



# Media eXperience Analyzer

Getting Started Guide



Intro to the Media eXperience Analyzer .....	3
Overview .....	3
Target Audience & Pre-requisites .....	3
System Requirements & Recommended Hardware .....	3
MXA Essentials .....	4
Collecting traces .....	4
Opening a trace in MXA .....	5
Understanding the MXA UI .....	6
Selection Trees .....	7
Datasets .....	8
Panels .....	8
Data Viewers .....	9
Zoom Levels .....	10
Toolbars .....	11
Control .....	11
Navigation .....	13
Mode .....	14
CPU .....	14
Filtering datasets .....	14
Searching datasets .....	16
Pivoting on datasets in panels .....	17
Pivoting on datasets in selection trees .....	19
Data viewers .....	20
Properties .....	20
Callstack .....	21
ReadyThread .....	22
Bitmap .....	23

Summary.....	24
Event Dumper .....	25
Datasets.....	26
GPU .....	26
DMA Operations .....	26
DX VSync ISR/DPC Cadence .....	27
Media.....	28
Video Glitches .....	28
Video Glitches Severity .....	29
Quick Reference .....	30
Keyboard Shortcuts .....	30
Navigation.....	32
Common Tasks.....	33
Frequently Asked Questions & Troubleshooting.....	34
Release Notes .....	34
Feedback.....	34

## INTRO TO THE MEDIA EXPERIENCE ANALYZER

### OVERVIEW

Media eXperience Analyzer (MXA) is a tool that engineers can use to analyze visual performance and then optimize the performance and quality of various media scenarios on Windows devices. MXA helps performance engineers infer meaningful information from a large amount discrete event data by representing the events visually and providing powerful filtering capabilities. MXA can be used to optimize quality, performance, and latency in the following scenarios:

- Audio and video playback and transcode
- Energy efficiency for media
- Real-time communications (Skype)
- Camera capture
- Casting

MXA supports loading trace files collected on Windows. MXA enables engineers to identify delays and bottlenecks in the CPU, GPU, network, memory, and disk. It is primarily used to:

- Find the root cause of audio and video glitches
- Identify opportunities to improve audio video synchronization
- Correlate a physical domain with a software domain
- Correlate a component's power usage with system activity
- Optimize latency between casting source and sink devices
- Optimize devices for full screen video playback and low power audio
- Verify critical media threads are enlisted in [Multimedia Class Scheduler Service](#) (MMCSS)

### TARGET AUDIENCE & PRE-REQUISITES

The primary audiences for this tool are IHVs, OEMs, ODMs, and ISVs that develop or ship media related software and hardware components on Windows devices. Prior experience with [Windows Internals](#), [Event Tracing for Windows](#) (ETW), [Windows Performance Toolkit](#) (WPT), and Multimedia applications is recommended.

### SYSTEM REQUIREMENTS & RECOMMENDED HARDWARE

	Minimum	Recommended
OS	Windows 8.1 Pro amd64 *	Windows 10 Pro AMD64

<b>CPU</b>	Dual core	Quad core or better
<b>Graphics</b>	Mid-range DX9 graphics card	High-end graphics card
<b>Memory</b>	2GB of RAM **	4GB of RAM or more ***
<b>Disk</b>	Any SSD or HDD that has 5GB or more of free space	SSD or fast HDD that has 10GB or more of free space
<b>Monitor</b>	Single monitor with 720p resolution	Dual monitor setup with monitors that support resolutions >= 1920 x 1080

\* MXA is not supported on Windows N SKUs.

\*\* When opening multiple trace files, that include one ETW log file and two videos, the minimum memory is 4GB of RAM.

\*\*\* The amount of free RAM should be 3 times the size of the trace. For example, when opening a 2GB trace, the amount of free RAM should be 6GB.

## MXA ESSENTIALS

This section describes how to collect, load, and start visualizing events in MXA.

### COLLECTING TRACES

MXA supports several trace formats including ETW, Windows software trace preprocessor (WPP), text power logs, and video files (mp4). The most commonly used trace format is ETW.

There are several ways to collect logs that can be analyzed in MXA:

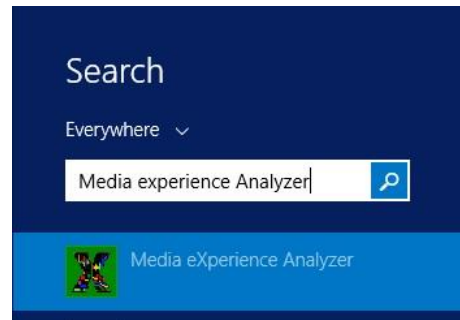
- Windows Performance Recorder (WPR)**  
 Install the Windows Performance Toolkit which ships inside the Windows Assessment and Deployment Kit (ADK) and use WPRUI.exe or WPR.exe to manually collect an ETW trace. This MXA installation includes a sample Windows Performance Recorder Profile (WPRP) file, Multimedia\_WPRP.wprp, that contains the essential kernel and user mode providers that are used for analyzing multimedia and power scenarios.
  - Click “Add profiles” in WPRUI to load this custom WPRP file, then collect a trace while a multimedia scenario is running.

- If the test device has 4GB or more of RAM, set the logging mode to Memory. If it has less than 4GB of RAM, set the logging mode to File.
- **Automated media tests in test kits**  
The GlitchFree media playback performance and quality assessment in the [Hardware Lab Kit](#) (HLK) has an option to generate verbose ETW logs on Windows 10. These logs can be used for analyzing audio and video glitches in MXA or WPA.
- **Author an MP4 video using a camcorder**  
Author an MP4 video of a scenario using a camera or camcorder. Cameras with a higher frame rate enables more precise correlation between the physical and software domains.
- **Generate a power trace using a data acquisition board**  
Instrumented systems and data acquisition boards can be used to generate power text logs that contain precise power measurements per component or hardware module.

