<> Code   (!) Issues 5   ⌥ Pull requests   💬 Discussions   ▷ Actions   ⊡ Projects 2

Edit   New issue   Jump to bottom

[Bug] Custom Section Headers don't render correctly on iOS. #6

(!) Open   **Jakar510** opened this issue on Aug 12 · 0 comments

Labels	Bug High Priority
Projects	⊡ Triage

👤 **Jakar510** commented on Aug 12   Owner

Description

Custom Section Headers don't render correctly. Note that this doesn't happen every time, but most of the time.
On iOS it usually will render as a pixel or two wide striped lines. On Android, works fine.

```
public class SectionHeader : Grid
{
        public bool IsExpanded { get; set; }
        protected Label _Label { get; set; }
        protected Image _Icon { get; set; }
        public ImageSource Collapsed { get; set; }
        public ImageSource Expanded { get; set; }


        public SectionHeader( Action command ) : this(new Command(command)) { }
        public SectionHeader( ICommand command ) => Init(command);
        ~SectionHeader()
        {
                ClearGesture(this);
                ClearGesture(_Label);
                ClearGesture(_Icon);
        }
        protected void Init( ICommand command )
        {
                CreateView();
                AddGesture(_Label, command);
                AddGesture(this, command);
```

```
                        {
                            MinimumHeightRequest = 30;
                            //HeightRequest = 40;
                            HorizontalOptions = LayoutOptions.FillAndExpand;
                            VerticalOptions = LayoutOptions.FillAndExpand;


                            ColumnDefinitions = new ColumnDefinitionCollection()
                            {
                                    new ColumnDefinition() { Width = new GridLength(0.1,
            GridUnitType.Star) },
                                    new ColumnDefinition() { Width = GridLength.Star },
                                    new ColumnDefinition() { Width = new GridLength(0.1,
            GridUnitType.Star) },
                            };

                            SetDynamicResource(BackgroundColorProperty, "HeaderBackgroundColor");


                            _Label = new Label()
                            {
                                    Padding = new Thickness(0, 10, 0, 10),
                                    VerticalTextAlignment = TextAlignment.Center,
                                    HorizontalTextAlignment = TextAlignment.Center,
                                    HorizontalOptions = LayoutOptions.FillAndExpand,
                                    VerticalOptions = LayoutOptions.FillAndExpand,

                            };
                            _Label.SetDynamicResource(Label.FontSizeProperty, "HeaderFontSize");
                            _Label.SetDynamicResource(Label.TextColorProperty, "HeaderTextColor");
                            _Label.SetDynamicResource(Label.BackgroundColorProperty,
            "HeaderBackgroundColor");
                            SetColumn(_Label, 1);


                            _Icon = new Image()
                            {
                                    Aspect = Aspect.AspectFit,
                                    HorizontalOptions = LayoutOptions.FillAndExpand,
                                    VerticalOptions = LayoutOptions.FillAndExpand,
                            };
                            SetColumn(_Icon, 0);
                            _Icon.SetDynamicResource(Image.BackgroundColorProperty,
            "HeaderBackgroundColor");


                            Children.Add(_Label);
                            Children.Add(_Icon);
                    }

                protected static void AddGesture( View view, ICommand command )
                {
                        if ( view is null )
```

```
                                        throw new ArgumentNullException(nameof(command));

                        var tap = new TapGestureRecognizer
                        {
                                NumberOfTapsRequired = 1,
                                Command = command
                        };

                        view.GestureRecognizers.Add(tap);
                }
                protected static void ClearGesture( View view )
                {
                        if ( view is null )
                                throw new ArgumentNullException(nameof(view));

                        view.GestureRecognizers.Clear();
                }

                public void Toggle()
                {
                        bool check = !IsExpanded;
                        IsExpanded = check;

                        _Icon.Source = IsExpanded ? Collapsed :  Expanded;
                }
                public void SetText( string text ) => _Label.Text = text;
        }
```

Steps to Reproduce

1. Create a few sections with the above SectionHeader class as the header.
2. Show the view.
3.

Expected Behavior

Show the header.

Actual Behavior

On iOS it usually will render as a pixel or two wide striped lines. On Android, works fine.

Platforms

☐ Android
☑ iOS

Basic Information

- Android Support Libray version: Xamarin.AndroidX.AppCompat 1.1.0.1
- Affected Devices:

Screenshots

Reproduction Link

Workaround

☺

Jakar510 added the (Bug) label on Aug 12

Jakar510 added this to **Needs triage** in **Triage** via (automation) on Aug 12

Jakar510 moved this from **Needs triage** to **High priority** in **Triage** on Aug 12

Jakar510 added the (High Priority) label on Aug 12

Assignees ⚙

No one—assign yourself

Labels ⚙

(Bug) (High Priority)

Projects ⚙

☷ Triage

High priority ▾

Milestone ⚙

No milestone

Linked pull requests ⚙

1 participant

⚲ Pin issue ⓘ

💬 Convert to discussion