

# AppMetrica

AppMetrica

10.06.2024

**Y**andex

AppMetrica. AppMetrica. Version 1.0

Document build date: 10.06.2024

This volume is a part of Yandex technical documentation.

© 2008—2024 Yandex LLC. All rights reserved.

## Copyright Disclaimer

Yandex (and its applicable licensor) has exclusive rights for all results of intellectual activity and equated to them means of individualization, used for development, support, and usage of the service AppMetrica. It may include, but not limited to, computer programs (software), databases, images, texts, other works and inventions, utility models, trademarks, service marks, and commercial denominations. The copyright is protected under provision of Part 4 of the Russian Civil Code and international laws.

You may use AppMetrica or its components only within credentials granted by the Terms of Use of AppMetrica or within an appropriate Agreement.

Any infringements of exclusive rights of the copyright owner are punishable under civil, administrative or criminal Russian laws.

## Contact information

Yandex LLC

<https://www.yandex.com>

Tel.: +7 495 739 7000

Email: [pr@yandex-team.ru](mailto:pr@yandex-team.ru)

16 L'va Tolstogo St., Moscow, Russia 119021

# Contents

Windows.....	4
Installation and initialization.....	4
Step 1. Integrating the library.....	4
Step 2. Initializing the library.....	4
Step 3. Setting up access permissions.....	5
.....	5
Reference.....	5
Yandex.Metrica.....	5
Analysis of app crashes.....	13
.....	13
Changelog.....	13
Version 3.5.2.....	13
Version 3.5.1.....	13
Version 3.5.0.....	13
Version 3.4.2.....	13
Version 3.4.1.....	13
Version 3.4.0.....	14
Version 3.3.0.....	14
Version 3.2.2.....	14
Version 3.2.1.....	14
Version 3.2.0.....	14
Version 3.1.....	14
Version 2.2.....	15
Version 2.1.....	15
Version 2.0.....	15
Version 1.6.....	15

---

# Windows

## Installation and initialization



**Attention:** We discontinued the development of new versions of the AppMetrica Windows SDK.

The AppMetrica SDK can be used for tracking mobile and desktop applications on Windows.

### Step 1. Integrating the library

The AppMetrica library is provided as a [NuGet package](#).

To connect the library, use the graphical interface or the command line of the NuGet package manager:

```
Install-Package Yandex.Metrica -Version 3.5.1
```

### Step 2. Initializing the library



**Attention:** In the desktop application, before initializing the library, specify a folder for storing service files using the [YandexMetricaFolder.SetCurrent\(string path\)](#) method. The maximum volume of locally stored data is 2 MB.

Initialize the library in the app using one of the methods shown below:

#### In the App.xaml file

Make changes to the Application item of the App .xaml file:

**WP 7, WP 8**

```
<Application
  ...
  xmlns:metrica="clr-namespace:Yandex.Metrica;assembly=Yandex.Metrica">
  ...
  <Application.Resources>
    ...
    <metrica:YandexMetricaActivator
      x:Key="String"
      ApiKey="API_key" />
    ...
  </Application.Resources>
</Application>
```

**RT 8, UWP 8.1, UWP 10**

```
<Application
  ...
  xmlns:metrica="using:Yandex.Metrica">
  ...
  <Application.Resources>
    ...
    <metrica:YandexMetricaActivator
      x:Key="String"
      ApiKey="API_key" />
    ...
  </Application.Resources>
</Application>
```

The library starts sending events after the app is launched.

#### In the application code

Initialize the library in the application code using the method:

```
YandexMetrica.Activate(string API_key)
```

The library begins sending events after the method is called.

#### What is the API key?

The *API key* is a unique application identifier that is issued in the AppMetrica web interface during [app registration](#).

Make sure you have entered it correctly.