## OPENING A TRACE IN MXA

### Option 1 – Windows Start screen

If you use WPR or WPRUI to capture an ETW log, MXA has access to the build and trace symbols path in the trace and will automatically resolve symbols. In these cases, MXA can be launched from the Windows Start screen without setting the symbol path at the command prompt. Manually setting the `_NT_SYMBOL_PATH` is required when analyzing a trace that contains code paths that do not have symbols on the public symbol server.



For details on how to set the symbol path, refer to **Option 2** below.

### Option 2 – Command Line

MXA supports loading between 1 and 10 traces from different data sources in the same timeline or instance of MXA. The offset window can be used to align multiple traces.

1. Set the symbol path  
Prior to loading a trace, set the symbol path so that call stacks can be resolved:
  - a. Launch an elevated command prompt
  - b. `cd /d "C:\Program Files\Media eXperience Analyzer"`
  - c. `set`  
`_NT_SYMBOL_PATH=C:\symbols;c:\symcache;srv*\symbols\symbols;http://msdl.microsoft.com/download/symbols`
2. Load the trace(s) using the following syntax:

# of traces	Sample Syntax	Comments
1	<code>xa.exe -i &lt;Path_To_ETW_Trace1.etl&gt;</code>	This is the most common usage. It opens MXA and loads the specified trace file.
2	<code>xa.exe -i &lt;Path_To_ETW_Trace1.etl&gt; -i1 &lt;Path_To_ETW_Trace2.etl&gt;</code>	Example application for loading two traces in the same timeline includes a client/server scenario.  MXA Supports loading 1 to 10 traces in the same timeline.
3	<code>xa.exe -i &lt;Path_To_ETW_Trace1.etl&gt; -i1 &lt;Path_To_Video1.etl&gt; -i2 &lt;Path_To_Video2.etl&gt; -p MIRACAST</code>	The -p argument is for MXA profile. The Miracast MXA profile contains a layout that is designed for conducting Miracast performance investigations. This profile is built-into the tool.

**Tip:** To reduce the amount of time it takes to load a trace in MXA:

- Click “Turn Symbols Off” on the MXA splash screen.
- To disable symbol lookup on load from the command line, pass in the “-q” argument.
  - Example: `xa.exe -i <Path_To_ETW_Trace1.etl> -q`
- Use a gigabit Ethernet connection instead of Wifi to connect to the network.
- Verify the \_NT\_SYMBOL\_PATH has a path to the local symbol cache (eg. `_NT_SYMBOL_PATH=C:\symbols;C:\symcache;`).

## UNDERSTANDING THE MXA UI

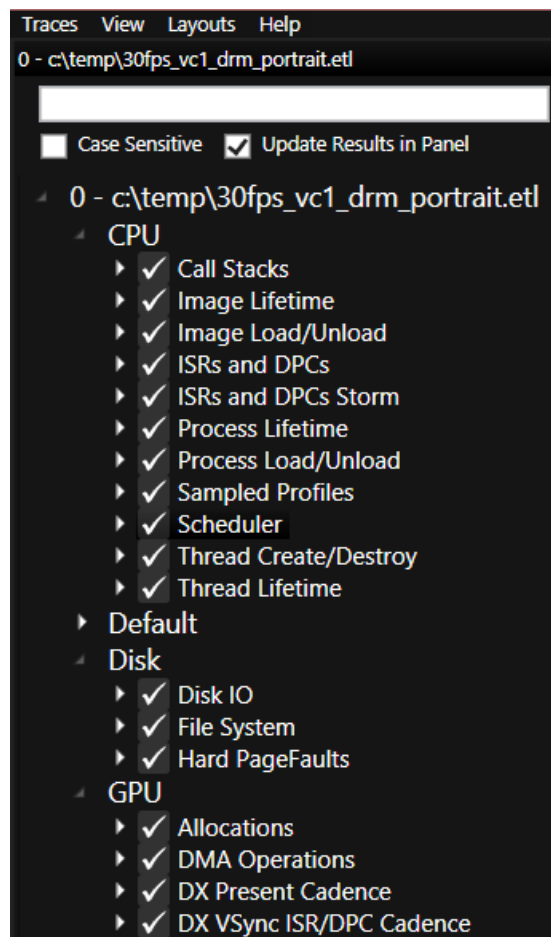
This section describes the 4 major components of the MXA’s UI:

1. [Selection Trees](#) are the entry point into a trace.
2. [Datasets](#) are logically grouped data accessible from selection trees.
3. [Panels](#) are windows that datasets can be visualized in.
4. [Data Viewers](#) provide additional details on the elements within a dataset.

