Rate My App Guide

Overview

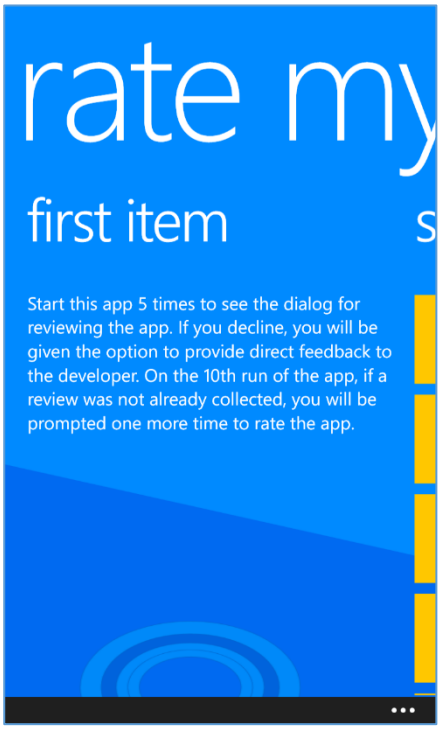
This document describes how to use Rate My App component to create prompts that appear at set intervals and allow the user to provide feedback and rate the application in the Windows Phone Store. This guide details important features of the Rate My App component and how to integrate it into your own applications, which can be as simple as adding one control to the main page of the application.

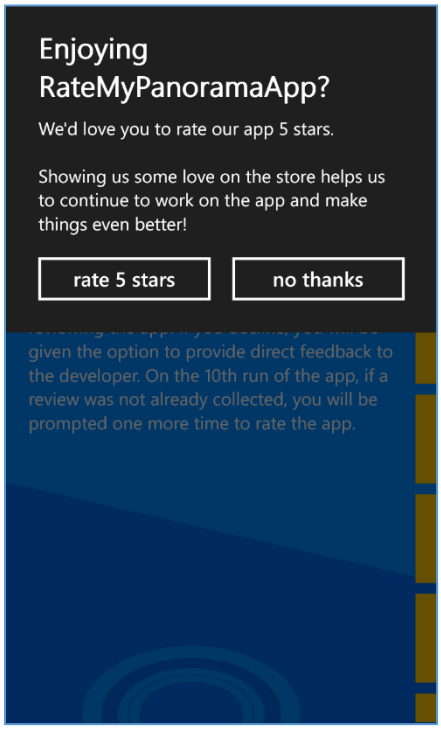
By default, when the application with Rate My App component is started for the 5th time, a dialog for reviewing the app is shown to the user. If the user declines to review the app, she will be given the option to provide direct feedback to the developer. On the 10th run of the app, if a review was not already collected, the user will be prompted one more time to rate the app. The interval of showing the dialogs, as well as the precise textual content of the dialogs can be configured to better suit your needs.

Rate My App component is available as an installable NuGet package, and the full source code for Rate My App component and accompanying demo applications is available in Nokia Developer GitHub repository (https://github.com/nokia-developer/rate-my-app). Using the source code in conjunction with this guide is recommended.

