

SimplePLM Sample

*This sample is compatible with the Xbox One XDK (April 2016).*

# Description

This sample shows the behavior of the PLM events and the events that are related to PLM. The sample will print to the screen and to debug output a timestamp, thread ID, function name, and any additional data relevant to that function for events related to PLM. This sample can be used to understand the behavior of PLM events.

The sample can also perform operations that cause PLM-related transitions to demonstrate what events and states are affected. These include launching into a fullscreen SystemOS experience (Settings), Showing the AccountPicker TCUI, and performing a RestartApplicationOnly.

The user can also toggle whether or not the sample uses a suspend deferral using the ‘B’ button. This will not change any PLM behavior but will demonstrate how the suspend can be completed off of the core window thread.

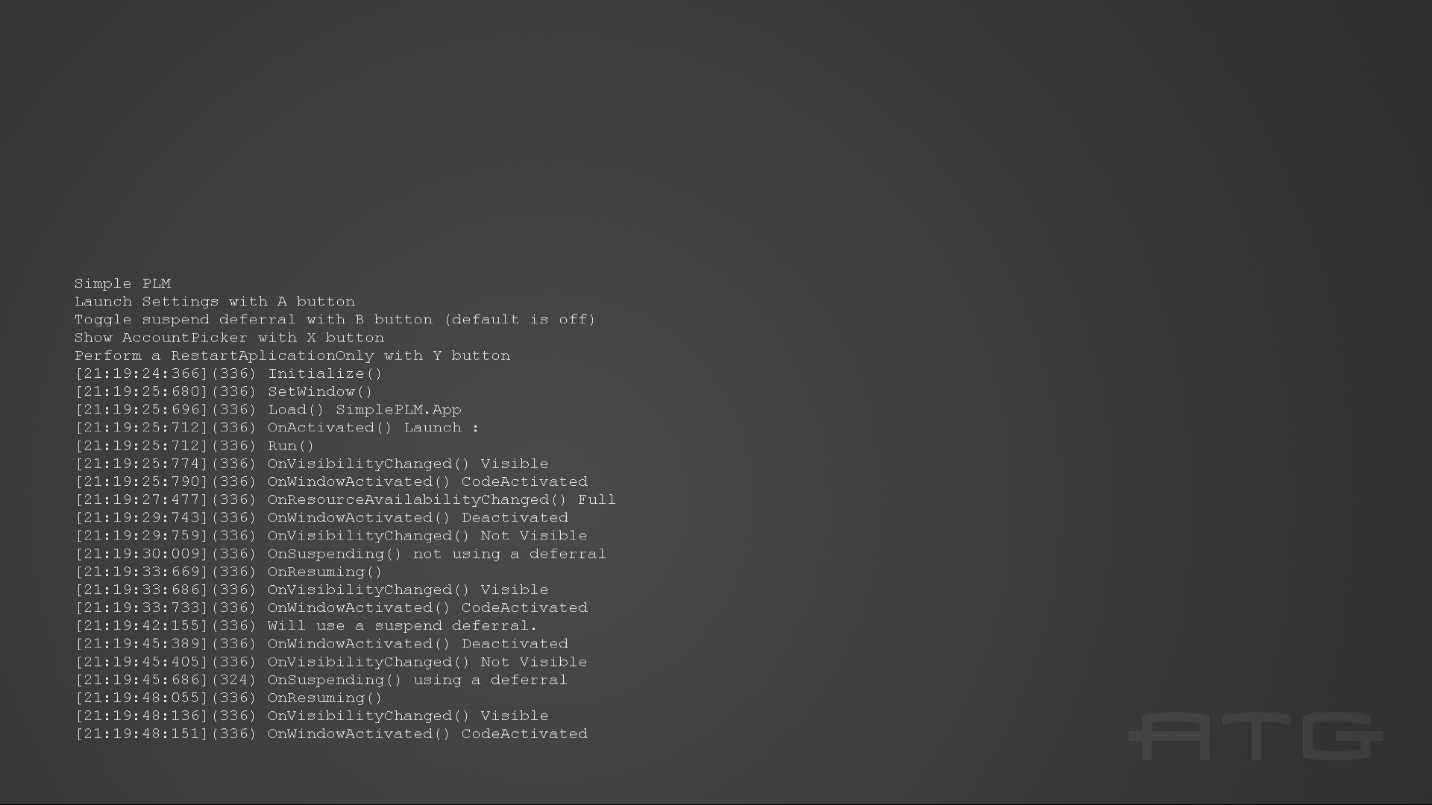
If you are looking for an example of how to manage saving data using connected storage and how that relates to the PLM events for suspending and resuming, the GameSave sample may be more appropriate as this sample does not use connected storage.

# Building the Sample

There are no additional steps necessary for building this sample. Simply open the solution in Visual Studio and then compile and deploy it to an Xbox One devkit.

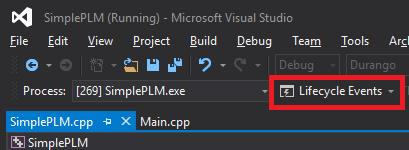
# Using the sample

## Main Screen



|  |  |
| --- | --- |
| Action | Xbox One Controller |
| Launch the Settings App | A |
| Toggle suspend deferral | B |
| Show the AccountPicker | X |
| Perform a RestartApplicationOnly | Y |
| Exit | View |

When running in Visual Studio to test suspend and resume operations, the user can use the Lifecycle Events menu to cause the app to suspend and resume shown below.



When not running under the Visual Studio debugger the user can cause an application to suspend by making the app not visible (launching settings will do this), the app will suspend after 10 minutes. To resume a suspended application, the user simply needs to make the application visible again. Alternately, the user can use the Xbox One Manager to perform Suspend and Resume operations.

Another option is to use the Xbapp.exe tool that is installed with the XDK Tools which will allow you to suspend and resume an app with the following commands:

Xbapp.exe suspend SimplePLM\_1.0.0.0\_x64\_\_zjr0dfhgjwvde

Xbapp.exe resume SimplePLM\_1.0.0.0\_x64\_\_zjr0dfhgjwvde

# Implementation notes

This sample currently logs events for the following functions:

IFrameworkView::Initialize

IFrameworkView::SetWindow

IFrameworkView::Load

CoreApplicationView::Activated

IFrameworkView::Run

CoreApplication::Suspending

CoreApplication::Resuming

CoreApplication::ResourceAvailabilityChanged

CoreWindow::VisibilityChanged

CoreWindow::Activated

Logging of these events occurs in Main.cpp in the associated functions and event handlers, not in SimplePLM.cpp. There are also logs for when different operations are chosen and an initial log to remind the user of the controls.

To enable the ResourceAvailabilityChanged event to fire for FullWithExtendedSystemReserve the <mx:GpuAvailability>variable</mx:GpuAvailability> tag is required in the Package.appxmanifest.

Due to the nature of RestartApplicationOnly it will not function properly if the app is being debugged by Visual Studio. If RestartApplicationOnly is performed while the Visual Studio debugger is attached the app will simply crash. To observe the effect of RestartApplicationOnly please run the sample without the Visual Studio debugger attached.

# Update history

Initial release May 2016