The screenshot shows the 'Settings' pane for an application named 'MyApp'. The 'API key' field is highlighted with a red box and contains the value '12345678-6aab-4b78-bca6-c69a8937950b'. Other fields include 'Application name' (MyApp) and 'Application ID' (1111).

### Step 3. Setting up access permissions

Configure the app's Capabilities depending on the platform.

**Note:** For .NET applications setting permissions is not necessary.

#### WP 7, WP 8

Add to the Properties/WMApManifest.xml file the following permissions:

```
<Capabilities>
  ...
  <Capability Name="ID_CAP_IDENTITY_DEVICE" />
  <Capability Name="ID_CAP_LOCATION" />
  <Capability Name="ID_CAP_NETWORKING" />
  ...
</Capabilities>
```

#### RT 8, UWP 8.1, UWP 10

Add to the Package.appxmanifest file the following permissions in code mode:

```
<Capabilities>
  ...
  <Capability Name="internetClient" />
  <DeviceCapability Name="location" />
  ...
</Capabilities>
```

#### Contact support

If you didn't find the answer you were looking for, you can use the feedback form to submit your question. Please describe the problem in as much detail as possible. Attach a screenshot if possible.

## Reference

### Yandex.Metrica

#### Classes

#### YandexMetrica class



**Attention:** We discontinued the development of new versions of the AppMetrica Windows SDK.

yandex.metrica

```
public class YandexMetrica
```

Methods of the class are used for configuring the library.

## Nested classes

class	<a href="#">YandexMetrica.YandexMetricaConfig</a>
class	<a href="#">YandexMetrica.Location</a>

## Methods

void [Activate](#)(string apiKey)  
Инициализирует библиотеку в приложении с указанным [API key](#).

void [Activate](#)(Guid apiKey)  
Инициализирует библиотеку в приложении с указанным [API key](#).

### [YandexMetrica.YandexMetricaConfig](#) [Config](#)

Метод для доступа к вложенному классу [YandexMetrica.YandexMetricaConfig](#).

void [ReportError](#)(string message, Exception )  
Отправляет сообщение об ошибке.

void [ReportEvent](#)(string eventName)  
Отправляет сообщение о событии.

void [ReportEvent](#)(string eventName, string jsonData)  
Отправляет сообщение о событии в формате JSON в виде строки.

void [ReportEvent](#)<TItem>(string eventName, TItem serializableItem)  
Отправляет сообщение о событии с дополнительными параметрами.

void [ReportLaunchUri](#)(Uri uri)  
Отправляет сообщение об открытии приложения с помощью deeplink.

void [ReportUnhandledException](#)(Exception exception)  
Отправляет сообщение о необработанном исключении.

void [Snapshot](#)()  
Кэширует неотправленные события в памяти устройства.

### *Method descriptions*

#### **Activate**

```
public static void Activate(string apiKey)
```

Initializes the library in the application with the specified [API key](#).

Use this method to initialize the library directly from the application code.



**Parameters:**

eventName	Short name or description of the event
serializableItem	The value of the event. It is converted to a JSON string by the <code>DataContractJsonSerializer</code> system class.

**ReportLaunchUri**

```
public static void ReportLaunchUri(Uri uri)
```

Sends a message about the app launching from a deeplink.

**Parameters:**

uri	The instance of the <a href="#">Uri</a> system class that contains a deeplink.
-----	--

**ReportUnhandledException**

```
public static void ReportUnhandledException(Exception exception)
```

Sends a message about an unhandled exception.

To avoid exception events duplication, set the `false` value for the [YandexMetrica.Config.CrashTracking](#) property.

**Parameters:**

exception	The instance of the <a href="#">Exception</a> system class.
-----------	---

**Snapshot**

```
public static void Snapshot()
```

Caches unsent events in the device memory.

This method must be called before shutting down C++ applications or background tasks.

The maximum volume of locally stored data is 2 MB.