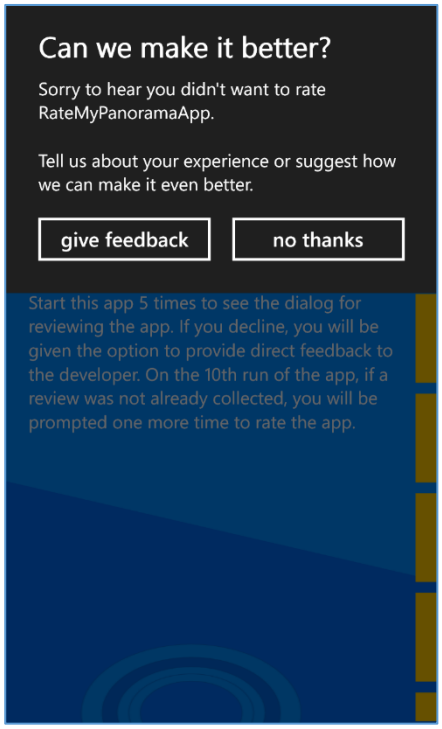
User flow

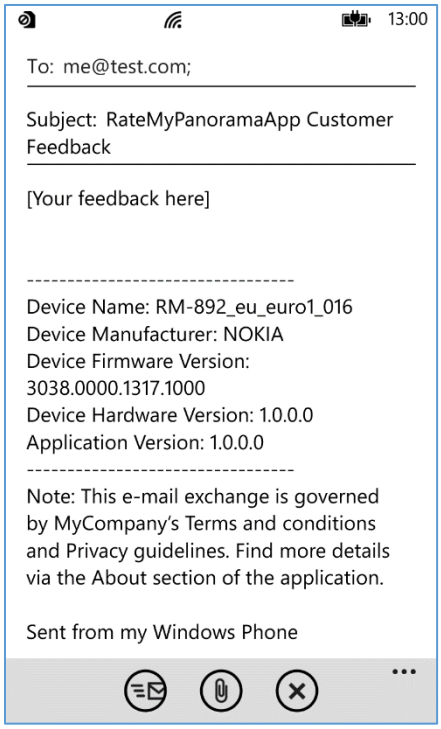
In the RateMyPanoramaApp demo application (available in the source code) only the main page of the application is shown on 1st to 4th launch of the application. On the 5th app launch, a dialog is shown on top of the main page asking user to review the app.



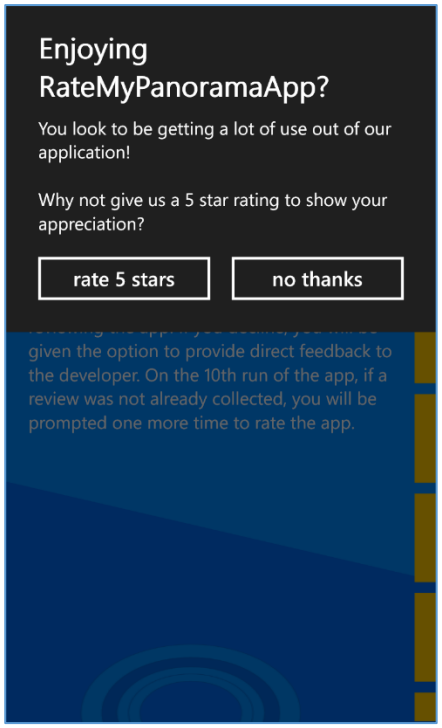


If the user chooses to rate the application, she is directed to the store to rate the app (which in the case of the demo apps gives an error because there is no real app to be rated in the store), and no more dialogs are shown. Otherwise, if the user selects “no thanks”, another dialog appears, asking for feedback via email. If the user chooses to give feedback, the ‘ComposeEmailTask’ is launched, otherwise the dialog closes.



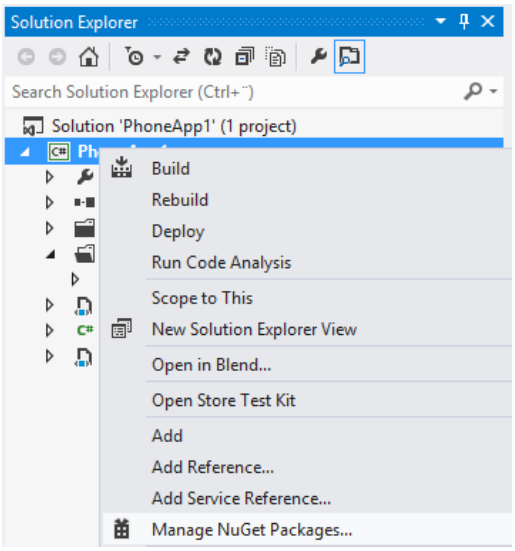


Dialogs are not shown again until the application is launched for the 10th time, when a final dialog is displayed to the user asking if they want to review the app:

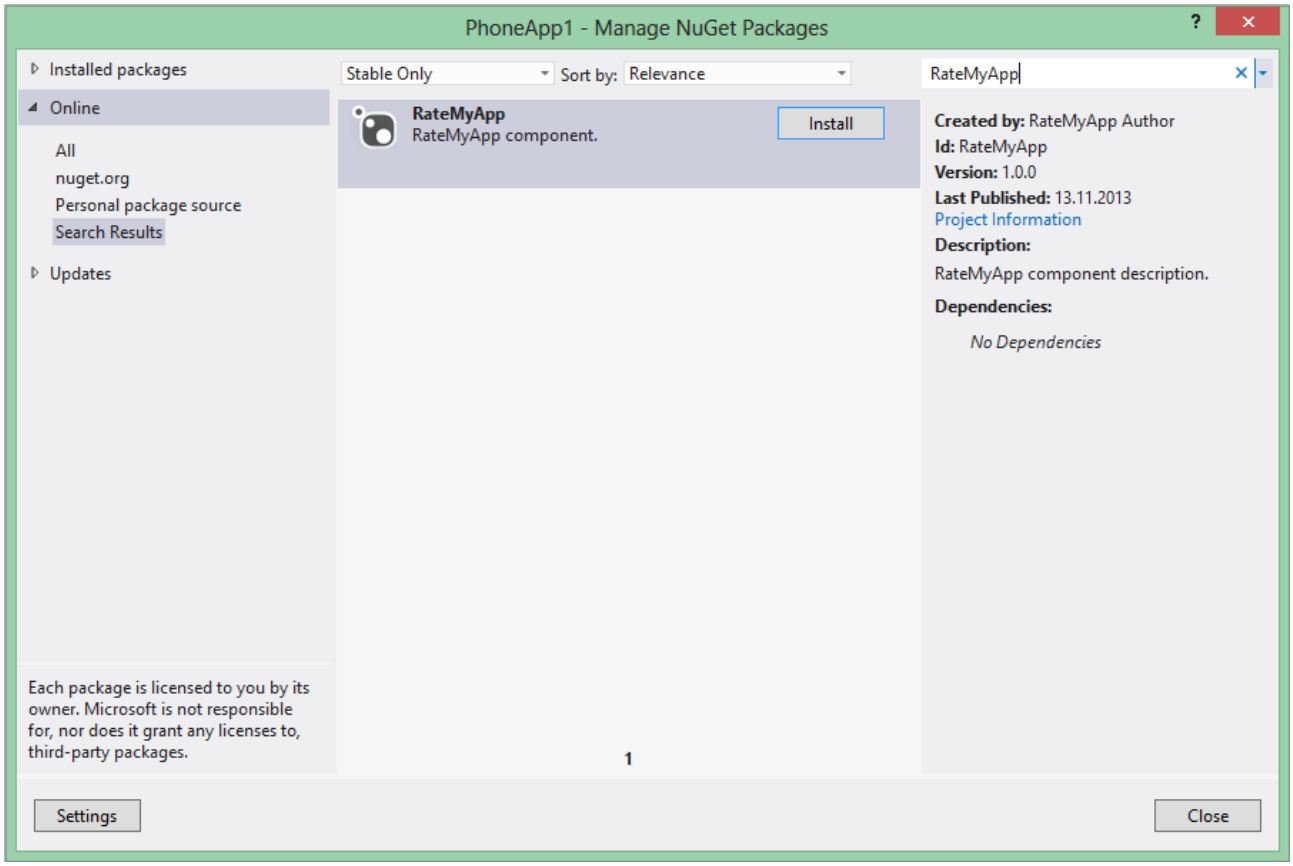


Integrating Rate My App component

Integrating Rate My App component to a Windows Phone application is very easy. First you should install the Rate My App component to your application using NuGet Package Manager. Right click your application project in Solution Explorer and select “Manage NuGet Packages…” menu item.



