

## **راهنمای استفاده از کنترل TimelineViewer**

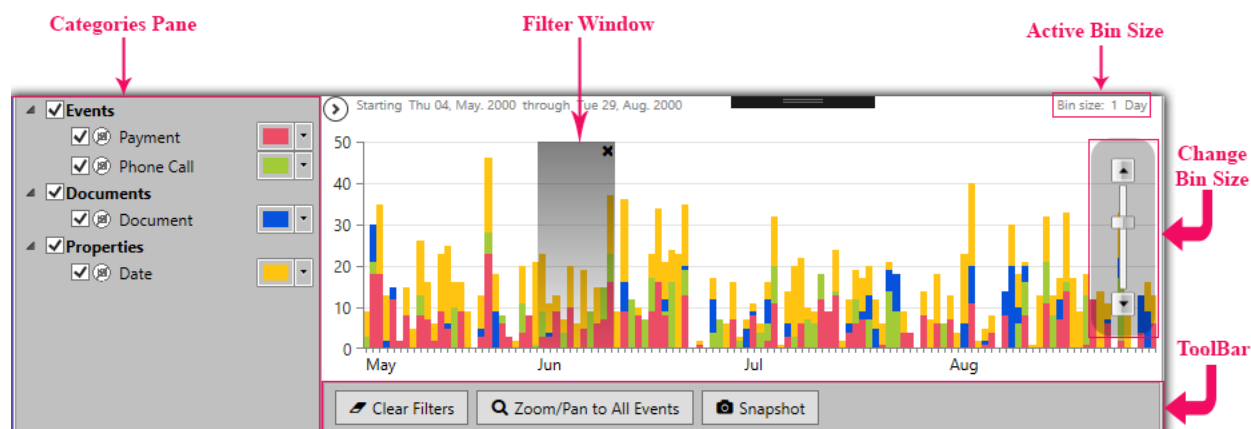
## 1- مقدمه

کنترل TimelineViewer در واقع یک نمودار میله‌ای پشته‌ای (Stack Bar Chart) با محور عمودی عددی و محور افقی زمان می باشد که با دیدگاه استفاده در پروژه های بیگ دیتا طراحی شده است.

این کنترل در پروژه های Windows Application در قالب WPF طراحی شده و مبتنی بر .NET Framework 4.5.2 می باشد.

کدنویسی این کنترل از خوانایی کافی برای سایر توسعه دهندگان برخوردار است و از راهنمای نام گذاری MSDN در نام گذاری ها استفاده شده است. اصول Clean Code در این پروژه رعایت شده است.