**YandexMetrica.Location class**

**Attention:** We discontinued the development of new versions of the AppMetrica Windows SDK.

```
yandex.metrica
```

```
public class YandexMetrica.Location
```

This class contains information about the device location.

**Properties**

double	<a href="#">Lat</a>	Sets/returns the longitude.
double	<a href="#">Lon</a>	Sets/returns the latitude.
ulong	<a href="#">Timestamp</a>	Sets/returns the time in milliseconds from January 1, 1970.
uint	<a href="#">Precision</a>	Sets/returns the accuracy of location detection.
uint	<a href="#">Direction</a>	Sets/returns the direction in degrees relative to the North in a clockwise direction.
uint	<a href="#">Speed</a>	Sets/returns the speed in meters per second.



int [Altitude](#) Sets/returns altitude in meters.

### *Property descriptions*

#### **Lat**

```
public double Lat { get; set; }
```

Sets/returns the longitude.

#### **Lon**

```
public double Lon { get; set; }
```

Sets/returns the latitude.

#### **Timestamp**

```
public ulong Timestamp { get; set; }
```

Sets/returns the time in milliseconds from January 1, 1970.

#### **Precision**

```
public uint Precision { get; set; }
```

Sets/returns the accuracy of location detection.

#### **Direction**

```
public uint Direction { get; set; }
```

Sets/returns the direction in degrees relative to the North in a clockwise direction.

#### **Speed**

```
public uint Speed { get; set; }
```

Sets/returns the speed in meters per second.

#### **Altitude**

```
public int Altitude { get; set; }
```

Sets/returns altitude in meters.

#### **YandexMetrica.YandexMetricaConfig class**



**Attention:** We discontinued the development of new versions of the AppMetrica Windows SDK.

```
yandex.metrica
```

```
public class YandexMetrica.YandexMetricaConfig
```

This class contains the extended startup configuration for the library.

#### **Methods**

```
void SetCustomLocation(YandexMetrica.Location location)
```

Задает собственную информацию о местоположении устройства.



Default value:

- For mobile apps is a package name.
- For desktop applications is a name of the main build.

### **CustomAppVersion**

```
public Version CustomAppVersion { get; set; }
```

Sets/returns the app version.

The default version is the app version specified in the `AssemblyInfo.cs` file for the application's main build.

### **HandleFirstActivationAsUpdate**

```
public bool HandleFirstActivationAsUpdate { get; set; }
```

Sets/returns a flag indicating that the first launch of the app is an update (not an installation).

The default value is `false`.

Possible values:

- `true` — The first launch is defined as an update.
- `false` — The first launch is defined as a new installation.

### **LibraryVersion**

```
public Version LibraryVersion { get; }
```

Returns the current version of the AppMetrica library.

### **LocationTracking**

```
public bool LocationTracking { get; set; }
```

Sets/returns a flag indicating the status of device location tracking.

The default value is `true`.

Possible values:

- `true` — Monitoring unhandled exceptions is enabled.
- `false` — Monitoring unhandled exceptions is disabled.

### **OfflineMode**

```
public bool OfflineMode { get; set; }
```

Sets/returns a flag indicating the status of sending statistics to the AppMetrica server.

The default value is `false`.

Possible values:

- `true` — Sending statistics is disabled.
- `false` — Sending statistics is enabled.

### **SessionTimeout**

```
public TimeSpan SessionTimeout { get; set; }
```

Sets/returns the session timeout in seconds.

The default value is 10 (minimum allowed value).

### YandexMetricaActivator class



**Attention:** We discontinued the development of new versions of the AppMetrica Windows SDK.

```
yandex.metrica
```

```
public class YandexMetricaActivator
```

The class is used for initializing the library from App.xaml. For more information, see [Installation and initialization](#).

