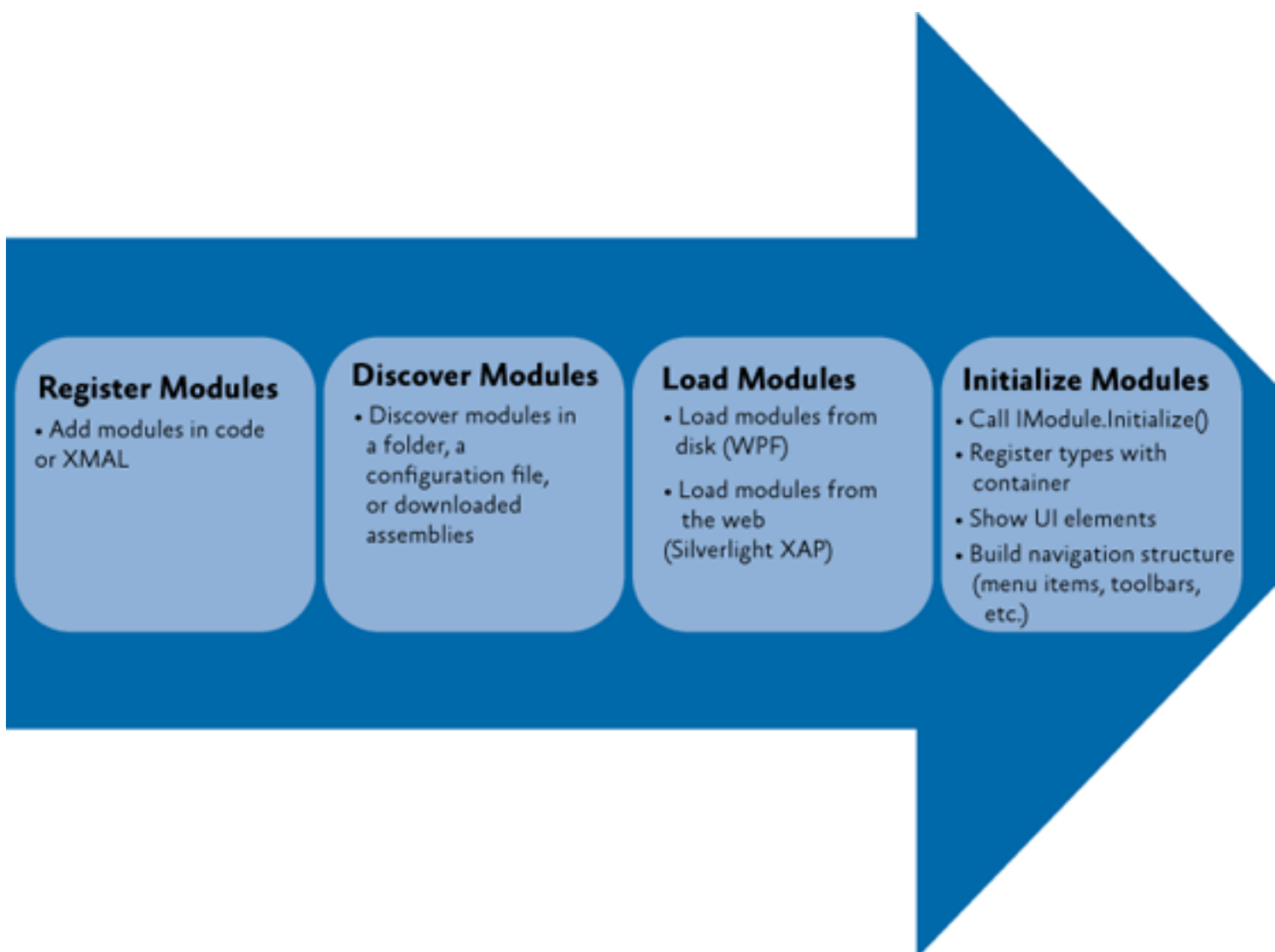
*توضیح درباره برنامه های ماژولار

در تولید پروژه های نرم افزاری بزرگ هر چه قدر هم اگر در تهیه فایل های اسمبلی، کلاس ها، اینترفیس ها و کلا طراحی پروژه به صورت شی گرا دقت به خرج دهیم باز هم ممکن است پروژه به صورت یک پارچه طراحی نشود. یعنی بعد از اتمام پروژه، توسعه، تست پذیری و نگهداری آن سخت و در بعضی مواقع غیر ممکن خواهد شد. برنامه نویسی ماژولار این امکان را فراهم می کند که یک پروژه با مقیاس بزرگ به چند پروژه کوچک تقسیم شده و همه مراحل طراحی و توسعه و تست برای هر کدام از این ماژول ها به صورت جدا انجام شود.