## 2- شمای کلی کنترل



تصویر شماره 1

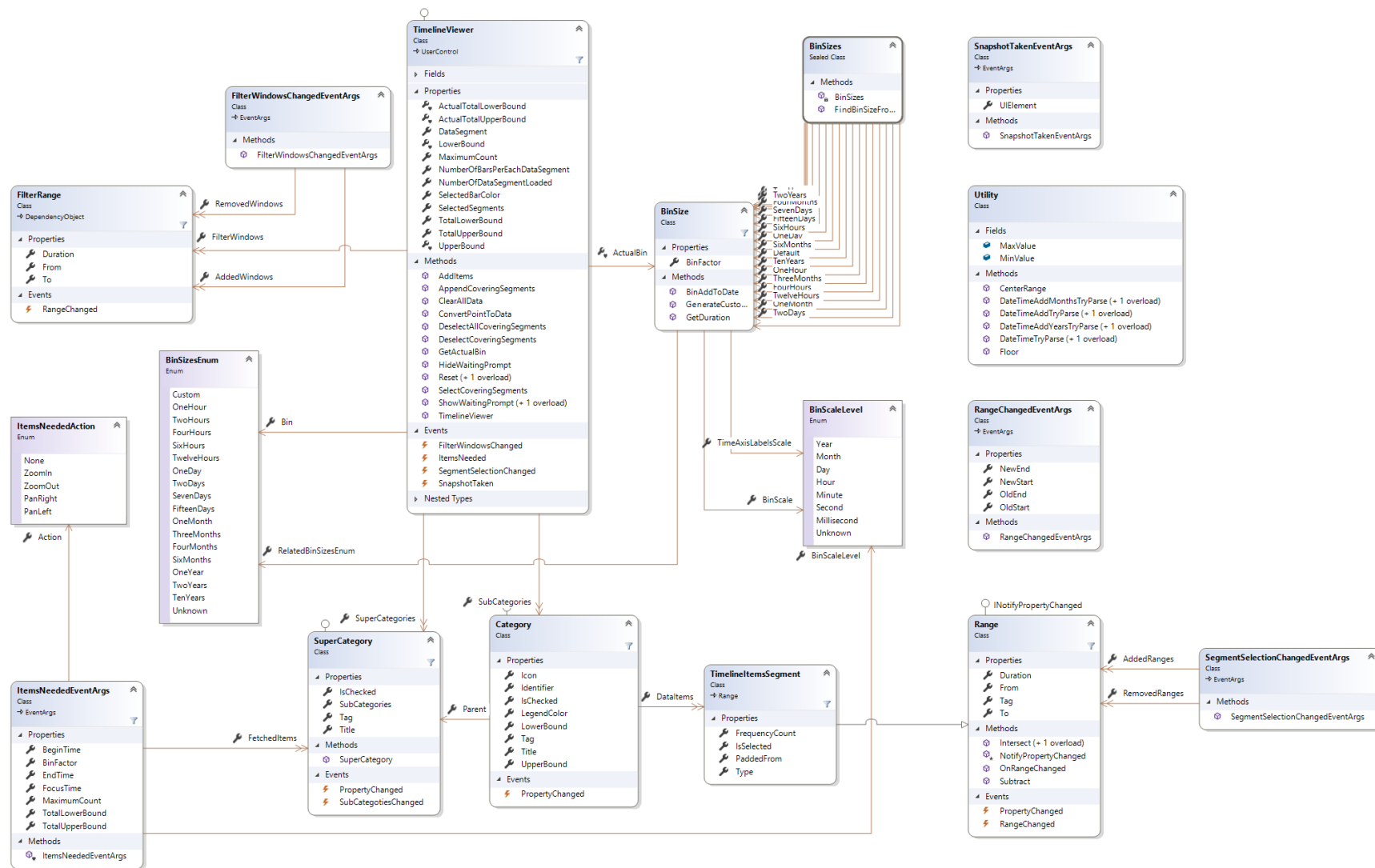
## کتابخانه های بکار برده شده در کنترل

در این کنترل علاوه بر کتاب خانه هایی که بطور پیش فرض در در یک پروژه Class Library به پروژه افزوده می شود از کتابخانه های ذیل استفاده شده است.

جدول شماره 1

Library	Version	Library	Version
FontAwesome.WPF	4.7.0.37774	GPAS.ColorPickerViewer	1.0.0.0
Telerik.Windows.Controls	2018.2.620.45	Telerik.Windows.Controls.Chart	2018.2.620.45
Telerik.Windows.Controls.Input	2018.2.620.45	Telerik.Windows.Controls.Navigation	2018.2.620.45
Telerik.Windows.Data	2018.2.620.45		

### 3- نمودار فنی کنترل



تصویر شماره 2

## 4- نحوه استفاده از کنترل

برای بارگزاری و نمایش داده ها در خط زمان یک رخداد به نام `ItemsNeeded` و دو متد `Reset` و `AddItems` در نظر گرفته شده است که استفاده کننده ابتدا باید متد `Reset` را فراخوانی کند، سپس کنترل یک رخداد `ItemsNeeded` صادر می کند و در ادامه گوش دهنده به رخداد `ItemsNeeded`، توسط متد `AddItems`، داده های مورد نظر خود را جهت نمایش به کنترل پاس دهد.

## 5. بخش های اصلی کنترل خط زمان (TimelineViewer)

در این بخش سعی شده تمام `Property` ها، متد ها و رخدادهای عمومی (`public`) کنترل به ترتیب اولویت توضیح داده شوند.

### 1-5- رخداد `ItemsNeeded`

همانطور که از نام این رخداد پیداست، `ItemsNeeded` هنگامی صادر می شود که کنترل به داده های جدیدی برای نمایش نیاز داشته باشد که در 5 حالت این متد فراخوانی می شود.