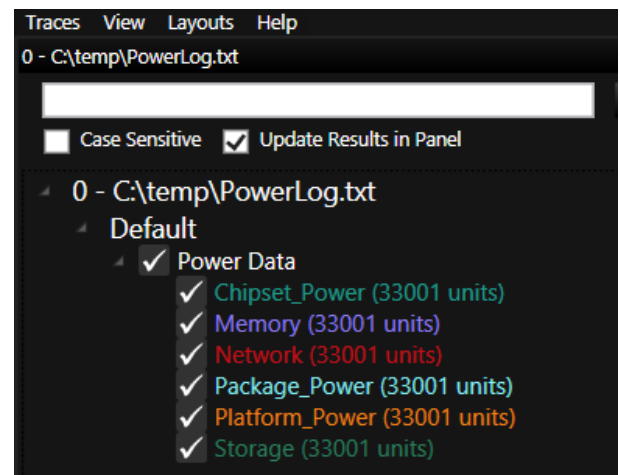




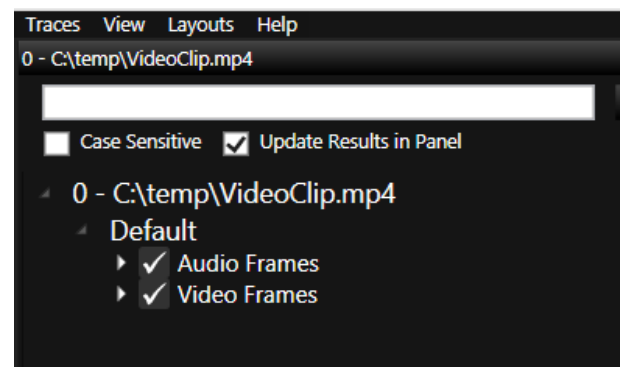
### Selection Tree: ETW Trace



### Selection Tree: Power Trace



### Selection Tree: Audio/Video MP4 file



## Datasets

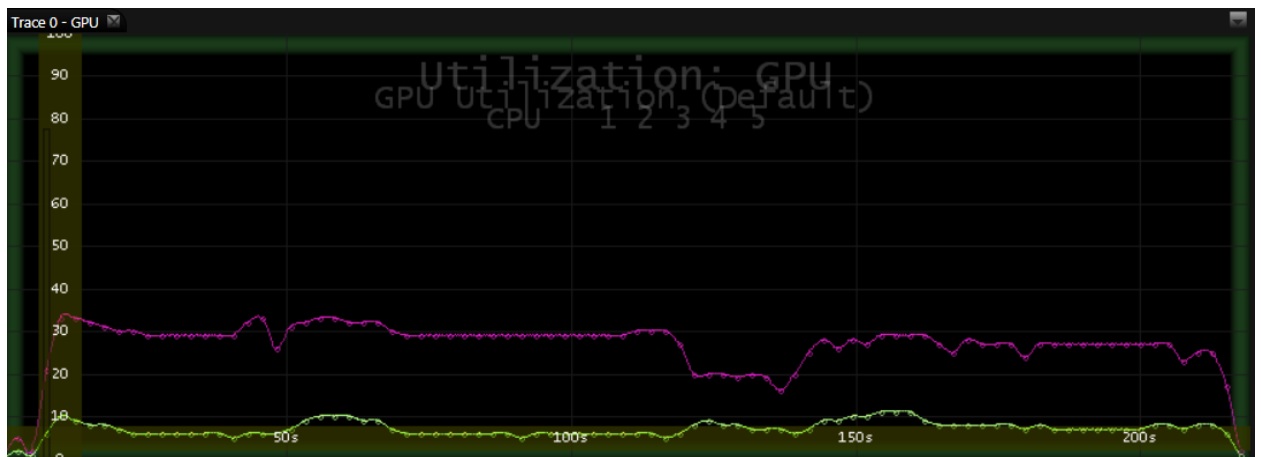
Selection trees contain the path to the trace file at the parent level. Immediate children of the selection tree are high-level labels that group datasets in logical groups. For example, in the ETW Trace above, all CPU related datasets, such as Scheduler and Sampled Profiles exist under the CPU node. Children of datasets are called data collections and children of data collections are referred to as data units. If a trace is missing datasets required for analyzing a performance or quality issue, verify the trace was collected with the full set of ETW providers required for conducting the analysis. MXA contains a sample WPR profile that contains the ETW providers necessary for analyzing media scenarios.

## Panels

Panels are the workspace where most of your interaction and analysis takes place. To visualize data within datasets, drag and drop datasets onto panels. To create new panels, press CTRL+N or go to the **View** menu and click **New Panel**. An empty panel looks like this:



A panel that had the Utilization GPU dataset dragged onto it looks like this:



## Data Viewers

Data viewers provide details or additional debug information around a single event or range of events. The most commonly used data viewers are Properties, CallStack, and ReadyThread. Here is a brief overview of the value each data viewer provides. The [Data Viewer](#) section covers each of these in more depth.

Data Viewer	Description
<a href="#">Properties</a>	Displays event payload information for user and kernel mode events.
<a href="#">CallStack</a>	Displays the call stacks from events in CPU Scheduler, CPU Sampled Profile, and instrumentation datasets.

<a href="#">ReadyThread</a>	Displays the call stacks for ReadyThreads in the CPU Scheduler dataset.
<a href="#">Event dumper</a>	Displays a list of all events near the current focus of the mouse.
<a href="#">Bitmap</a>	The Bitmap data viewer is used to visualize video frames in a media clip.
<a href="#">Summary</a>	Displays output from custom plugins. For example, the power plugin displays power stats by pressing ALT + dragging the mouse over a range of power stats.

## ZOOM LEVELS

There are several ways to zoom in and out within the timeline with the mouse and keyboard.

Action	Shortcut
Zoom in on a specific part of the timeline	Click and drag the mouse cursor over the portion of the timeline that you want to zoom in on.
Zoom out to 100% of the full timeline	Press Esc
Zoom in	<p>&lt;</p> <p>z</p> <p>Scroll wheel on mouse</p> <p>Click zoom in on the control toolbar</p>
Zoom out	<p>&gt;</p> <p>a</p> <p>Scroll button on mouse</p> <p>Click zoom out on the control toolbar</p>





Action	Shortcut
Auto zoom on the Y axis	`
Undo Zoom	Click undo zoom in on the control toolbar
Redo Zoom	Click redo zoom in on the control toolbar
Toggle between zooming in on full and partial part of Y axis	Y

## TOOLBARS

The toolbars can help you navigate and modify the MXA's UI. There are four toolbars accessible from the Toolbars menu:











- Controls
- Navigation
- Mode
- CPU

### Control





	<b>Toggle Vertical Line</b>	Display or hide the vertical line that follows the mouse movements across all panels. This can help you align events in multiple datasets.
	<b>Toggle Highlight</b>	Highlight or un-highlight an element in a panel. You can also right-click on an element in a panel and select Toggle Highlight.
	<b>Toggle Border</b>	Display or hide the grey border around each panel.
	<b>Toggle Grid</b>	Display or hide the grid in the background of each panel.

