|  |  |
| --- | --- |
| Video Telemetry  A Solution in Azure | Abstract  We present a Video Telemetry solution in Azure, based on Application Insights. This solution can reach all major client platforms (iOS, Android, Windows and web browsers) via various SDKs. Video Telemetry data can be accessed via Azure portal, Power BI and an ad hoc query tool.  William Zhang, Amit Rajput, Vishal Sood  Microsoft |

Video Telemetry: A Solution in Azure

# Introduction

Microsoft offers an expansive platform for building online video streaming/multi-screen solutions, consisting of the following:

|  |  |
| --- | --- |
| Video Technology Category | Microsoft Offering |
| Video Processing/Streaming Services | [Azure Media Services](https://azure.microsoft.com/en-us/services/media-services/) |
| CDN | [Azure CDN](https://azure.microsoft.com/en-us/services/cdn/) |
| DRM | [PlayReady Product Suite](https://www.microsoft.com/PlayReady/features/) |
| Native Client | Windows 10 media stack including   * [Native DASH and HLS support for UWP apps](ttps://blogs.msdn.microsoft.com/ie/2015/01/29/simplified-adaptive-video-streaming-announcing-support-for-hls-and-dash-in-windows-10/), * [PlayReady 3.0 client](https://www.microsoft.com/PlayReady/features/EndPoints.aspx), * UHD/HEVC/[HW-DRM](https://msdn.microsoft.com/en-us/library/windows/apps/mt429380.aspx), * [Universal Smooth Streaming Client SDK](https://visualstudiogallery.msdn.microsoft.com/1e7d4700-7fa8-49b6-8a7b-8d8666685459?SRC=Home), * [Microsoft Media Foundation](https://msdn.microsoft.com/en-us/library/windows/desktop/ms694197(v=vs.85).aspx). |
| Browser Client | * [MSE/EME/CDM for modern browsers](https://blogs.windows.com/msedgedev/2015/01/29/simplified-adaptive-video-streaming-announcing-support-for-hls-and-dash-in-windows-10/)/DASH * [Native support of DASH and HLS in MS Edge](https://blogs.msdn.microsoft.com/ie/2015/01/29/simplified-adaptive-video-streaming-announcing-support-for-hls-and-dash-in-windows-10/) * [Azure Media Player](https://azure.microsoft.com/en-us/services/media-services/media-player/) |
| Console Client | [Xbox](http://www.xbox.com/) |

However, we do not yet have an offering for Video Telemetry/user analytics which can reach all of the popular client platforms. Today, our customers are using 3rd party products at different price ranges. For example:

1. High-end video telemetry, such as [Conviva](http://www.conviva.com/), as used in NBC Sports/Olympics solution;
2. Mid-range video telemetry, such as [Google Analytics Premium](https://www.google.com/analytics/premium/);
3. Low-end/free video telemetry, such as [Google Analytics](https://www.google.com/analytics/).

This document provides a summary of the requirements, design and how-to.

## Pricing

This solution consists of client SDKs, Azure backend and Video Telemetry code file. The Client SDKs are [Application Insights Client SDKs](https://github.com/Microsoft/ApplicationInsights-Home) targeting various platforms, which are open sourced. For its backend in Azure, like everything offered in Azure, it does not require licensing, totally Azure consumption based. Customers pay on Application Insights consumption. Application Insights price info can be found [here](https://azure.microsoft.com/en-us/pricing/details/application-insights/).

If you use [Continuous Export](https://azure.microsoft.com/en-us/documentation/articles/app-insights-export-telemetry/) feature to export Video Telemetry data for long term retention and/or analysis, data storage cost would be separate.

## How to Use this Doc?

If all you want is to add Video Telemetry to your Azure Media Player app, you can skip directly to the section “[How Do I Use it?](#_How_Do_I)”

# Requirements and Approach

A video telemetry/user analytics solution needs to meet the following high level requirements:

|  |  |
| --- | --- |
| Requirement | Details |
| Able to COLLECT ANY data a client can CAPTURE | 2 things:   1. CAPTURE by client device: client device needs to have the capability to capture the desired data; 2. COLLECT by such video telemetry solution: client app needs to collect the desired data as long as they can be captured. |
| Reach most major client platforms/languages  Dev community | 1. iOS 2. Android 3. Browsers (supports [Azure Media Player](http://amp.azure.net/libs/amp/latest/docs/), but NOT limited to it) 4. Windows 10 5. Server analytics (e.g. multi-DRM license server analytics): ASP.NET, Java, PHP, Python |
| Portal | Easy to use portal for data visualization |
| Tool for advanced purpose | Tool for ad hoc query for deeper BI analysis |
| Low latency | Either low latency or near-zero latency, depending on positioning:   1. High-end video telemetry, such as [Conviva](http://www.conviva.com/); 2. Mid-end video telemetry, such as [Google Analytics Premium](https://www.google.com/analytics/premium/); 3. Low-end video telemetry, such as [Google Analytics](https://www.google.com/analytics/). |
| Support, road map and developer community | * Enterprise grade support * Long term road map * Developer community |

Application Insights is an extensible analytics service that monitors a live application. It processes 2.5 billion events and messages daily, from half a million applications, including some Microsoft services like Outlook and OneDrive. Application Insights is currently in Public Preview.