1. فراخوانی متد `Reset`

2. `Zoom In` (کوچکتر شدن `Bin` کنترل)

3. `Zoom Out` (بزرگتر شدن `Bin` کنترل)

4. `Pan Right` (حرکت عرضی داده ها به سمت راست تا نقطه ای که نیاز به بارگزاری آیتم های لود نشده احساس شود).

5. `Pan Left` (حرکت عرضی داده ها به سمت چپ تا نقطه ای که نیاز به بارگزاری آیتم های لود نشده احساس شود).

این رخداد یک پارامتر از نوع `ItemsNeededEventArgs` به متدهای مرتبط پاس می دهد که تنظیم صحیح پارامترهای این متد سهم زیادی در نمایش و عملکرد صحیح کنترل دارد.

## ItemsNeededEventArgs - 5-2

متد سازنده (Constructor) این آرگومان از نوع `internal` می باشد که به معنای این است که از بیرون کنترل نمی توان به متغیر از نوع `ItemsNeededEventArgs` ساخت.

`property` های موجود در این آرگومان با دو رویکرد ورودی یا خروجی طراحی شده اند.

`Property` ها با رویکرد ورودی: این نوع از مشخصه ها لازمه رخداد `ItemsNeeded` هستند و بدون آن ها داده ها قابل واکنشی نیستند. این مشخصه ها عبارتند از:

`BeginTime` و `EndTime`: کنترل خط زمان (`TimelineViewer`) `BeginTime` و `EndTime` را بر اساس ابتدا و انتهای بازه ای که داده ای آن را لازم دارد پر می کند.

`BinScaleLevel` و `BinFactor`: کنترل خط زمان `BinScaleLevel` و `BinFactor` را بر اساس `Bin Size` جاری پر می کند. این دو مشخصه با هم نمایانگر `Duration` یک میله در کنترل هستند اما به دلیل اینکه در `BinScale` های ماهانه و سالانه `Duration` ثابتی وجود ندارد (مثلا یک ماه 28 روز و ماه دیگر 31 روز می باشد) به صورت دو پارامتر مجزا و قابل محاسبه دریافت می شود.

`Action`: کنترل خط زمان `Action` را بر اساس نوع درخواست رخداد مقدار دهی می کند که مشتمل بر 5 حالت `None` (تنها در فراخوانی متد `Reset`)، `ZoomIn`، `ZoomOut`، `PanRight` و `PanLeft` است.

`FocusTime`: کنترل خط زمان `FocusTime` را بر اساس موقعیت زمانی روی نمایشگر کاربر مقداردهی می کند. به طور مثال هنگامی که کاربر با گرفتن کلید `Ctrl` و اسکرول ماوس اقدام به `Zoom` می کند `FocusTime` اشاره به زمان مرتبط با موقعیت مکانی ماوس دارد.

`Property` ها با رویکرد خروجی: این نوع از مشخصه ها لازمه تابع `AddItems` هستند و بدون مقداردهی صحیح آن ها یا داده ای اصلا وجود ندارد یا اینکه به درستی قابل نمایش نیستند که عبارتند از:

**FetchItems**: کنترل خط زمان انتظار دارد گوش دهنده به رخداد، **FetchItems** را با داده‌های خواسته‌شده پر کند؛ در صورتی که این ویژگی **null** باشد، یعنی کسی به آن پاسخ نداده است و حالت «نمای بدون داده» روی کنترل به نمایش درمی‌آید.

**TotalLowerBound** و **TotalUpperBound**: کنترل خط‌زمان انتظار دارد گوش دهنده به رخداد، مقادیر **TotalLowerBound** و **TotalUpperBound** را به درستی مقدار دهی کند. اگر گوش دهنده به رخداد آن‌ها را مقدار دهی نکند مقادیر قبل از ارسال رخداد بدون تغییر باقی می‌مانند. در صورت عدم مقدار دهی این مقادیر با داده‌های جدید هیچ پیام خطایی صادر نمی‌شود اما ممکن است کنترل در نمایش داده‌ها مختل شود.

**MaximumCount**: کنترل خط زمان انتظار دارد گوش دهنده به رخداد، مقدار **MaximumCount** را به درستی مقدار دهی کند. اگر گوش دهنده به رخداد آن را مقدار دهی نکند مقدار قبل از ارسال رخداد بدون تغییر باقی می‌مانند. در صورت عدم مقدار دهی این مقادیر با داده‌های جدید هیچ پیام خطایی صادر نمی‌شود اما ممکن است کنترل در نمایش داده‌ها مختل شود.