	<b>Toggle Top Ruler</b>	Display or hide the ruler at the top of a panel. This ruler shows the amount of time displayed at the current zoom level.
	<b>Toggle Bottom Ruler</b>	Display or hide the ruler at the bottom of a panel. This ruler displays the timestamps of the current location in the trace. (eg. wallclock time).
	<b>Toggle Left Ruler</b>	Display or hide the ruler on the left side of a panel. This ruler displays data that is unique to each dataset.
	<b>Toggle Right Ruler</b>	Display or hide the ruler on the right side of a panel. This ruler displays data that is unique to each dataset.
	<b>Toggle Text</b>	Display or hide the text displayed in the background of each panel that describes the name of the dataset presently in view in the panel.
	<b>Toggle Help</b>	Display or hide the help text that is overlaid on a panel.
	<b>Toggle Frame</b>	Display or hide the frame of the entire MXA window.
	<b>Toggle Red Highlight</b>	Enable or disable the red blinking highlighting feature that assists with visually identifying events selected in the selection tree.
	<b>Toggle Scalable Y Axis</b>	Switch between zooming in on part or all of the Y axis.

## Navigation

	<b>Zoom out 100%</b>	Zoom out to full trace. Keyboard shortcut: ESC
	<b>Zoom to Active</b>	Zoom in to region of the trace that has data.
	<b>Zoom in</b>	Zoom in.
	<b>Zoom out</b>	Zoom out.
	<b>Undo Zoom action</b>	Undo previous zoom actions.
	<b>Redo Zoom action</b>	Redo last zoom action.
	<b>Pan Up</b>	Pan up in a panel that supports panning up.
	<b>Pan Down</b>	Pan down in a panel that supports panning down.
	<b>Pan Left</b>	Pan left. Keyboard shortcut: Left arrow
	<b>Pan Right</b>	Pan right. Keyboard shortcut: Right arrow

## Mode

	<b>Toggle Color Mode Backward</b>	Go to the previous color mode
	<b>Toggle Color Mode Forward</b>	Go to the next color mode
	<b>Toggle Drawing Mode Backward</b>	Go to previous dataset pivot Keyboard shortcut: d Mouse shortcut: Right click and select pivot
	<b>Toggle Drawing Mode Forward</b>	Go to next dataset pivot Keyboard shortcut: f Mouse shortcut: Right click and select pivot

## CPU

Display or hide all activity in a panel that contains a CPU related dataset.



Each dataset contains thousands or hundreds of thousands of events. To identify patterns and the cause of performance issues you can filter out events that are not required for the analysis. One of the filtering options in MXA is offered in the selection tree. Next to each dataset and each dataset's child nodes are check boxes that provide three filtering modes: Show, highlight, and hide elements.

1. **Show elements** –All elements under the node are displayed. This is the default setting after you load a trace.



2. **Highlight all elements** – Click the checkbox next to a dataset once to highlight all elements under the node.

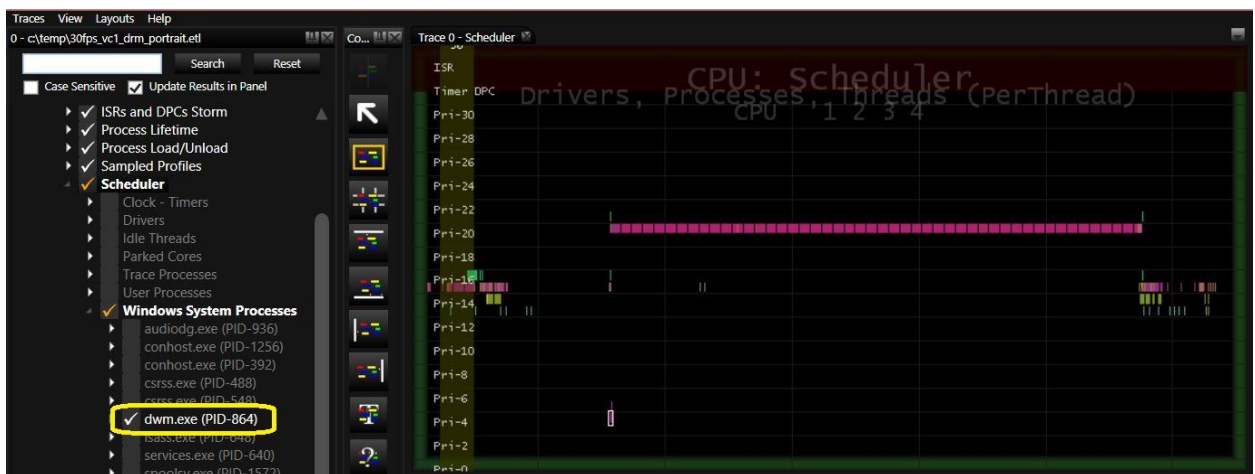


3. **Hide elements** – Click the checkbox next to a dataset again, to hide all elements under the node.





Here's an example of a dataset that is partially filtered. All processes and threads in the trace are hidden except for DWM.exe. To do this, hide all elements in the CPU Scheduler dataset, then select or show dwm.exe.