Application Insights documentation can be found [here](https://azure.microsoft.com/en-us/documentation/services/application-insights/). In this document, we do not assume readers are experienced with Application Insights.

Application Insights technology suites provides the following with respect to the above requirements:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Currently Supported by App Insights? | AI Support Details | |
| Able to COLLECT any data a client can CAPTURE | Yes | AI can define and COLLECT any **custom event, custom metric**, with optional **custom attributes**, as long as client device can CAPTURE the data for collection. There is only limit on what can be captured and no limit on what can be collected by AI. | |
| Reach most major client platforms/languages | Yes | * [Javascript API](https://github.com/Microsoft/ApplicationInsights-JS/blob/master/API-reference.md) (and AMP) * [ASP.NET API](https://msdn.microsoft.com/library/mt420197.aspx) * [Java SDK API](http://dl.windowsazure.com/applicationinsights/javadoc/) * [SDK for Node.js](https://github.com/Microsoft/ApplicationInsights-node.js) * [SDK for Xamarin](https://github.com/Microsoft/ApplicationInsights-Xamarin) | * [PHP](https://github.com/Microsoft/ApplicationInsights-PHP) * [Python](https://github.com/Microsoft/ApplicationInsights-Python) * [Ruby](https://github.com/Microsoft/ApplicationInsights-Ruby) * [SDK for iOS](https://github.com/Microsoft/ApplicationInsights-iOS) * [SDK for OS X](https://github.com/Microsoft/ApplicationInsights-OSX) * [SDK for Android](https://github.com/Microsoft/ApplicationInsights-Android) |
| Portals for data visualization | Yes | * [Azure Portal](https://ms.portal.azure.com/) * [Power BI](https://powerbi.microsoft.com/) via [Application Insights Content Pack for Power BI](https://azure.microsoft.com/en-us/blog/application-insights-content-pack-for-power-bi/) | |
| Ad hoc data exploration tool | Yes | Please contact the Microsoft account team for info on this tool. | |
| Low latency | 3 min latency today | Proposed to App Insights Team:   1. Current 3+ min latency as the default offering; 2. Near-real time latency as “Premium” offering at a much higher price using more expensive infrastructure. The concern is customers are not willing to pay. | |
| Support, road map and developer community | Yes | * Supported by Azure Support; * Backed by Visual Studio Team in C+E; * [Community contribution](https://github.com/Microsoft/ApplicationInsights-Home/blob/master/CONTRIBUTING.md); * The source code for SDKs on all platforms is [Open sourced](https://github.com/Microsoft/ApplicationInsights-Home); | |

A video telemetry solution must have the following components:

1. SDKs for all major client platforms such as iOS, Android, Windows and browsers;
2. Infrastructure: Data collection, processing, storage;
3. UI: Data reporting such as portal and BI tool;
4. Advanced query tools for ad hoc/advanced data analysis.
5. The road map and support plan for each component.

Instead of re-inventing wheels, our Video Telemetry solution leverages the following things from Application Insights suite:

* AI SDKs reaching most client platforms;
* [HockeyApp](http://hockeyapp.net/features/) acquisition and its future impact to SDKs on various mobile platforms;
* AI infrastructure in Azure: data collection, processing, storage;
* AI integration with Azure portal;
* [AI Content Pack for Power BI](https://azure.microsoft.com/en-us/blog/application-insights-content-pack-for-power-bi/) and its future improvements;
* Advanced ad hoc query tool;
* AI tight integration with Visual Studio 2015;
* AI developer community;
* AI [Continuous Export](https://azure.microsoft.com/en-us/documentation/articles/app-insights-export-telemetry/) of data to Azure blob storage.

In other words, our Video Telemetry is built on AI (Application Insights) + BI (Power BI with AI Content Pack) + KI (Kusto Integration).

With this solution, we should be able to answer all of the following questions (10 W’s):

1. **Who** is the user?
2. **When** did the user watch?
3. **Where** is the user located?
4. **Which** video did the user watch?
5. **Which** quality levels did the user watch (either auto-adapt or manual switch)?
6. **Which** audio stream/language did the user tune in?
7. **Which** closed caption track/language did the user select?
8. **Which** tech did the player use?
9. **Which** device type/OS version did the user use?
10. **How** long did the user watch for each video?

# How Do I Use it?

In this section, we will focus on implementing such a video telemetry solution for a video player based on [Azure Media Player API](http://amp.azure.net/libs/amp/latest/docs/). You can find Azure Media API document and sample code from this [link](http://amp.azure.net/libs/amp/latest/docs/).