### 3-5- متد Reset

این متد در دو حالت تعریف و فراخوانی می‌شود.

حالت اول - بدون پارامتر ورودی - **Reset()**: در این حالت مشخصه **Bin** کنترل با مقدار پیش‌فرض تنظیم شده و سپس رخداد **ItemsNeeded** فراخوانی می‌شود.

حالت دوم - با یک پارامتر ورودی از نوع **BinSizeEnum - Reset(BinSizeEnum)**: در این حالت مشخصه **Bin** کنترل با پارامتر ورودی تنظیم شده و سپس رخداد **ItemsNeeded** فراخوانی می‌شود.

### 4-5- متد AddItems

این متد با یک پارامتر ورودی از نوع **ItemsNeededEventArgs** به شکل زیر فراخوانی می‌شود.

**AddItems(ItemsNeededEventArgs)**

در واقع این متد در ادامه رخداد `ItemsNeeded` است و اگر گوش دهنده به این رخداد این متد را فراخوانی نکند کار رخداد ناقص باقی می ماند.

متد `AddItems` یک پارامتر از نوع `ItemsNeededEventArgs` به کنترل پاس می دهد که قبل از پاس دادن باید `property` هایی که نقش خروجی در آرگومان `ItemsNeededEventArgs` را دارند مقدار دهی شده باشند.

## 5-5- مشخصه `Bin (Property)`

مشخصه `Bin` از نوع `BinSizeEnum` می باشد که نمایانگر `Bin Size` جاری فعال (جاری) کنترل است. مفهوم `Bin Size` یک مفهوم کلیدی در نمایش خط زمان است که نمایانگر طول هر میله در محور زمان است. به طور مثال `Bin = TwoDays` به این معناست که طول هر میله در نمودار `TimelineViewer` از منظر زمانی دو روز است.

با هر تغییر `Bin` یک رخداد `ItemsNeeded` اتفاق می افتد که `Action` از نوع `ZoomIn` یا `ZoomOut` است. مقادیری که مشخصه `Bin` می تواند داشته باشد را در تصویر شماره 2 می توانید مشاهده کنید.

## 5-6- مشخصه `ActualBin` و متد `GetActualBin()`

این مشخصه و متد `Bin Size` واقعی کنترل را که از نوع `BinSize` می باشد بر می گردانند.

### 5-6-1- کلاس `BinSize`

این کلاس تعاریف داخلی `Bin` را در خود نگه می دارد.

مشخصه ها و متد های اصلی کلاس `BinSize`:

1. مشخصه `BinScale`: این مشخصه از نوع `BinScaleLevel` است که یک `Enum` می باشد و مقادیر آن را در تصویر شماره 2 می توانید مشاهده کنید. به طور مثال `Bin = TwoDays` مقدار `BinScale` برابر `Day` می باشد.

2. مشخصه **BinFactor**: این مشخصه یک مقدار عددی **double** است که نمایانگر تعداد **BinScale** می باشد. به طور مثال **Bin = TwoDays** مقدار **BinFactor** برابر با 2 می باشد.
3. مشخصه **RelatedBinSizesEnum**: این مشخصه از نوع **BinSizeEnum** است که ارتباط دهنده **BinSizeEnum** با اصلی است. اگر مقدار معادل **BinSize** در **BinSizeEnum** موجود نبود این مشخصه با **Custom** مقداردهی می شود.
4. مشخصه **TimeAxisLabelScale**: این مشخصه نمایانگر نحوه نمایش مقادیر روی محور زمان کنترل است. به طور مثال **Bin = TwoDays** مقدار **TimeAxisLabelScale** برابر **Month** می باشد. به طور معمول مقدار این مشخصه یک **Level** از **BinScale** بیشتر است.
5. متد **GetDuration**: این متد می بایست طول **BinSize** را بر حسب **TimeSpan** برگرداند. با توجه به اینکه **BinScale** های ماهانه و سالانه **Duration** ثابتی وجود ندارد (مثلا یک ماه 28 روز و ماه دیگر 31 روز می باشد) تابع **GetDuration** یک پارامتر ورودی بنام **beginTime** می گیرد که **Duration** مربوط به هر **BinSize** از این نقطه شروع می شود. به طور مثال **Bin = TwoDays** این تابع یک مقدار **TimeSpan** معادل 2 روز را برمی گرداند.