## SEARCHING DATASETS

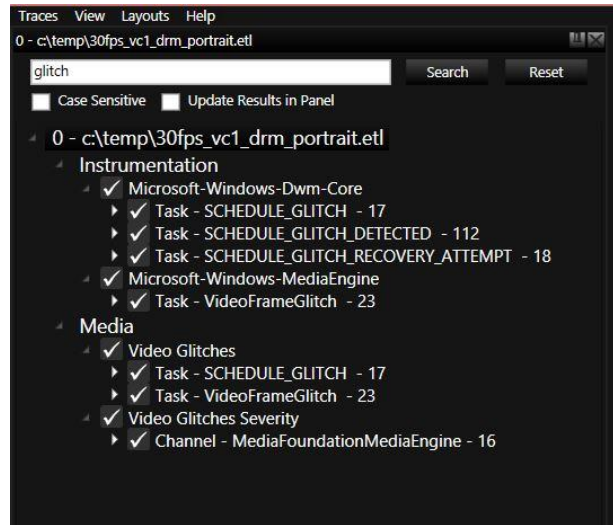
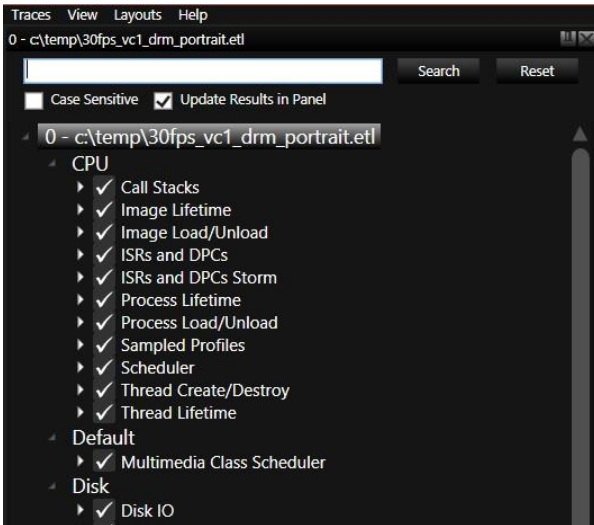
You can use the search text box in the selection tree to filter the list of datasets displayed in a selection tree. In the following example, the screenshot on the left displays the full list of all datasets generated by the MXA profile. The screenshot on the right is filtered based on the keyword search "glitch".

### Before Search

Full Selection Tree displayed after loading a trace.

### After Search

Selection tree after searching for keyword "glitch".



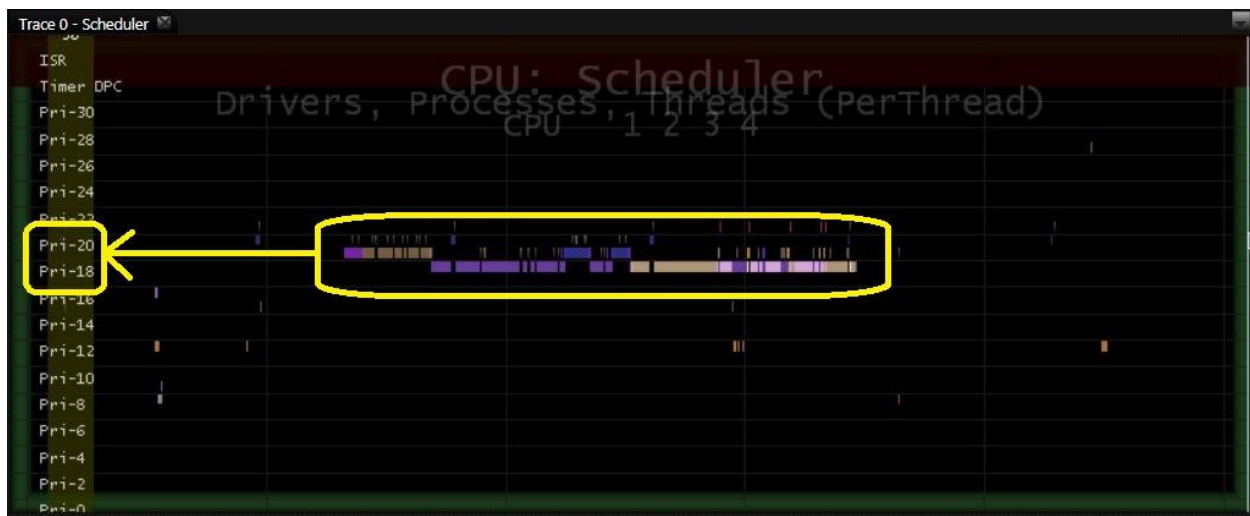
## PIVOTING ON DATASETS IN PANELS

Most datasets offer several pivots. Pivoting on datasets is done by right clicking on a panel and selecting a new pivot, by pressing “d”, or “f” on the keyboard.

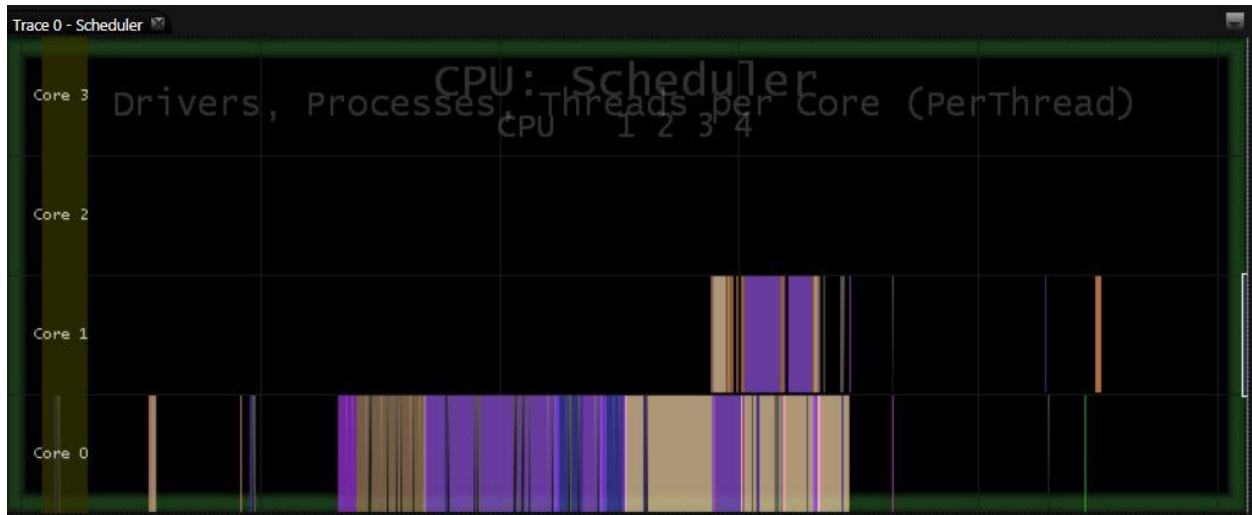
The following example demonstrates the value of pivoting on datasets:

Example: CPU Scheduler dataset

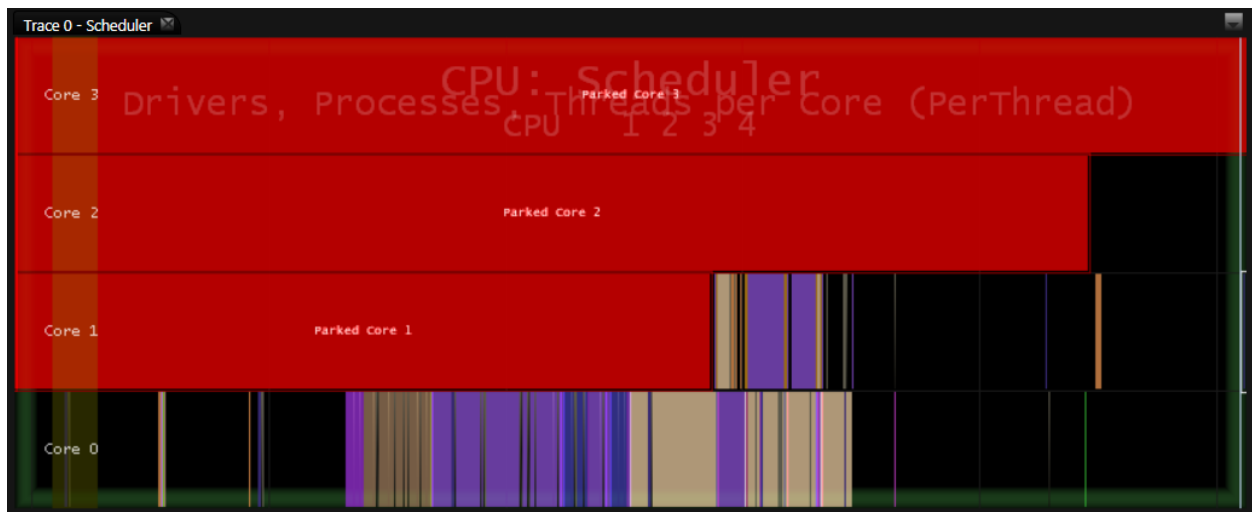
By default, this dataset displays thread priority in the Y axis. This view enables an engineer to quickly determine what priority each thread and driver is running at any given point in time. As shown in the following screenshot, most of the threads in this panel are running at priority 18 or 19.



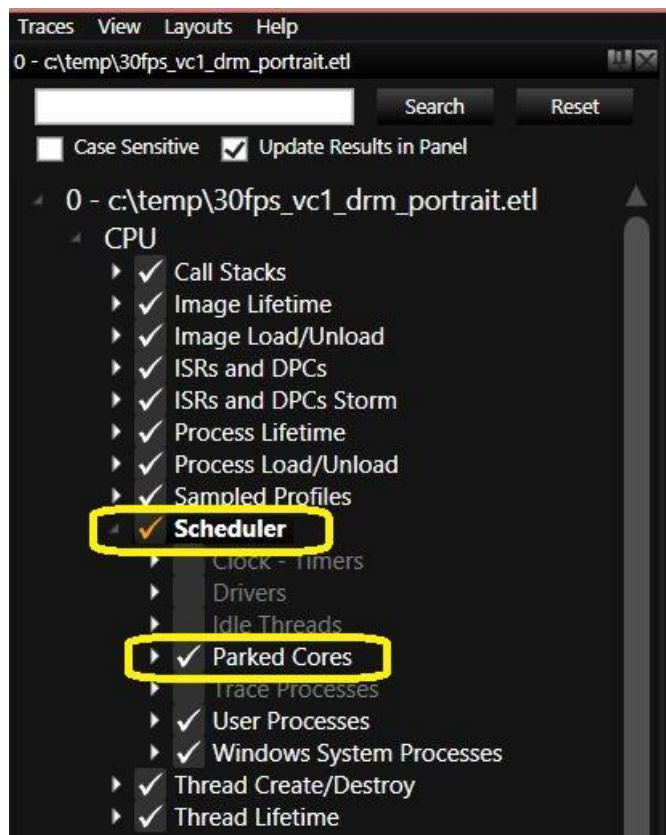
Changing the pivot to “Drivers, processes, and threads per core”, changes the Y axis from thread priority to thread activity per core. Each row in the Y axis is now labeled with a core.



Notice how there is no activity on core 2 & 3. If we bring back the parked cores node into the view by selecting them from the selection tree, we see the cores that were idle were parked. Parked cores are cores that have become inactive to save power.

