Mobile Application Analytics

Windows Phone 8 SDK Instructions

SDK version 3.1.2
Updated: 1/23/12

Welcome to Flurry Analytics!

This file contains:

---

1. Introduction

The Flurry Windows Phone 8 Analytics Agent allows you to track the usage and behavior of your Windows Phone application on users' phones for viewing in the Flurry Analytics system. It is designed to be as easy as possible with a basic setup complete in under 5 minutes.

This archive should contain these files:

- **FlurryWP8SDK.dll**: The library containing Flurry's collection and reporting code.
- **ProjectApiKey.txt**: This file contains the name of your project and your project's API key.
- **Analytics-README.pdf** : This PDF file containing instructions.
- **Analytics-RELEASENOTES.pdf** : A PDF file containing information about this release.

---

2. Developer Portal

The portal for analytics can be accessed at dev.flurry.com. To enable analytics for a Windows Phone 8 project, the first step is create a new application on the dev portal site.

This SDK remains compatible with portal applications previously created for prior versions of Windows Phone.

---

2. Integration

To integrate Flurry Analytics into your Windows Phone 8 application:

1. Reference **FlurryWP8SDK.dll** in your application project.

Note: this version of the SDK is backwards compatible with Windows Phone 7 applications.

Also note: if you encounter the error "A reference to a higher version or incompatible assembly cannot be added to the project," you first might need to unblock the dll file by right-clicking the file, selecting Properties, and - next to where it reads "Security: This file came from another computer and might be blocked to help protect this computer." - clicking "Unblock" and saving the change.

2. Configure WMAppManifest.xml:
Required Capability:

`ID_CAP_NETWORKING`
Required to send analytics data back to the flurry servers

`ID_CAP_IDENTITY_DEVICE`
Required for unique device reporting.

`ID_CAP_IDENTITY_USER`
Required for unique user reporting (by accessing the ANID).

Specify the Version attribute in the manifest to have data reported under that version.

3. Add lifecycle calls

- Insert a call to `FlurryWP8SDK.Api.StartSession(string apiKey)` to begin a session, passing it your project's API key. It is recommended to insert this call after the application is launched or reactivated. If you don't intend to explicitly end the session before application closes, this is all you need to do.

- `FlurryWP8SDK.Api.EndSession()` is already associated with the Closing function, and `FlurryWP8SDK.Api.PauseSession()` is already associated with the Deactivated function, so there is no need to make explicit calls to these functions in your application unless you need to call EndSession explicitly before the application terminates.

- If you have an active session, make sure StartSession is called when the application is reactivated to resume the current session. Session length, usage frequency, events and errors will continue to be tracked as part of the same session. If the application is reactivated after more than 10 seconds of deactivation, the current session will be ended and a new one will be started. If you wish to change the window during which a session can be resumed, call `FlurryWP8SDK.Api.SetSessionContinueSeconds(int seconds)` after the session has started.

- If you want to track Activity usage, we recommend using LogEvent, described below.

Note: Windows Phone 8 projects that are fully native Direct3D C++ (as detailed here: http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj662943(v=vs.105).aspx#BKMK_NativeonlyApplications) cannot support the Flurry SDK dll.

Instead, a solution that is a combination of C# with XAML projects and Direct3D C++ (such as one based on the "Windows Phone Direct 3D with XAML App" project template in Visual Studio or any of the managed solution types detailed here: http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj662943(v=vs.105).aspx#BKMK_MixedmodeApplications) may be used.

With such solutions, the start session call should be made within the C# XAML project that is part of the solution, when launching and when activating the app, as detailed above. Other Flurry API calls may be made, as well, from this part of the solution, although the Flurry Windows Phone 8 SDK dll will not be referenceable from any C++ project in the solution and Flurry API calls may not be made directly from any C++ parts of the code.

4. Build your application normally.

You're done! That's all you need to do to begin receiving basic metric data.

---

3. Optional Features

- `FlurryWP8SDK.Api.LogEvent(string eventId, bool timed, List<Parameter> parameters)`
  Use LogEvent to track user events that happen during a session. You can track how many times each event occurs, what order events happen in, as well as what the most common parameters are for each event. This can be useful for measuring how often users take various actions, or what sequences of actions they usually perform. Each project supports a maximum of 100 events. The timed argument and the parameters argument are both optional. Each event id, parameter key, and parameter value must be no more than 255 characters in length. Each event can have no more than 10 parameters. If the timed argument is true, that means you are logging a timed event, which can be ended by calling the `FlurryWP8SDK.Api.EndTimedEvent(string eventId, List<Parameter> parameters)` function using the same eventid used to start the timed event.

- `FlurryWP8SDK.Api.LogError(String message, Exception exception)`
  Use LogError to report application errors. Flurry will report the last 10 errors to occur in each session. (max length 255 chars)

Optional configuration methods
Call these methods after calling `StartSession` to change the configuration:

- `FlurryWP8SDK.Api.SetVersion(string versionName)`
  To change the version name your analytic data is reported under. If this is not specified, the version name is retrieved from the application descriptor.

- `FlurryWP8SDK.Api.SetUserId(string userId)`
  `FlurryWP8SDK.Api.SetAge(int age)`
  `FlurryWP8SDK.Api.SetGender(Gender gender)`
  `FlurryWP8SDK.Api.SetLocation(double latitude, double longitude, float accuracy)`
  To record demographic data about the user.

- `FlurryWP8SDK.Api.SetSessionContinueSeconds(int seconds)`
  Pass a value to change the number of seconds for which paused sessions will be continued. After this amount of time has passed with no activity, a new session is assumed to have started.

Please let us know if you have any questions. If you need any help, just email winmosupport@flurry.com!

Cheers,

The Flurry Team
http://www.flurry.com
winmosupport@flurry.com