In the “Manage NuGet Packages” dialog, enter “RateMyApp” to the search box on the top-right corner, wait for search to find RateMyApp component, and press the “Install” button.



A namespace for RateMyApp component must then be declared inside PhoneApplicationPage definition on the first page in the app. 'FeedbackOverlay' user control from the RateMyApp component should then be placed as the last element inside the layout grid of the first page, and it should span all layout grid rows and columns so it is not obscured:

```
<phone:PhoneApplicationPage
    ...
    xmlns:rma="clr-namespace:RateMyAppLib.Controls;assembly=RateMyAppLib"
    ...>

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">

        ...

        <!-- Feedback overlay -->
        <ctrl:FeedbackOverlay x:Name="FeedbackOverlay"
                              Grid.RowSpan="2"
                              FeedbackTo="me@test.com"
                              ApplicationName="MyApp"
                              CompanyName="MyCompany"/>

    </Grid>
</phone:PhoneApplicationPage>
```

Customizing Rate My App component

Rate My App component can be customized in several ways. You can change texts shown in the dialogs and provide your own localizations for the texts shown in the dialogs, you can set the intervals when dialogs are shown, and you can set whether each and every application launch should be counted against the set interval or only one launch per day.

The following table describes the properties of the 'FeedbackOverlay' control and their default values.

Property	Default value	Notes
EnableAnimation	true	Dialog flip animation
IsVisible	false	
IsNotVisible	true	
RatingTitle	Enjoying {0}?	{0} is replaced with ApplicationName property.
RatingMessage1	We'd love you to rate our app 5 stars. Showing us some love on the store helps us to continue to work on the app and make things even better!	
RatingMessage2	You look to be getting a lot of use out of our application! Why not give us a 5 star rating to show your appreciation?	
RatingYes	rate 5 stars	
RatingNo	no thanks	
FeedbackTitle	Can we make it better?	

FeedbackMessage1	Sorry to hear you didn't want to rate {0}. Tell us about your experience or suggest how we can make it even better.	{0} is replaced with ApplicationName property.
FeedbackYes	give feedback	
FeedbackNo	no thanks	
FeedbackTo	null	Mandatory. Email address for sending feedback to.
FeedbackSubject	{0} Customer Feedback	{0} is replaced with ApplicationName property.
FeedbackBody	[Your feedback here] ---------- Device Name: {0} Device Manufacturer: {1} Device Firmware Version: {2} Device Hardware Version: {3} Application Version: {4} ---------- Note: This e-mail exchange is governed by {5}'s Terms and conditions and Privacy guidelines. Find more details via the About section of the application.	{0} is replaced with device name. {1} is replaced with device manufacturer. {2} is replaced with firmware version number. {3} is replaced with hardware version number. {4} is replaced with application version number. {5} is replaced with CompanyName property.
ApplicationName	null	Mandatory. Name of the application.
CompanyName	null	Mandatory. Name of the company.
FirstCount	5	After how many launches the 1st review and feedback dialogs are shown.
SecondCount	10	After how many launches the 2nd review dialog is shown.
CountDays	false	True if only one launch per day is to be counted against FirstCount and SecondCount.
LanguageOverride	null	Culture code supported by the RateMyApp component ("de" or "sl-SI", for example).

Note that the properties include three mandatory strings (ApplicationName, CompanyName and FeedbackTo) which should be defined in order to replace placeholders in other string properties, for example in RatingTitle.

Below you can see an example of customized FeedbackOverlay:

```
<phone:PhoneApplicationPage
    ...>

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        ...

        <ctrl:FeedbackOverlay Grid.RowSpan="2"
                              EnableAnimation="True"
                              RatingTitle="RatingTitle"
                              RatingMessage1="Rating Message 1"
                              RatingMessage2="Rating Message 2"
                              RatingYes="Yes"
                              RatingNo="No"
                              FeedbackTitle="FeedbackTitle"
                              FeedbackMessage1="Feedback Message 1"
                              FeedbackYes="Yes"
                              FeedbackNo="No"
                              FeedbackTo="feedback@company.com"
                              FeedbackSubject="Feedback Subject"
                              FeedbackBody="Feedback Body"
                              CompanyName="MyCompany"
                              ApplicationName="MyApplication"
                              FirstCount="5"
                              SecondCount="10"
                              CountDays="False"
                              />
    </Grid>
</phone:PhoneApplicationPage>
```