Keep in mind that this approach works for video players on [most major platforms](https://github.com/Microsoft/ApplicationInsights-Home). The development languages are different but the approach stays the same.

The screenshots are for the actual Video Telemetry data for the [Azure Media Test Player](http://aka.ms/amtest) collected using this solution. Azure Media Test Player is a comprehensive test tool developed by the author to be used for building solutions on Azure Media Services. [This page](http://openidconnectweb.azurewebsites.net/) shows the technologies this test tool is built on.

## Prerequisites

For the ease of the flow, we assume you already have the following ready:

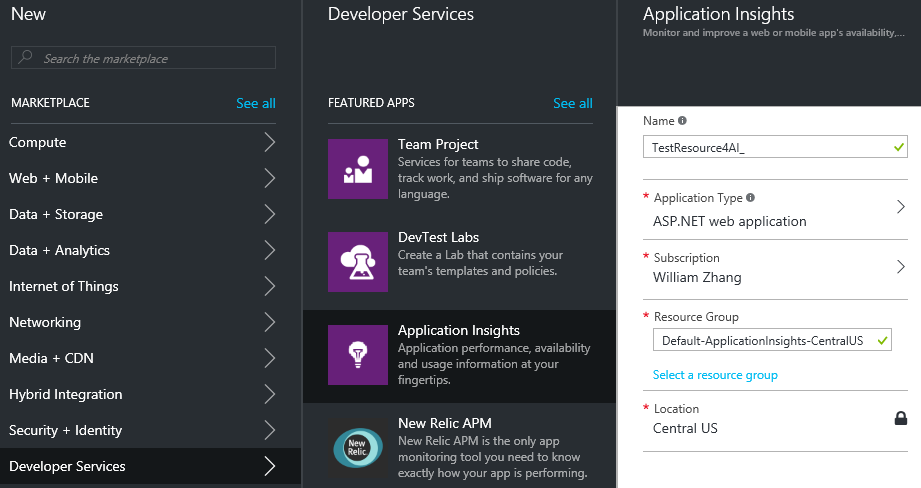
1. An Azure subscription;
2. Visual Studio 2015;
3. An ASP.NET MVC project (or any web project) containing the video player.
   1. The video player is based on Azure Media Player API.
   2. To be practical, it would be better your ASP.NET has user authentication included. If your player application does not need or have user authentication, everything still works but you will not be able to collect information on authenticated user or track each user’s activity.
   3. You have a couple of test videos you can play in the video player, regardless where the videos are hosted or whether the video is DRM protected.

## Preparation in Azure

Application Insights enabled client applications send data to Azure for storage, processing and reporting. Therefore, as the first step, we need to create an Application Insights resource in [Preview Azure Portal](https://ms.portal.azure.com/).

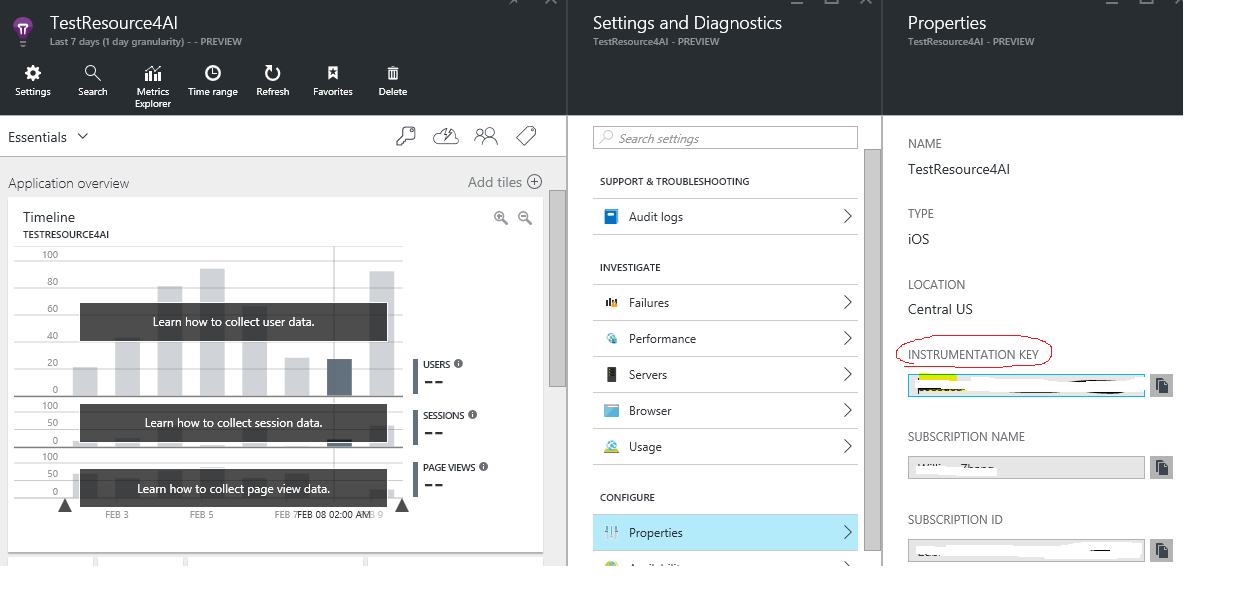
If your Azure Media Player application is an ASP.NET MVC application deployed as an Azure Web App, it may already have its own Application Insights resource. You can just use this resource and skip the following step for creating a new Application Insights resource.