Prism امکاناتی رو برای طراحی و توسعه این گونه پروژه ها به صورت ماژولار فراهم کرده است:

ابتدا باید نام و مکان هر ماژول رو به Prism معرفی کنیم که می توانیم اون ها رو در کد یا Xaml یا Configuration File تعریف کنیم. با استفاده از Metadata باید وابستگی ها و مقادیر اولیه برای هر ماژول مشخص شود. با کمک تزریق وابستگی ها ارتباطات بین ماژول ها میسر می شود. ماژول مورد نظر به دو صورت OnDemand و Available لود خواهد شد.

در شکل زیر مراحل بالا قابل مشاهده است:



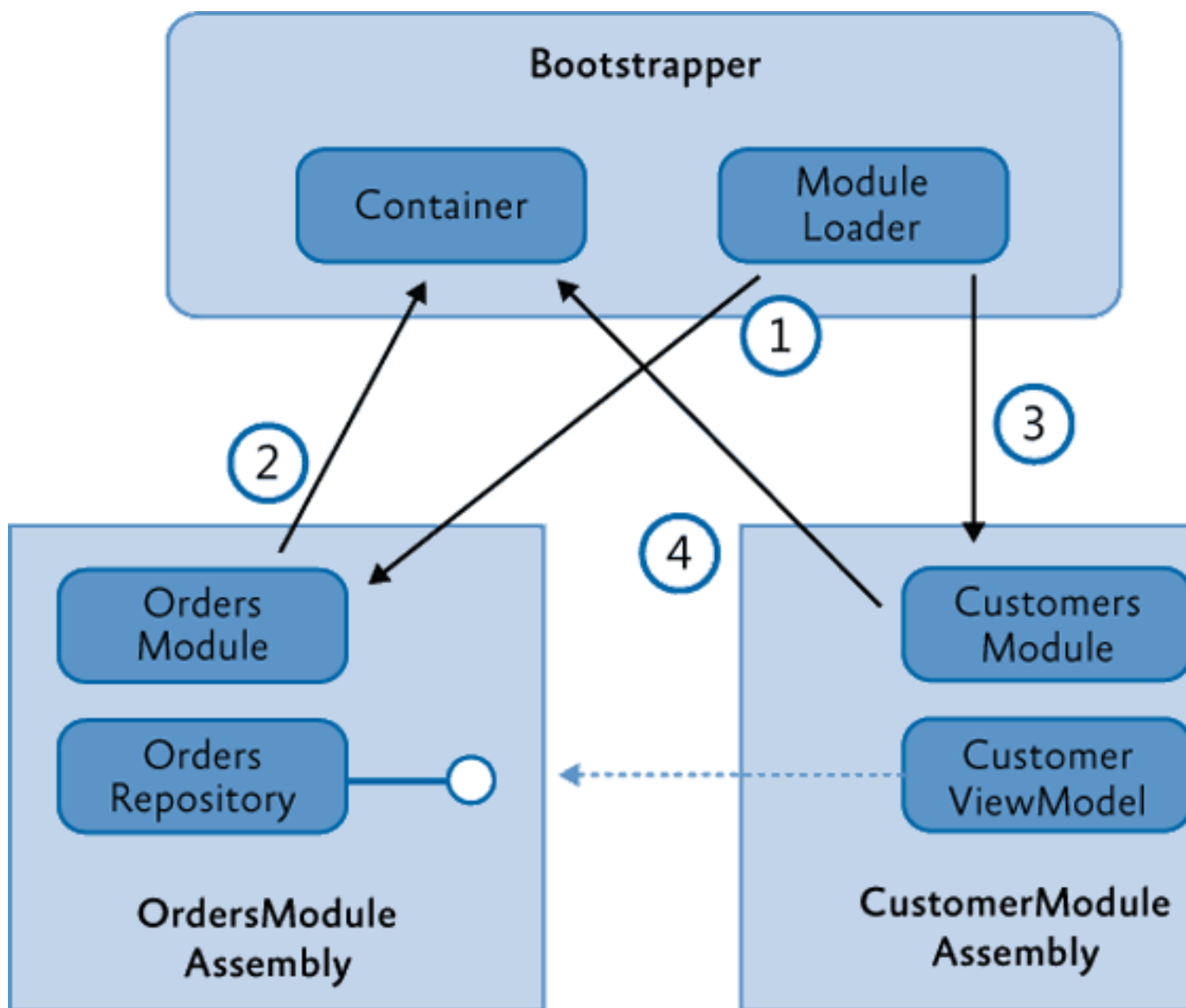
Bootstrapper چیست؟

در هر پروژه ماژولار (مختص Prism نیست) برای اینکه ماژول‌های مختلف یک پروژه، قابلیت استفاده به صورت یک پارچه رو در یک Application داشته باشند باید مفهومی به نام Bootstapper رو پیاده سازی کنیم که وظیفه اون شناسایی و پیکربندی و لود ماژول هاست. در Prism دو نوع Bootstrapper پیش فرض وجود دارد.