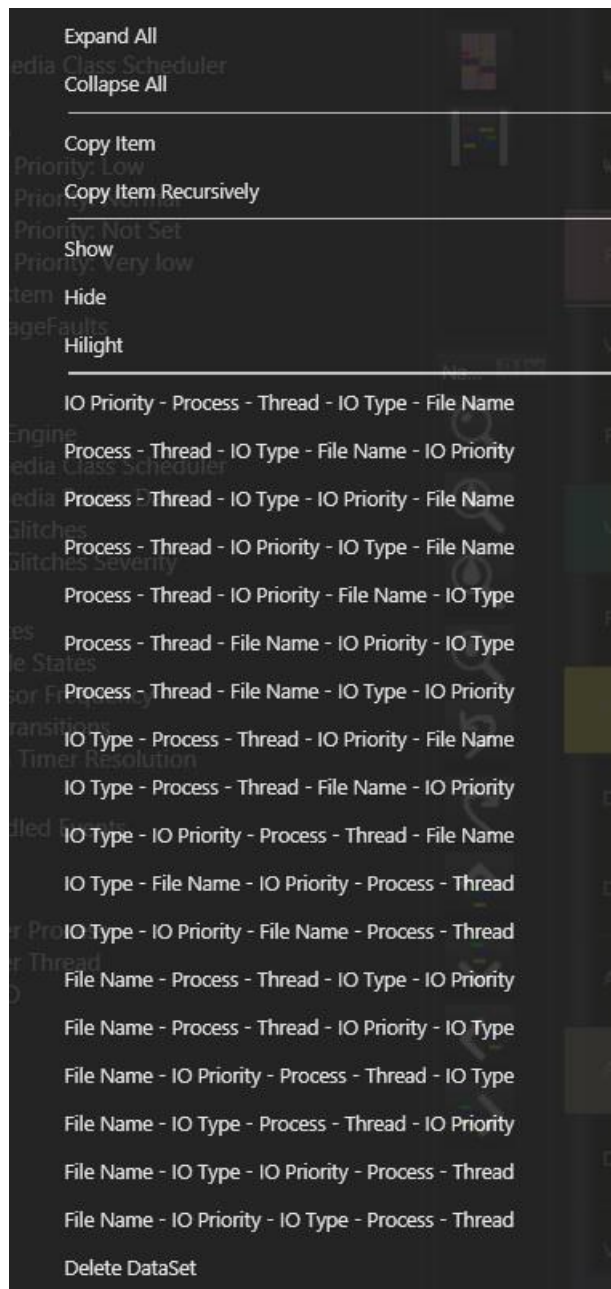


This screenshot shows how to visualize parked cores in the CPU Scheduler dataset.



## PIVOTING ON DATASETS IN SELECTION TREES

Datasets have a default order in which the child nodes are ordered. To change this hierarchy, right-click the dataset and select a different order or pivot. For example, in the Disk > Disk IO dataset, the default hierarchy is IO Priority > Process > Thread > IO Type > File Name. In some investigations where the process and thread of interest are known, it may be easier to pivot on these attributes. The following screenshot is an example of the types of pivots that are available for the Disk IO dataset.



## DATA VIEWERS

Data viewers provide details or additional debug information around a single event or range of events. The most commonly used data viewers are Properties, CallStack, and ReadyThread. Here is a brief overview of the value each data viewer provides.

### PROPERTIES

The properties data viewer is used to view the properties of an event payload. The event could be a CSWITCH event or a user mode provider event. The following screenshots are examples of user and kernel mode event payloads displayed in the properties data viewer.

Microsoft-Windows-MediaEngine User mode provider	CPU Scheduler Kernel mode CSWITCH event																																																																										
<div><div>Properties</div><table><tr><th>Property</th><th>Value</th></tr><tr><td>Unit 0 of 1</td><td></td></tr><tr><td>Task - VideoFrameGlitch</td><td>- 23</td></tr><tr><td>Event Id - 119 (Version 0) Opcode - win:Info</td><td>- 0</td></tr><tr><td>Process Id-</td><td>4804</td></tr><tr><td>Channel - MediaFoundationMediaEngine</td><td>- 16</td></tr><tr><td>Level - win:Verbose</td><td>- 5</td></tr><tr><td>Thread Id-</td><td>1908</td></tr><tr><td>Time(ns)</td><td>8,211,909,523</td></tr><tr><td>Event Local Time</td><td>10:18:13:134 04-04-2014</td></tr><tr><td>Event Time</td><td>17:18:13:134 04-04-2014</td></tr><tr><td>CPU</td><td>1</td></tr><tr><td>object</td><td>51436056</td></tr><tr><td>sampleTime</td><td>3330000</td></tr><tr><td>FramesLate</td><td>2</td></tr></table></div>	Property	Value	Unit 0 of 1		Task - VideoFrameGlitch	- 23	Event Id - 119 (Version 0) Opcode - win:Info	- 0	Process Id-	4804	Channel - MediaFoundationMediaEngine	- 16	Level - win:Verbose	- 5	Thread Id-	1908	Time(ns)	8,211,909,523	Event Local Time	10:18:13:134 04-04-2014	Event Time	17:18:13:134 04-04-2014	CPU	1	object	51436056	sampleTime	3330000	FramesLate	2	<div><div>Properties</div><table><tr><th>Property</th><th>Value</th></tr><tr><td>Unit 46 of 744</td><td></td></tr><tr><td>User Processes</td><td></td></tr><tr><td>mfmpm.exe (PID-4804)</td><td></td></tr><tr><td>Thread-4688 (mfcore.dll!0x53CA402B)</td><td></td></tr><tr><td>Start(ns)</td><td>8,222,265,998</td></tr><tr><td>Event Local Time</td><td>10:18:13:144 04-04-2014</td></tr><tr><td>Event Time</td><td>17:18:13:144 04-04-2014</td></tr><tr><td>CPU</td><td>0</td></tr><tr><td>Duration(ns)</td><td>22,272</td></tr><tr><td>Start Priority</td><td>21</td></tr><tr><td>End Priority</td><td>21</td></tr><tr><td>Reason</td><td>UserRequest - 6</td></tr><tr><td>Ideal Proc</td><td>2</td></tr><tr><td>TimeSinceLast</td><td>5,647,100</td></tr><tr><td>Time waiting Ready</td><td>17,664</td></tr><tr><td>Released on CPU:</td><td>1</td></tr><tr><td>Originated in:</td><td>Thread</td></tr><tr><td>Adjustment Increment:</td><td>1</td></tr><tr><td>Adjustment Reason:</td><td>Unwait - 1</td></tr><tr><td>Releasing PID</td><td>0</td></tr><tr><td>Releasing TID</td><td>4,700</td></tr></table></div>	Property	Value	Unit 46 of 744		User Processes		mfmpm.exe (PID-4804)		Thread-4688 (mfcore.dll!0x53CA402B)		Start(ns)	8,222,265,998	Event Local Time	10:18:13:144 04-04-2014	Event Time	17:18:13:144 04-04-2014	CPU	0	Duration(ns)	22,272	Start Priority	21	End Priority	21	Reason	UserRequest - 6	Ideal Proc	2	TimeSinceLast	5,647,100	Time waiting Ready	17,664	Released on CPU:	1	Originated in:	Thread	Adjustment Increment:	1	Adjustment Reason:	Unwait - 1	Releasing PID	0	Releasing TID	4,700
Property	Value																																																																										
Unit 0 of 1																																																																											
Task - VideoFrameGlitch	- 23																																																																										
Event Id - 119 (Version 0) Opcode - win:Info	- 0																																																																										
Process Id-	4804																																																																										
Channel - MediaFoundationMediaEngine	- 16																																																																										
Level - win:Verbose	- 5																																																																										
Thread Id-	1908																																																																										
Time(ns)	8,211,909,523																																																																										
Event Local Time	10:18:13:134 04-04-2014																																																																										
Event Time	17:18:13:134 04-04-2014																																																																										
CPU	1																																																																										
object	51436056																																																																										
sampleTime	3330000																																																																										
FramesLate	2																																																																										
Property	Value																																																																										
Unit 46 of 744																																																																											
User Processes																																																																											
mfmpm.exe (PID-4804)																																																																											
Thread-4688 (mfcore.dll!0x53CA402B)																																																																											
Start(ns)	8,222,265,998																																																																										
Event Local Time	10:18:13:144 04-04-2014																																																																										
Event Time	17:18:13:144 04-04-2014																																																																										
CPU	0																																																																										
Duration(ns)	22,272																																																																										
Start Priority	21																																																																										
End Priority	21																																																																										
Reason	UserRequest - 6																																																																										
Ideal Proc	2																																																																										
TimeSinceLast	5,647,100																																																																										
Time waiting Ready	17,664																																																																										
Released on CPU:	1																																																																										
Originated in:	Thread																																																																										
Adjustment Increment:	1																																																																										
Adjustment Reason:	Unwait - 1																																																																										
Releasing PID	0																																																																										
Releasing TID	4,700																																																																										