## 5-7- مشخصه **SuperCategories**

این مشخصه مجموعه ای از **SuperCategory** های کنترل خط زمان را در بر می گیرد که مشخصه های اصلی هر **SuperCategory**، عنوان (**Title**) و مجموعه ای از فرزندان **SuperCategory** تحت عنوان **SubCategories** می باشد.

## 5-8- مشخصه **SubCategories**

این مشخصه مجموعه ای کامل از **SubCategories** های تمام **SuperCategory** های کنترل است که از نوع **Category** می باشند. مشخصه های اصلی هر **SubCategory** شامل عنوان (**Title**)، تصویر (**Icon**)، رنگ (**LegendColor**)، **Identifier** (مشخصه کلیدی و منحصر بفرد هر **Category**)، دسته پدر (**Parent**) و اصلی ترین مشخصه **DataItems** است که مجموعه ای از **TimelineItemsSegment**



می باشد و داده های Fetch شده در متد گوش دهنده به رخداد ItemsNeeded برای هر دسته در DataItems ریخته می شود.

مشخصه اصلی هر DataItem شامل شروع بازه زمانی (From)، انتهای بازه زمانی (To)، طول بازه زمانی (Duration)، حالت انتخاب/ عدم انتخاب (IsSelected) و عددی که DataItem در نمودار باید نمایش دهد (FrequencyCount) است.

لازم است در مورد خاصیت رنگ هر Category این توضیح داده شود که حتما نیاز نیست از سمت استفاده کننده رنگی به دسته اختصاص داده شود و خود کنترل به طور پیش فرض رنگی را به دسته اختصاص می دهد که البته این رنگ از طریق کد نویسی توسط استفاده کننده از کنترل و همچنین قسمت Categories Pane موجود در تصویر 1 قابل تغییر می باشد.

نکته: ویژگی Identifier شناسه ی یکتایی برای هر دسته تعیین می کند که در تمام ارجاعات به دسته بندی های تعریف شده (در AddItems) از آن استفاده می شود.

## 9-5 متد ( ) ClearAllData

تمام داده های موجود در کنترل پاک شده و کنترل به حالت اولیه (قبل از بارگزاری هر داده ای) باز می گردد.

## 10-5- مشخصه DataSegment

این مشخصه از نوع TimeSpan می باشد و نمایانگر طول زمانی کنترل در نمایی است که کاربر در حال حاضر می بیند. یعنی در Bin = TwoDays در صورتی که تعداد میله ها در نمای حاضر 100 عدد باشد، DataSegment مقدار 200 روز را نمایش می دهد.

## 11-5- مشخصه NumberOfDataSegmentLoaded

این مشخصه بیشترین تعداد DataSegment های مجاز برای بارگزاری توسط کنترل را مشخص می کند. در صورتی که به داده های فراتر از این مجموعه نیاز باشد رخداد ItemsNeeded فراخوانی شده، داده های

خارج از محدوده پاک می شود و داده های داخل محدوده جدید بارگزاری می شوند. نحوه بارگزاری و مدیریت داده ها در بخش 6 همین سند به تفصیل توضیح داده شده است.

## 12-5- مشخصه MaximumCount

مقدار ماکزیمم محور عمودی کنترل را مشخص می کند که همیشه مضربی از 5 است. اگر مقدار تخصیص داده شده به این مشخصه مضربی از 5 نباشد به نزدیکترین مضرب 5 بزرگتر از عدد داده شده گرد می شود. بهتر است این مشخصه از طریق متد AddItems مقداردهی شود.

## 13-5- مشخصه های TotalUpperBound و TotlLowerBound

این دو مشخصه مقداری از نوع DateTime می گیرند که نمایانگر محدوده کلی بازه کنترل خطزمان می باشند. مقداردهی این مشخصه ها به این معناست که هیچ داده لود شده/لود نشده ای در خارج این محدوده وجود ندارد. بهتر است این مشخصه ها از طریق متد AddItems مقداردهی شوند.