### Properties

string [ApiKey](#) Sets/returns [API key](#) of the application

*Property descriptions*

#### ApiKey

```
public string ApiKey { get; set; }
```

Sets/returns [API key](#) of the application

### YandexMetricaFolder class



**Attention:** We discontinued the development of new versions of the AppMetrica Windows SDK.

```
yandex.metrica
```

```
public class YandexMetricaFolder
```

The class method is used to set the path to the AppMetrica service folder. It stores files that are required for the library.

This class is only available for desktop applications.

### Methods

```
void SetCurrent(string path)
```

Устанавливает путь к папке, в которой будут храниться служебные файлы AppMetrica.

### Properties

string [Current](#) Returns the path to the folder where AppMetrica service files are stored.

*Method descriptions*

#### SetCurrent

```
public static void SetCurrent(string path)
```

Sets the directory for storing AppMetrica files.

The app must have access rights to read this directory and write files to it. Otherwise, the library won't function correctly

#### Parameters:

path The path to the folder for storing AppMetrica service files.

### *Property descriptions*

#### **Current**

```
public static string Current { get; }
```

Returns the path to the folder where AppMetrica service files are stored.

## **Analysis of app crashes**



**Attention:** We discontinued the development of new versions of the AppMetrica Windows SDK.

The library does not symbolicate or deobfuscate app crashes. These operations are performed on the server or on the client side.

#### **Related information**

[Windows reference](#) on page 5

#### **Contact support**

If you didn't find the answer you were looking for, you can use the feedback form to submit your question. Please describe the problem in as much detail as possible. Attach a screenshot if possible.

## **Changelog**



**Attention:** We discontinued the development of new versions of the AppMetrica Windows SDK.

### **Version 3.5.2**

Released 20 February 2019

- Fixed errors on the HoloLens and IoT platforms.

### **Version 3.5.1**

Released 23 October 2017

- Improved stability and reliability of the library.

### **Version 3.5.0**

Released 8 September 2017

- Improved stability and reliability of the library.
- Updated the name of the configuration class
- Fixed bugs.

### **Version 3.4.2**

Released 19 January 2017

- Fixed potential errors when sending events.
- Improved how the `static void Snapshot()` method works for background tasks.

### **Version 3.4.1**

Released 9 January 2017

- Fixed minor bugs.

## Version 3.4.0

Released 20 December 2016

- Added [tracking app openings with deeplinks](#).
- Now you can inform AppMetrica that the app was installed on the device before the AppMetrica SDK library was enabled. This allows AppMetrica to distinguish the first app start after integrating the SDK from the very first app start. This type of app launch won't be interpreted as a new user. [Learn more](#).
- Added the ability to [set the library's working directory](#) for storing internal files.
- Updated the data transfer protocol.
- Fixed bugs.

## Version 3.3.0

Released 12 November 2016

- Added the [CustomAppId](#) property.
- Added the [static void Snapshot\(\)](#) method as a trigger for saving the AppMetrica state.
- Fixed bugs.

## Version 3.2.2

Released 31 October 2015

- Improved stability and reliability of the library.

## Version 3.2.1

Released 17 October 2015

- Added support for background tasks.
- Optimized performance.
- Fixed bugs.

## Version 3.2.0

Released 21 September 2016

- Introduced a new format for numbering versions.
- Internal changes to JSON serialization.
- Added a self-repair feature for the AppMetrica library to recover functionality if system files are damaged.

## Version 3.1

Released 16 August 2016

- The app version number uses the value specified in the `AssemblyInfo.cs` field in the application's main build if the [CustomAppVersion](#) property isn't set.
- The application ID uses the value specified in the `AssemblyInfo.cs` file for the application's main build.
- Reduced the size of builds.
- Optimized the internal logic of the SDK.
- Reduced the delay when launching AppMetrica.
- Added support for Windows Phone 7.
- Changed the following methods and properties:
  - `void SetTrackLocationEnabled(bool enabled)`. Use the [bool Config.LocationTracking { get; set; }](#) property.
  - `void SetLocation(double latitude, double longitude)`. Use the method [static void SetCustomLocation\(YandexMetrica.Location location\)](#).
  - `void SetReportCrashesEnabled(bool enabled)`. Use the [bool Config.CrashTracking { get; set; }](#) property.
  - `bool ServerInteractionEnabled { get; set; }`. Use the [static bool Config.OfflineMode { get; set; }](#) property.
- Deleted the `void ReportExit()` method.