MefBootstrapper : کلاس پایه Bootstrapper که مبنای آن MEF است. اگر قصد استفاده از MEF رو در پروژه‌های خود دارید ([^](#)) Bootstrapper شما باید از این کلاس ارث ببرد.

UnityBootstrapper : کلاس پایه Bootstrapper که مبنای آن UnityContainer است. اگر قصد استفاده از UnityContainer یا Service Locator ([^](#)) رو در پروژه‌های خود دارید Bootstrapper شما باید از این کلاس ارث ببرد.

تصویری از ارتباط Bootstrapper با ماژول‌های سیستم



مفهوم Shell

در پروژه‌های WPF، در فایل App.xaml توسط یک Uri نقطه شروع پروژه را تعیین می‌کنیم. در پروژه‌های Silverlight به وسیله خاصیت RootVisual نقطه شروع سیستم تعیین می‌شود. در Prism نقطه شروع پروژه توسط bootstrapper تعیین می‌شود. دلیل این امر این است که Shell در پروژه‌های مبتنی بر Prism متکی بر Region Manager است. از Region برای لود و نمایش ماژول‌ها استفاده خواهیم کرد.

ادامه دارد...

نظرات خوانندگان

نویسنده: محمد احمدی
تاریخ: ۱۳۹۲/۰۴/۰۱ ۱۰:۰۶

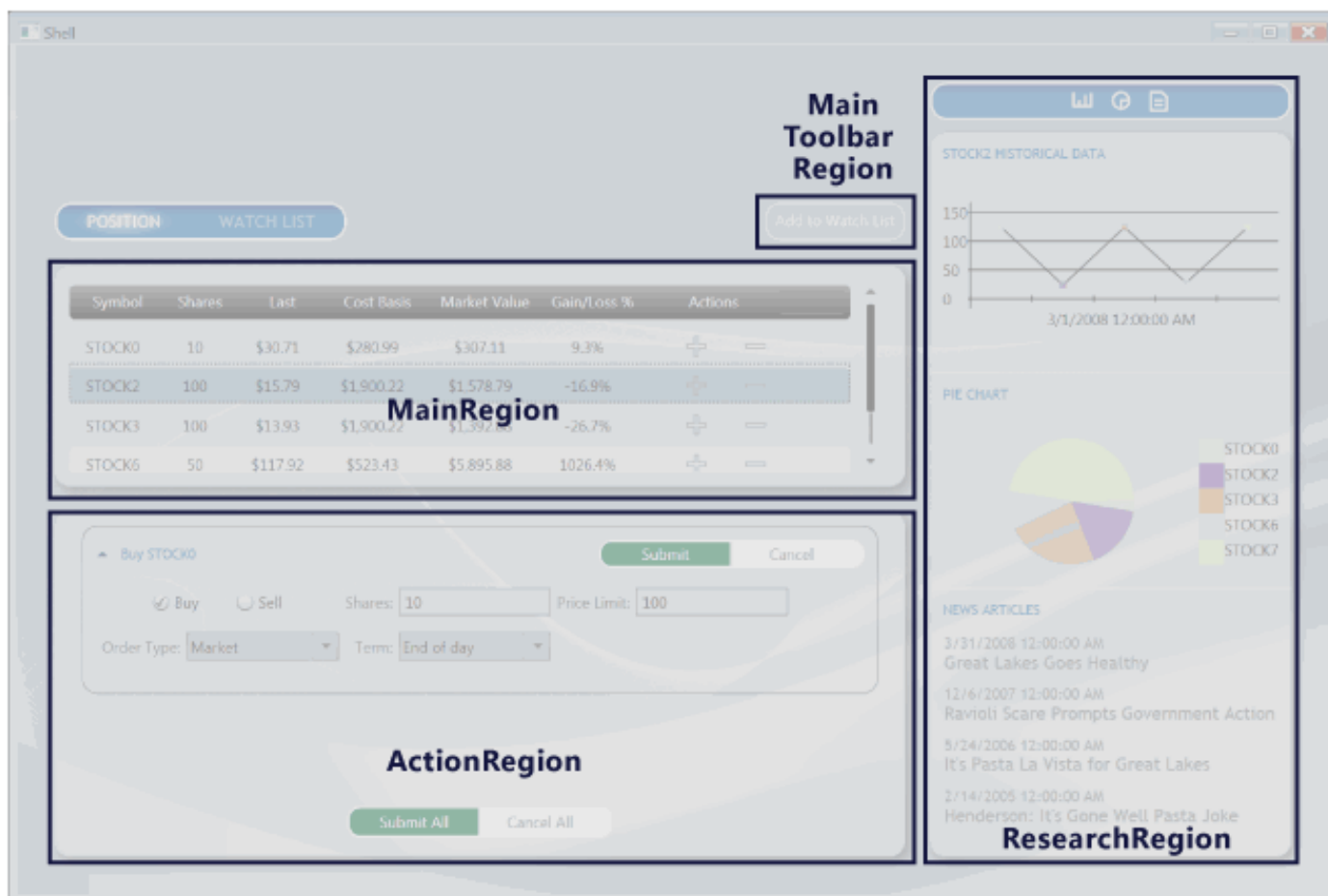
با سلام و تشکر از مطلب مفیدتون
همانطور که می‌دانید مدل‌های مختلف توسعه MVVM برای مقاصد مختلف بهتر است و به طور کلی نمی‌توان گفت که کدام بهتر است
لطفا در ادامه مطلب این فریم ورک را با MVVM Light هم مقایسه بفرمائید تا موارد استفاده هر کدام بهتر مشخص شود