## 14-5- UpperBound و LowerBound

این دو مشخصه مقداری از نوع DateTime می گیرند که نمایانگر محدوده بارگزاری شده کنترل خطزمان می باشند.

## 15-5- مشخصه SelectedSegments

این مشخصه شامل مجموعه ای از Range های زمانی است که مشخص کننده بازه های انتخاب شده در کنترل می باشد. هر میله از نمودار که با یکی از بازه های مجموعه SelectedSegments اشتراک داشته باشد به حالت انتخاب در می آید که مشخصه این حالت انتخاب، تغییر رنگ تمام میله به رنگ تخصیص داده شده در مشخصه SelectedBarColor است.

هر تغییر در مجموعه SelectedSegments موجب صدور رخداد SegmentSelectionChanged می شود.

## SegmentSelectionChanged - 5-16 رخداد

این رخداد در دو حالت فراخوانی می شود.

حالت اول: با کلیک روی هر یک از میله‌های در حال نمایش خط زمان تمام انتخاب‌های قبلی حذف شده و بازه مربوط به آن میله به مشخصه **SelectedSegments** افزوده می شود. با گرفتن کلید **Ctrl** و کلیک روی میله قابلیت **Multi Select** فعال می شود.

حالت دوم: با فراخوانی متد های زیر؛

- متد **SelectCoveringSegments(Ranges)**؛ در این صورت انتخاب‌های قبلی لغو می شوند، مواردی که میله‌شان، تاریخ‌های ورودی را پوشش دهد، انتخاب می شوند و در **SelectedSegments** ریخته شوند.
- با فراخوانی تابع **AppendCoveringSegments(Ranges)**؛ در این صورت، مواردی که میله‌شان، تاریخ‌های ورودی را پوشش دهد، به مواردی که از قبل انتخاب شده اند افزوده می شوند و به **SelectedSegments** نیز اضافه می گردند.
- با فراخوانی تابع **DeselectCoveringSegments(Ranges)**؛ در این صورت مواردی که میله هایشان، تاریخ ورودی را پوشش می دهد از حالت انتخاب خارج شده و همچنین از **SelectedSegments** حذف می شوند.
- با فراخوانی تابع **DeselectAllCoveringSegments( )**؛ همه میله ها از حالت انتخاب خارج شده و همچنین **SelectedSegments** خالی شود.

## ShowWaitingPrompt([string]) - 5-17

نمایش حالت انتظار بارگزاری داده با پیام ارسال شده در پارامتر ورودی متد. اگر پارامتری در ورودی متد پاس داده نشود مقدار پیش فرض نمایش داده می شود.

## HideWaitingPrompt( ) - 5-18

مخفی کردن حالت انتظار بارگزاری داده.

## 19-5- مشخصه FilterWindows

این مشخصه دربردارنده مجموعه ای از FilterWindow ها هست که هر FilterWindow قابلیت تغییر، جابجایی، اجاد و حذف از روی خطزمان را دارد. با گرفتن کلید Ctrl و درگ رو قسمت نمودار خطزمان می توانید یک FilterWindow ترسیم کنید. پس از ایجاد، حذف، جابجایی یا تغییر سایز یک FilterWindow از مجموعه FilterWindows رخداد FilterWindowsChanged فراخوانی می شود.

## 20-5- رخداد FilterWindowsChanged

این رخداد یک آرگومان بنام FilterWindowsChangedEventArgs دارد که شامل دو مجموعه از FilterWindow های اضافه شده و حذف شده است که این مجموعه های تنها در بردارنده بازه های زمانی پوشش داده شده توسط FilterWindow ها هستند.

## 21-5- قسمت ToolBar

در قسمت Tool Bar مشخص شده در تصویر 1 سه دکمه مشاهده می کنیم که عملکرد آنها به شرح زیر است:

- دکمه Clear Filters: تمامی FilterWindow ها را پاک می کند و طبیعتا رخداد FilterWindowsChanged را فراخوانی می نماید.
- دکمه Zoom/Pan to All Events: این دکمه Bin فعال را در حالتی قرار می دهد که تمامی داده های موجود در یک نما (درون یک DataSegment) مشاهده شوند و Pan را نیز به مرکز نمودار کنترل خطرمان منتقل می کند.
- دکمه Snapshot: وظیفه این دکمه عکس گرفتن از کنترل است که با کلیک برروی دکمه منوی زیر که شامل دو گزینه است نمایان می شود.