To create a new Application Insights resource, in Azure Preview Portal, New -> Developer Services -> Application Insights



After an Application Insights resource is created. We need to get its Instrumentation Key which will be needed in the video player application.

Browse -> Application Insights, then select the Application Insights resource you have just created. Click on Settings icon on the top and select Properties on Settings and Diagnostics tab. Take a note of the instrumentation key for later use.



## Add Application Insights to Player App

Next we need to enable Application Insights for your ASP.NET project. Follow steps listed [here](https://azure.microsoft.com/en-us/documentation/articles/app-insights-asp-net/) to add Application Insights into your ASP.NET project hosting the Azure Media Player code.

## Custom Code for Video Telemetry

**STEP 1: Download** [**this javascript file for video telemetry**](http://openidconnectweb.azurewebsites.net/Content/appInsights.js) and save it to the /Content folder in your project.

CAUTION: Please make sure to replace the instrumentation key in the configuration section of the javascript file with your own instrumentation key. Otherwise you will not be able to see your data.

Let’s reference the javascript file by adding the following line at the header portion of the video player page. If you use layout file for your ASP.NET app, you could choose to put it in the header portion of \_Layout.cshtml file so that performance, availability and exception metric can all be collected as a side benefit for every page.

@Scripts.Render("~/Content/appInsights.js")

**STEP 2: Capture Authenticated User Context:**

If your ASP.NET application supports user authentication, you can add the following code right after the above line so that authenticated user context can be captured and correlated to video telemetry.

<script>

AppInsightsUtils.setAuthNUserContext(@System.Web.HttpContext.Current.Request.IsAuthenticated.ToString().ToLower(), "@System.Web.HttpContext.Current.User.Identity.Name");

</script>

**STEP 3: Track Video Player Events:**

One of the most popular requirements of video telemetry is to collect statistics of user actions/player events such as play, pause, seek, bitrate adaptation/change, go full screen, change volume, playback finish, etc. Fortunately, Azure Media Player exposes 19 different events including all of these.

After amPlayer object is initialized, you can register a desired set or all of the events by calling the following method (you may comment out those events you don’t care).

AMPUtils.registerEvents(amPlayer, ampEventHandler);

where the function and ampEventHandler are defined below:

AMPUtils.registerEvents = function (amPlayer, ampEventHandler) {

var events = [amp.eventName.loadstart,

amp.eventName.durationchange,

amp.eventName.loadeddata,

amp.eventName.loadedmetadata,

amp.eventName.canplaythrough,

amp.eventName.waiting,

amp.eventName.play,

amp.eventName.playing,

amp.eventName.ended,

amp.eventName.seeking,

amp.eventName.seeked,

amp.eventName.pause,

amp.eventName.fullscreenchange,

amp.eventName.volumechange,

amp.eventName.error,

amp.eventName.timeupdate,

amp.eventName.playbackbitratechanged,

amp.eventName.downloadbitratechanged

];

for (var i = 0; i < events.length; i++) {

amPlayer.addEventListener(events[i], ampEventHandler);

}

}

function ampEventHandler(evt) {

//any event handling code here, if any

AppInsightsUtils.ampEventHandler4AppInsights(evt, amPlayer);

}

The last line in ampEventHandler function calls the custom javascript code in appInsights.js to track different kind of events with custom properties.

That is all you need to do to add video telemetry into your Azure Media Player web application. After some test runs, you can look into your telemetry data in any of the following ways. The latency is usually around 3 minutes. The latency of data arriving into Kusto is a bit longer.

# Accessing Telemetry Data

Now you have added video telemetry capability into your player application, next we will look into three ways to access/visualize your telemetry data.

1. Using Azure Portal;
2. Using Power BI;
3. Using Kusto Client, which provides ad hoc/advanced data query via a query language.

Of course, it is your choice. If you don’t have a need for ad hoc query or deeper data insights, there is no need to learn KQL or to use Kusto client.

## Viewing Telemetry Data in Azure Portal

Log into [Azure Preview portal](https://ms.portal.azure.com/). Browse -> Application Insights, then select the Application Insights resource you set up and used in previous section. You will see the pane containing general info of your resource and the Setting pane also appears on the right.