نویسنده: مسعود م. پاکدل
تاریخ: ۱۳۹۲/۰۴/۰۱ ۱۱:۵۶

ممنون.
من از Prism به عنوان بهترین فریم ورک نام نبردم بلکه از عنوان قوی‌ترین فریم ورک استفاده کردم
"می‌تونیم Prism رو به عنوان قوی‌ترین فریم ورک برای پیاده سازی پروژه‌های بزرگ و قوی و ماژولار با تکنولوژی WPF یا Silverlight بنامیم." که لزوماً به معنی بهترین نیست.
MVVM Light در حال حاضر به عنوان محبوب‌ترین فریم ورک برای MVVM است که این محبوبیت بیشتر به خاطر راحتی کار با اون هست.

MVVM Light نظیر Prism هم قابلیت استفاده در WPF را دارد و هم Silverlight (مزیت). MVVM Light راهکار مشخصی برای پیاده سازی پروژه‌های ماژولار ندارد (منظور Modular Composite Application است) در حالی که Prism برای تولید Modular Composite Application طراحی شده است. برای اینکه بتوانید، بعضی از قابلیت‌ها موجود در Prism را برای پروژه‌های ماژولار شبیه سازی کنید باید از ترکیب MEF و MVVM Light استفاده کنید.

Prism به شما این امکان رو می‌ده که حتی برای Popup Window ها هم Region طراحی کنید (مزیت). با Prism می‌تونید به راحتی برای یک Command رفتار تعریف کنید (به صورت توکار از Interaction ها استفاده می‌کنه (مزیت)) برای این کار در MVVM Light شما باید از EventToCommand ها استفاده کنید که اصلاً قابل مقایسه به مباحث Composite Command و Command Behavior نیست.
معادل Messaging در MVVM Light در Prism شما EventAggregator ها رو در اختیار دارید.
Prism به صورت توکار از dependency Injection استفاده میکنه و دو فریم ورک هم به شما پیشنهاد میده یکی MEF و دیگری UnityContainer (مزیت).
Prism به صورت توکار از Composite UI هم پشتیبانی می‌کند. به تصویر زیر دقت کنید:



به راحتی می‌تونید با استفاده از RegionManager موجود در Prism نواحی هر صفحه رو تقسیم بندی کنید و هر ناحیه هم می‌تونه توسط یک ماژول لود شود. برای طراحی و مدیریت صفحات در MVVM Light باید خودتون دست به کار بشید. یادگیری و استفاده از قابلیت‌های MVVM Light در حد دو یا سه روز زمان می‌برد در حالی که برای یادگیری قابلیت‌های Prism یک کتاب نوشته شده است ([^](#))

*در پایان دوباره تاکید می‌کنم که اگر نیازی به تولید و توسعه پروژه به صورت ماژولار رو ندارید بهتره که اصلاً به Prism فکر نکنید.

نویسنده: Petek

تاریخ: ۱۳۹۲/۰۴/۰۲ ۰:۳۶

با سلام

دوست عزیز ممنون میشم این مطلب جالب و مفید رو هر چه بیشتر و سریعتر ادامه بدید . با تشکر

نویسنده: محمد احمدی

تاریخ: ۱۳۹۲/۰۴/۰۲ ۱۳:۱۴

دوست عزیز

ممنونم از راهنمایی جامع و مفیدتون . امیدوارم هر چه زودتر مطالب بیشتری در این زمینه از شما یاد بگیریم