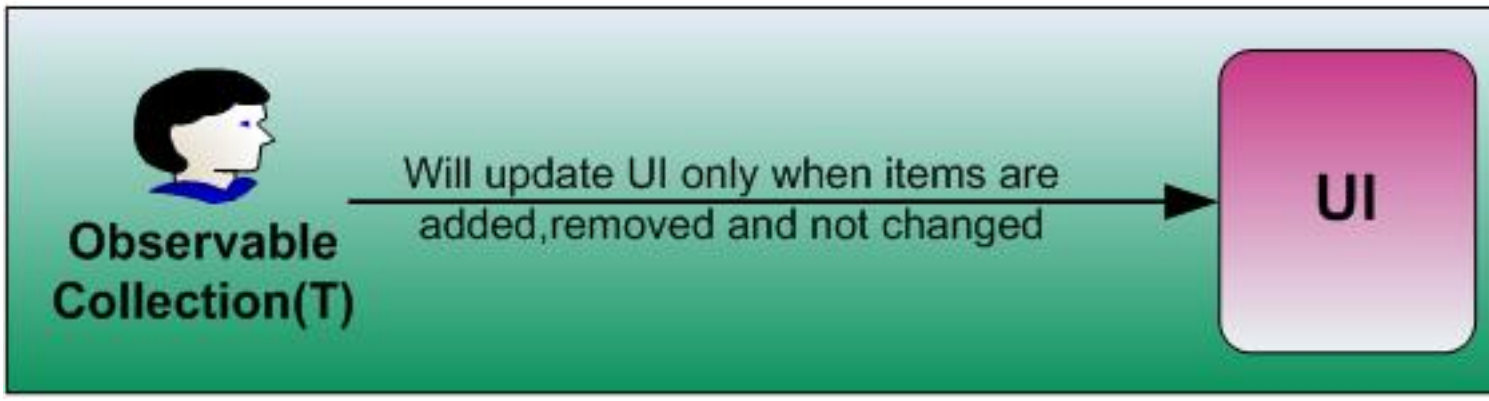
عنوان: **خلاصه‌ای کاربردی در مورد Observable collection**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۱۲/۱۴ ۱۳:۳۲:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WPF

Observable collection
در WPF را می‌توان نوعی لیست جنریک ویژه تعریف کرد که زمانیکه به کنترلی بایند شد، کنترل را از تغییرات خودش آگاه می‌کند. برای مثال اگر آیتمی به این لیست اضافه شد بلافاصله آن آیتم را در کنترل مقید به آن نیز خواهید دید، به همین ترتیب در مورد ویرایش و یا حذف یک آیتم، بدون نیاز به کوچکترین تماسی با کنترل مورد نظر. برای مثال اگر مقدار یک خاصیت را تغییر دادید، بلافاصله بدون اینکه به کنترل مقید به آن اعلام کنیم که لطفا این مورد ویژه را برای من تغییر بده، شاهد نتیجه‌ی نهایی خواهیم بود.



اما استفاده‌ی پیشرفته از این لیست جنریک ویژه به همینجا ختم نشده و حین اضافه کردن کمی پیچیدگی به برنامه مشکلات عدیده‌ای بروز می‌کنند که آن‌ها را جهت دسترسی ساده‌ی بعدی در زیر لیست می‌کنم:

الف) اصلا Observable collection چیست؟ چکار می‌کند؟
List vs ObservableCollection vs INotifyPropertyChanged in Silverlight

ب) نمی‌توانم از این مجموعه‌ی اشیای خودآگاه سازنده در یک ترد استفاده کنم. مشکل کجاست؟
این روزها نمی‌توان یک برنامه‌ی دسکتاپ خوب را بدون استفاده از تردها متصور شد. اما به محض سعی در به روز رسانی این لیست جنریک در یک ترد دیگر (ترد دیگر منظور هر تردی بجز ترد اصلی برنامه است که کار مدیریت رابط کاربر را به عهده دارد) خطای زیر ظاهر می‌شود:

```
This type of CollectionView does not support changes to its SourceCollection from a thread different
from the Dispatcher thread
```

راه حل:
Adding to an ObservableCollection from a background thread

ج) یکی از خاصیت‌های یک شیء این لیست جنریک ویژه را تغییر داده‌ام. اما هیچ تغییری در کنترل بایند شده به آن مشاهده نمی‌کنم. مشکل در کجاست؟
راه حل: پیاده سازی اینترفیس INotifyPropertyChanged را فراموش کرده‌اید:

Data Binding in WPF with the Monostate Pattern

د) خوب، این که خیلی دردسر دارد! راه ساده‌تری برای تعریف این موارد نیست؟!
هوشمندانه‌ترین روشی که برای حل این مساله تابحال دیده‌ام:
An easier way to manage INotifyPropertyChanged

نظرات خوانندگان

نویسنده: ..::A-3BT::..
تاریخ: ۱۳۸۸/۱۲/۱۴ ۱۴:۰۴:۳۳

تا پیش از این , این الگو فقط در WPF از طرف ماکروسافت پیاده سازی شده بود ولی در 4 .net. ماکروسافت اون رو به عنوان بخشی از BCL در نظر گرفته

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۱۲/۱۴ ۱۴:۱۰:۳۷

البته بهتره بگیم استفاده از این الگو وگرنه تعریف آن در فضای نام system قرار دارد (System.Collections.ObjectModel) .

نویسنده: Mehran
تاریخ: ۱۳۸۸/۱۲/۱۴ ۱۶:۴۱:۰۰

ممنون از مطالب مفید شما واقعا بر روی لبه تکنولوژی های دات نت قدم بر می دارید. در مورد COLLECTION های OBSERVABLE این نکته هم حائز اهمیت است که این لیست ها دقیقا به اندازه نیاز کاربر در UI برنامه مقادیر را در حافظه لود کرده و اصطلاحا می توان با بکاربری این COLLECTIONها در یک کنترل خاصیت LazyLoading به کنترل بخشید.

نویسنده: mojtabakaviani
تاریخ: ۱۳۸۸/۱۲/۱۹ ۰۹:۳۰:۱۴

ممنون از مطالب مفیدتون...
اگه ممکنه قسمت خبر ها و تازه های دنیای کامپیوتر رو دوباره اضافه کنید...
از لینک های که می گذاشتید به یه عالمه تازه ای بیشتر می رسیدیم.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۱۲/۱۹ ۱۰:۵۹:۵۳

سلام
هنوز هم اینکار رو می‌کنم. اما محل انتشار آن تغییر کرده:
http://www.idevcenter.com/links/upcoming

نویسنده: Meysam
تاریخ: ۱۳۸۹/۰۱/۱۰ ۲۰:۰۳:۳۷

تو این CheckForIllegalCrossThreadCalls نیست؟ اینکار استفاده از Thread بسیار کم میکنه!

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۱/۱۰ ۲۳:۳۴:۲۳

چرا، Dispatcher.CheckAccess دارد.