## Version 2.2

Released 12 May 2016

- Added support for Windows 10.
- Fixed the SDK crash for Windows Phone Silverlight when minimizing an app.
- Performed an internal update of the protocol.
- Fixed minor bugs.

## Version 2.1

Released 17 December 2015

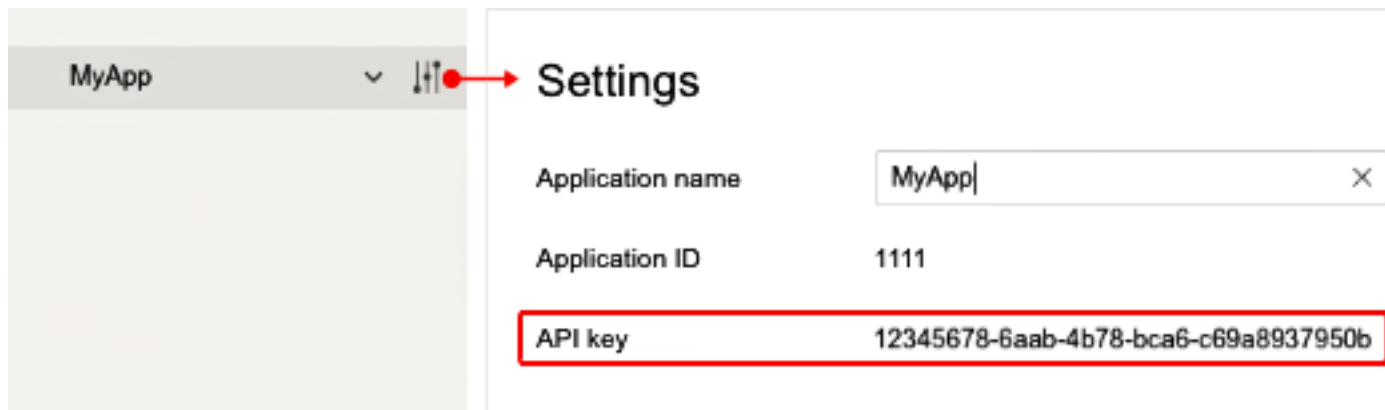
- Optimized performance.
- Improved quality of statistics collection.

## Version 2.0

Released 25 September 2015

- The format of "Api key" has changed. The new format of "Api key" is available in the [AppMetrica web interface](#) in app editing mode.

### Where to find ApiKey



- The library initialization method in the app has been renamed from `YandexMetrica.Start(uint apiKey)` to `YandexMetrica.Activate(string apiKey)`.
- Changed the length of the session timeout. Now it is 10 seconds, by default.
- Deprecated methods and properties have been deleted.

### Click to see the list

`static void StartNewSessionManually()` — Starts a new session manually.

`static void SendEventsBuffer()` — Sends all accumulated events without waiting for them to automatically be sent to the server.

`bool ReportsEnabled { get; set; }` — Lets you enable and disable sending reports.

`static TimeSpan DispatchPeriod { get; set; }` — Sets the interval in seconds between sending accumulated events to the server.

`static uint MaxReportsCount { get; set; }` — Sets the maximum number of events that can be stored up before sending all accumulated events to the server.

## Version 1.6

Released 29 October 2014

- Renamed public classes: `Counter` was renamed to `YandexMetrica`, and `CounterBootstrapper` was renamed to `Bootstrapper`.
- Added events with additional parameters.
- Added support for Windows 8.1 (Phone and RT).

- Discontinued support for Windows Phone 7.1.