* You can specify the time range of your interest via Time Range icon at the top;
* You can go into Metric Explorer by clicking on Metric Explorer icon at the top.
* You can invoke Settings and Diagnostics pane any time by clicking on Settings icon at the top;
* In Settings and Diagnostics pane, you can click on Usage button to go to Usage data;

Please keep in mind of the following:

1. You can add a chart;
2. After adding a chart, you can customize it by selecting those attributes you are interested in.

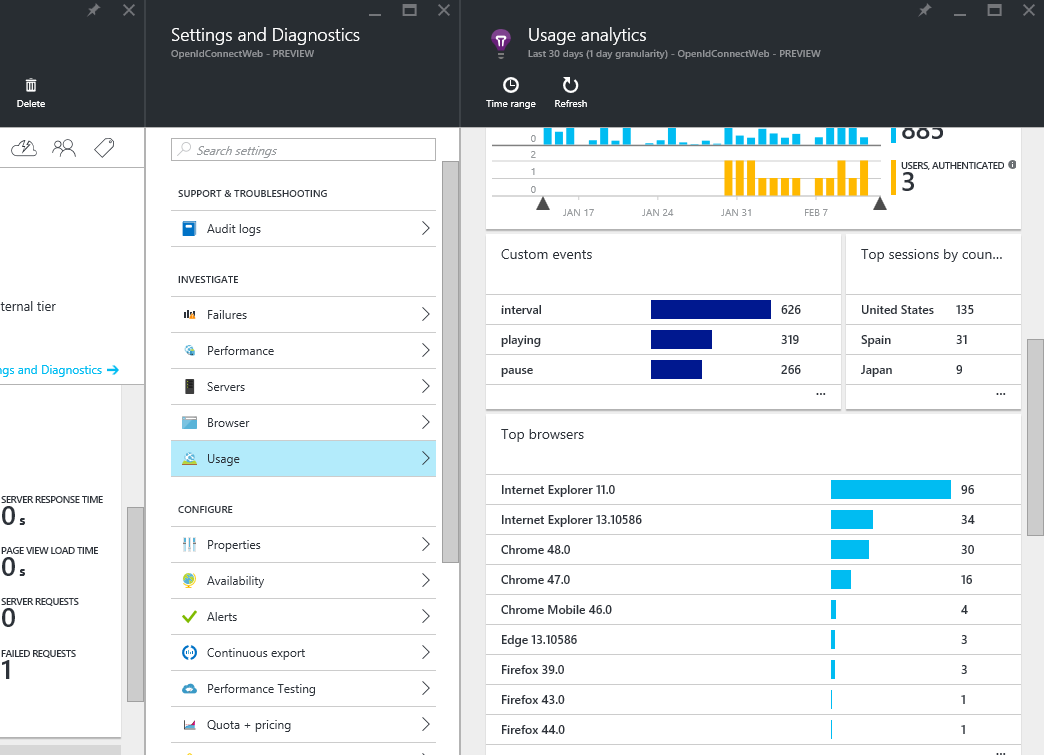
As part of video telemetry, we collect data related to the following events:

1. Playing;
2. Pause;
3. Go full screen;
4. Change audio volume;
5. Seek;
6. Switch bitrate (either auto-adaptation or manual change);
7. Interval (a timer event firing periodically at an interval you set, only during video playback).

The following table shows each event with corresponding custom properties and custom metrics collected together with the event:

|  |  |  |
| --- | --- | --- |
| Event | Custom Property | Custom Metric |
| Play | Video URL  Tech | Video duration |
| Pause |  | Absolute time |
| Full screen change |  |  |
| Volume change |  |  |
| Seek |  |  |
| Playbackbitratechanged\_[bitrate] |  | Current bitrate |
| Ended |  |  |
| Interval | URL  Tech  Audio stream name  Caption track language | Current bitrate  Time watched (seconds) |