Localization

Rate My App component has built-in support for the following cultures:

Culture code	Culture
de	German
en-US	English (United States) (default)
en-GB	English (United Kingdom)
ro	Romanian
sl-SI	Slovenian (Slovenia)
zh-CN	Chinese (Simplified, PRC)

We aim to provide localized text for all the languages supported by the Windows Phone platform, but currently only a subset is available. The list of currently supported languages can also be found in the release notes document. Please visit http://www.getlocalization.com/ratemyapp/ in order to contribute your own translation improvements to the Rate My App component.

You can override default texts by providing your own texts as hardcoded strings or use localized resources:

```
<ctrl:FeedbackOverlay FeedbackTo="feedback@company.com" />
```

or

```
<ctrl:FeedbackOverlay FeedbackTo="{Binding Path=LocalizedResources.FeedbackTo,
Source={StaticResource LocalizedStrings}}" />
```

It should be noted that a default localization provided with the Rate My App component requires that the application supports the same culture as well. For example, let's assume that the phone language has been set to German. In order for the review and feedback notifications to be shown in German, make sure the application supports German language by including a resource file for German (AppResources.de.resx).

Be aware that setting the LanguageOverride property to a specific language overrides explicit definitions of string properties with a closest matching string resource of RateMyApp component. It means that even if you have defined, for example, the 'RatingTitle' to be "Title" and you set 'LanguageOverride' property to be "de", the 'RatingTitle' will be either "Gefällt Ihnen {0}?" or "Enjoying {0}?", depending on whether the app supports German culture or not.

Visibility control

´IsVisible´ and ´IsNotVisible´ dependency properties of 'FeedbackOverlay' user control are included to help with MVVM apps and can be useful for showing and hiding the (bindable) application bar.

In RateMyPanoramaApp featured earlier in this document, the following code is used to control the visibility of application bar:

```
public partial class MainPage : PhoneApplicationPage
{
    // Constructor
    public MainPage()
    {
        ...

        BuildApplicationBar();

        FeedbackOverlay.VisibilityChanged += FeedbackOverlay_VisibilityChanged;
    }

    void FeedbackOverlay_VisibilityChanged(object sender, EventArgs e)
    {
        ApplicationBar.IsVisible = (FeedbackOverlay.Visibility != Visibility.Visible);
    }

    ...
}
```

As already instructed, the 'FeedbackOverlay' user control should be placed as the last element inside the layout grid of the first page and span all rows and columns. This way the 'FeedbackOverlay' control fills the whole page when it is visible, and the controls underneath it cannot be interacted with.

Resetting the counter

You can reset the review and feedback counter of RateMyApp component by calling the Reset() method of the FeedbackOverlay component. This kind of behavior could be justified, for example, when a new version of the application with lots of new or modified functionality is made available in Windows Phone Store.

Modifying the Rate My App component

In addition to Rate My App component being available from the NuGet repository, the source code for Rate My App component is available in Nokia Developer GitHub repository at https://github.com/nokia-

developer/rate-my-app. You are free to modify it and implement additional functionality to make it more suitable for your own needs.

Rate My App Demos

The source code includes several demo applications to further help you to understand how to integrate the Rate My App component for various different kinds of applications:

- Windows Phone App for WP8
- Windows Phone Panorama App for WP8
- Windows Phone XAML and Direct3D App for WP8
- Windows Phone App for WP7
- Windows Phone XAML and XNA App for WP7

Despite the differences in application styles, all the demo applications integrate Rate My App component in exactly the same manner, with a 'FeedbackOverlay' control definition as the last control in the layout grid of the first page in the application. Reset functionality is made available using a menu item for resetting the counter.