- گزینه اول Sanpshot only timeline: این گزینه تنها از قسمت نمودار کنترل عکس برداری می کند.

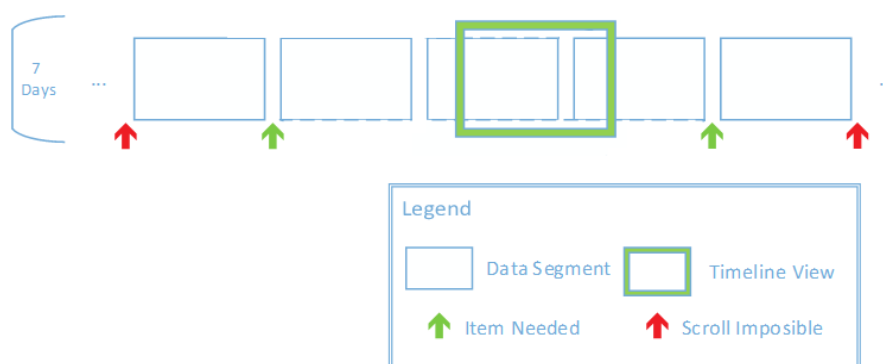
- گزینه دوم Snapshot timeline with category pan: این گزینه از همه کنترل خط زمان (بجز قسمت Toolbar) عکس برداری می کند.

با انتخاب هر یک از د گزینه فوق رخداد SnapshotTaken فراخوانی می شود که یک آرگومان از نوع SnapshotTakenEventArgs را برای گوش دهنده به رخداد فراخوانی می کند که این آرگومان شامل یک مشخصه از نوع UIElement است که در بردارنده کنترلی است که باید از آن تصویر برداری شود.

## 6- بارگذاری در حد نیاز (On-demand)

ممکن است داده های زمانی خط زمان زیاد باشند و یا آنکه پراکندگی زیادی داشته باشند، به همین خاطر بارگذاری یکجای داده ها در Bin Scale های مختلف کار معقولی نیست. از طرفی انتظار می رود کاربر با جابجایی در محور زمان (Pan) و حتی با تغییر Bin Scale ها (Zoom)، منتظر بارگذاری داده نماند؛ به همین دلیل ساختاری در نظر گرفته شده تا فقط داده های محدوده ی نمایشی (DataSegment جاری) کاربر، به علاوه داده های همسایه ی آن بارگذاری شود و غیر از آن ها در صورت نیاز بارگذاری شود.

نمودار ۱ یک ساختار پیشنهادی برای میانگیری داده ها در راستای خط است. در این نمودار، کادر سبز رنگ محدوده ی در حال نمایش توسط خط زمان و مستطیل ها، نشان دهنده ی داده های بارگذاری شده است.



نمودار 1- شمای میانگیری داده ها در خط زمان

به این صورت که تا قبل از رسیدن محدوده‌ی نمایشی به فلش‌های سبزنگ، داده‌ای بارگزاری نمی شود ولی بعدازآن، درصورتی که Pan به نقطه‌ای رسید که فاصله آن کمتر از یک DataSegment تا قسمتی که داده‌ای وجود ندارد بود، به صورت پیش‌دستانه، داده‌ی سمتی که رو به فقدان می‌رود (سمت مخالف حرکت Pan) را با صدور رخداد ItemsNeeded تأمین کند (و البته از سمت دیگر، داده‌ها را از حافظه خارج نماید). پیش‌فرض این است که دریافت‌کننده‌ی رخداد مذکور، آن را به صورت غیرهمگام (Async) پاسخ می‌دهد تا رابط کاربری در این شرایط قفل نشود. درصورتی که پیش از تأمین داده‌های موردنیاز، پیمایش کاربر به نقطه‌ی فاقد داده رسید، باید پیام انتظار (توسط متد ShowWaitingPrompt) به نمایش درآید و کاربر نتواند به پیمایش افقی خود ادامه دهد تا داده‌ها تأمین شود.