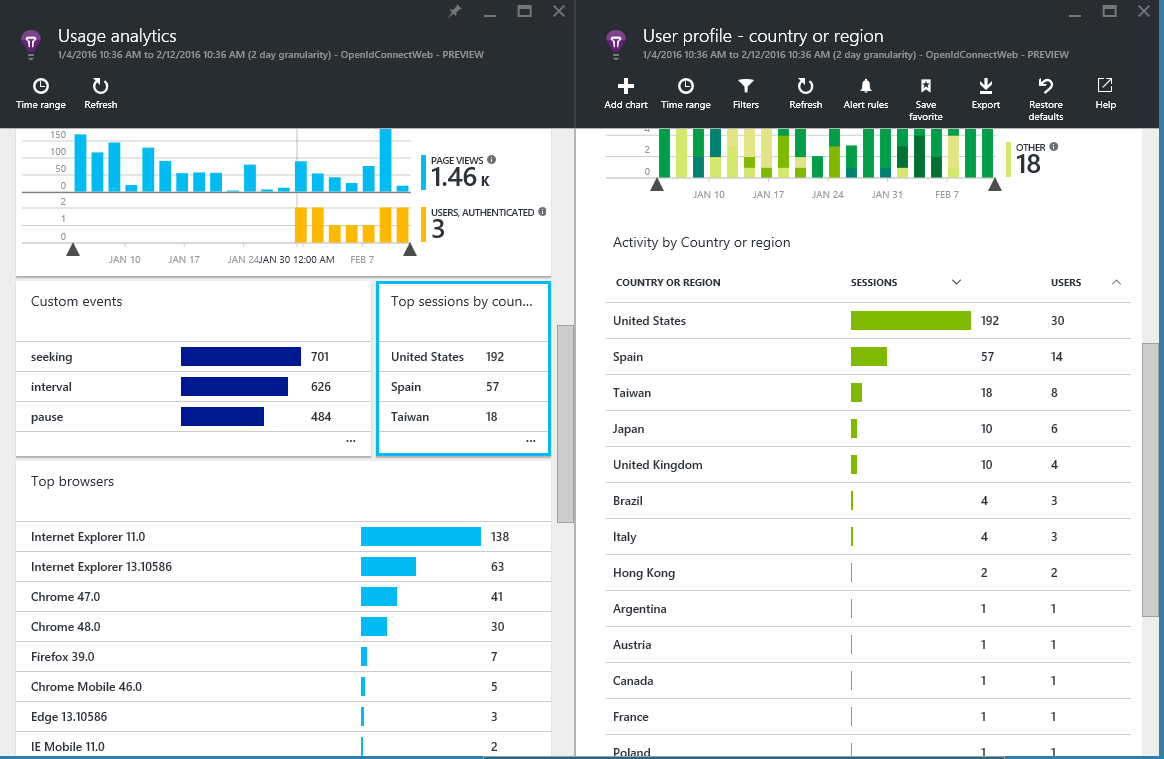
Let’s first look at usage data. Click on Settings icon, then Usage button, the Usage Analytics pane shows up:



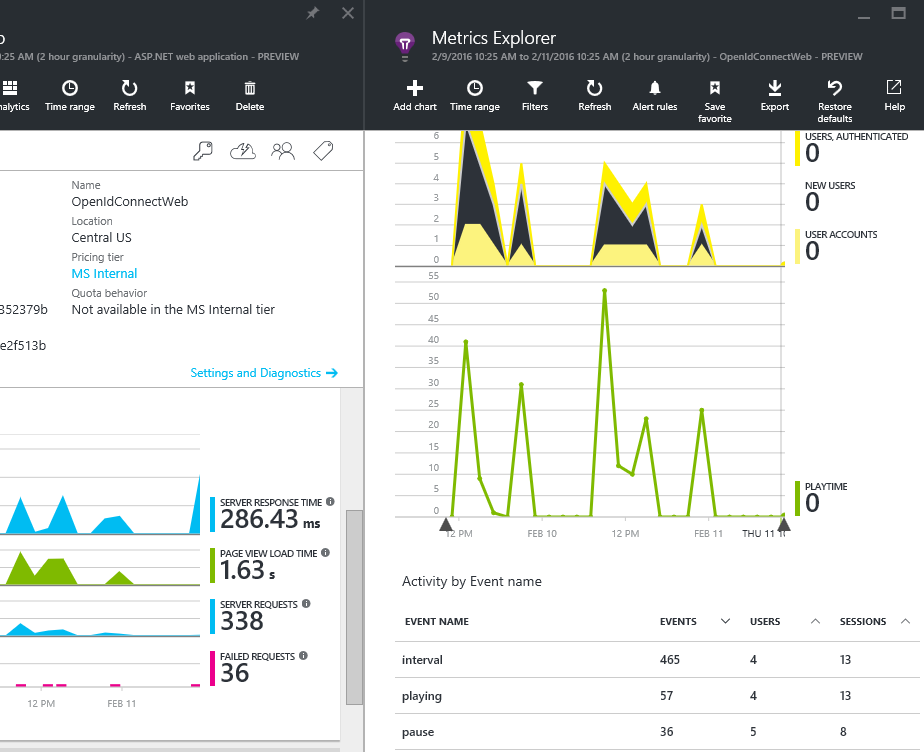
It contains a few charts.

* If you click on the top area of each chart, a details pane shows up with a full display and other details for that chart;
* If you click on a row in a chart, a details pane shows up with a full display and other details about that specific row of the chart.

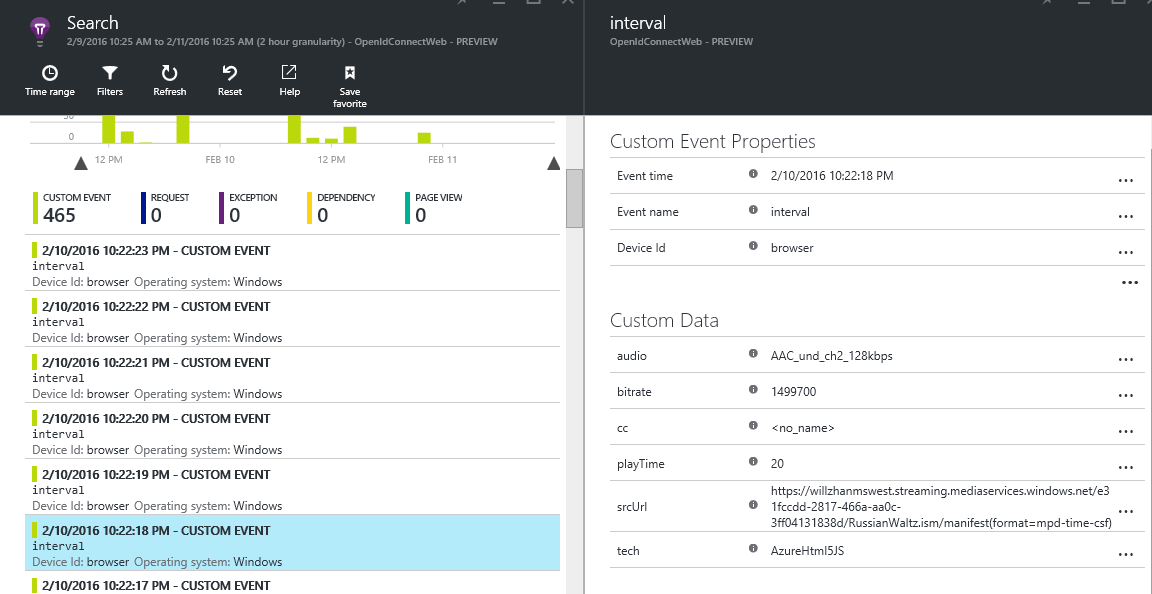
For example, if you click on the top portion of “Top sessions by country” chart, it will display details of user/session versus country:



Next, let’s go into Metric Explorer by clicking on the Metric Explorer icon at the top of the main pane for the Application Insights resource.



If you click on interval row, the Search pane appears. If you click on each event, a detail pane shows up:



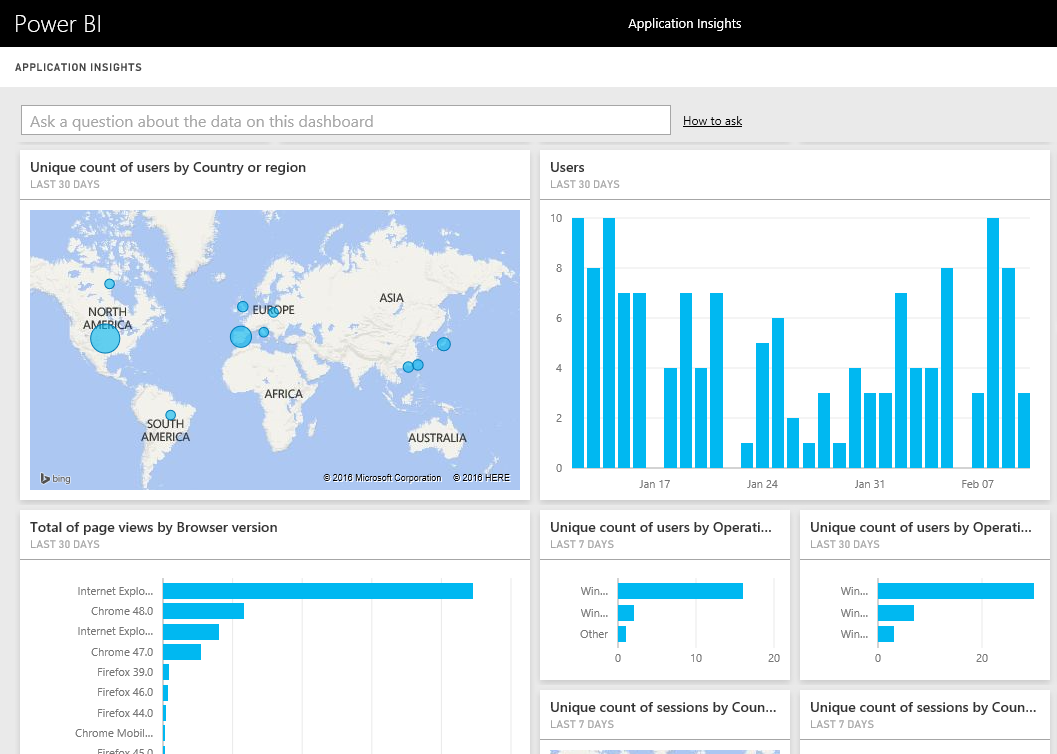
The “interval” details pane shows the custom properties and custom measurements. These data are collected by video telemetry at fixed time interval during video playback so that a full picture of user video playback can be captured. Of course, if this kind of details is not of interest to you, you can disable interval event.

## Viewing Telemetry Data in Power BI

Application Insights team created an Application Insights Content Pack for Power BI. With this content pack, one can view video telemetry data in Power BI.

[This document](https://powerbi.microsoft.com/en-us/documentation/powerbi-content-pack-application-insights/) describes the steps to set up Application Insights Content Pack for Power BI.

After connecting to Application Insights resource, Power BI will now import your data, and in about another minute, you’ll see a dashboard like below for your telemetry data.



## Advanced Data Query

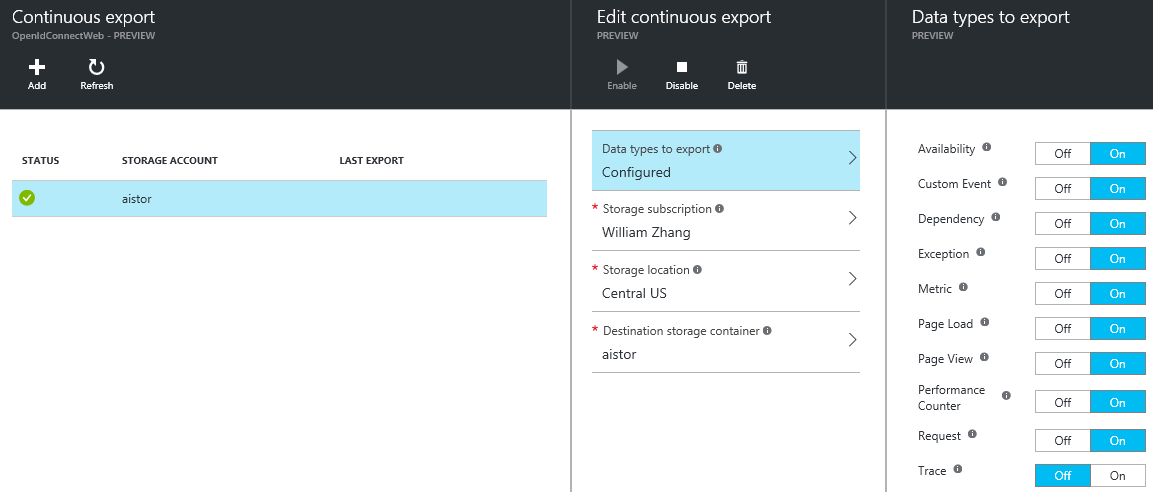
What we can see in Azure portal and Power BI via Application Insights Content Pack are high level statistics, similar to “canned” queries in relational database world. Even though the data is there, the UI does not expose all. Sometimes, we woud like to do deeper and more detailed inspection of telemetry data to understand more. For this, there is an ad hoc query tool which is not yet available externally. Please reach out to the Microsoft account team to see how this tool can be leveraged.

## Continuous Export of Telemetry Data

If you want to export Video Telemetry data for long term retention or custom analysis, you can perform Continuous Export on Application Insights data. The events you see in the Application Insights portal can be exported to storage in Microsoft Azure in JSON format. Continuous Export is available in the free trial period and on the [Standard and Premium pricing plans](https://azure.microsoft.com/pricing/details/application-insights/).

For detailed steps of setting up Continuous Export of your Application Insights, please see [here](https://azure.microsoft.com/en-us/documentation/articles/app-insights-export-telemetry/).

Continuous Export will export telemetry data into Azure blob storage as JSON files.



## Azure Stream Analytics

Azure Stream Analytics (ASA) makes it easy to set up real-time analytic computations on data streaming from devices, sensors, web sites, applications and infrastructure systems. It supports a powerful high-level SQL-like language that dramatically simplifies the logic needed to visualize, alert, or act on incoming events in near real-time.

You could use Stream Analytics to pipe your data to SQL DB, Table Storage, Event Hub or Power BI. This opens up many possibilities of how you process, analyze and view your data. For example, the following two document explain how to set up Stream Analytics to pipe data to Power BI:

* [Use Stream Analytics to feed Power BI from Application Insights](https://azure.microsoft.com/en-us/documentation/articles/app-insights-export-power-bi/);
* [Outputting Real-Time Stream Analytics data to a Power BI Dashboard](https://powerbi.microsoft.com/en-us/blog/outputting-real-time-stream-analytics-data-to-a-power-bi-dashboard/)

# Summary

We have presented an all-Azure solution for Video Telemetry/user analytics based on AI + BI + KI.

We provided detailed steps and source code for such a solution for Azure Media Player client.

We also presented three ways to look at Video Telemetry Data.

Have fun with Video Telemetry!

Your feedback is very much appreciated. Please direct your questions/feedback to me at [willzhan@microsoft.com](mailto:willzhan@microsoft.com).

**ACKNOWLEDGMENT:** The encouragement and support of Amit Rajput and Vishal Sood of Azure Media Services Team is greatly appreciated.