CALLSTACK

The CallStack data viewer displays the call stacks from events in CPU Scheduler, CPU Sampled Profile, and instrumentation datasets. In the following screenshot, this is the NewThreadStack from the CPU Scheduler CSWITCH event.

```
CallStack
CallStack
ntoskrnl.exe!SwapContext_PatchLdMxCsr
ntoskrnl.exe!KiSwapContext
ntoskrnl.exe!KiSwapThread
ntoskrnl.exe!KiCommitThreadWait
ntoskrnl.exe!KeWaitForMultipleObjects
ntoskrnl.exe!ObWaitForMultipleObjects
ntoskrnl.exe!NtWaitForMultipleObjects
ntoskrnl.exe!KiSystemServiceCopyEnd
ntoskrnl.exe!KiServiceLinkage
win32k.sys!CTokenManager::ProcessTokens
win32k.sys!CTokenManager::TokenThread
win32k.sys!NtTokenManagerThread
ntoskrnl.exe!KiSystemServiceCopyEnd
-----
dcomp.dll!ZwTokenManagerThread
dwmcore.dll!CLocalSurfaceManager::ProcessKernelTokens
dwmcore.dll!CLocalSurfaceManager::s_TokenThreadMain
kernel32.dll!BaseThreadInitThunk
ntdll.dll!RtlUserThreadStart
```

### CSWITCH / NewThreadStack

When you hover over elements in the CPU Scheduler dataset, the CallStack data viewer displays the call stack of the CSwitch event or NewThreadStack. The stack on the top is the kernel stack and the one on the bottom is user mode.

### Sampled Profile call stacks

When you hover over elements in the CPU Sampled Profiles dataset, the CallStack data viewer displays the call stack of a sampled profile event. With stackwalking enabled, the ETW sub-system captures the instruction pointer and the call stack of what was happening on the CPU every 1ms (default interval).

## READYTHREAD

The ReadyThread data viewer displays the call stacks of ReadyThreads in the CPU Scheduler dataset.



```
ReadyThread
CallStack
ntoskrnl.exe!KeSetEvent
dxgmms1.sys!DXGPRESENTHISTORYTOKENQUEUE::PropagatePresentHistoryToken
dxgmms1.sys!VidSchiCompleteFlipEntry
dxgmms1.sys!VidSchiProcessDpcVSyncCookie
dxgmms1.sys!VidSchDdiNotifyDpc
dxgkrnl.sys!DxgNotifyDpcCB

This part of the callstack  
has been removed

dxgkrnl.sys!DpiFdoDpcForIsr
ntoskrnl.exe!KiExecuteAllDpcs
ntoskrnl.exe!KiRetireDpcList
ntoskrnl.exe!KiRetireDpcList
ntoskrnl.exe!KiDispatchInterruptContinue
ntoskrnl.exe!KiDpcInterruptBypass
ntoskrnl.exe!KiInterruptDispatch
ntoskrnl.exe!RtlpLookupUserFunctionTableInverted
ntoskrnl.exe!RtlpLookupUserFunctionTable
ntoskrnl.exe!RtlpLookupFunctionEntryForStackWalks
ntoskrnl.exe!RtlpWalkFrameChain
ntoskrnl.exe!RtlWalkFrameChain
ntoskrnl.exe!EtwTraceStackWalk
ntoskrnl.exe!EtwStackWalkApc
ntoskrnl.exe!KiDeliverApc
ntoskrnl.exe!KiSwapThread
ntoskrnl.exe!KiCommitThreadWait
ntoskrnl.exe!KeDelayExecutionThread
ntoskrnl.exe!NtDelayExecution
ntoskrnl.exe!KiSystemServiceCopyEnd
```

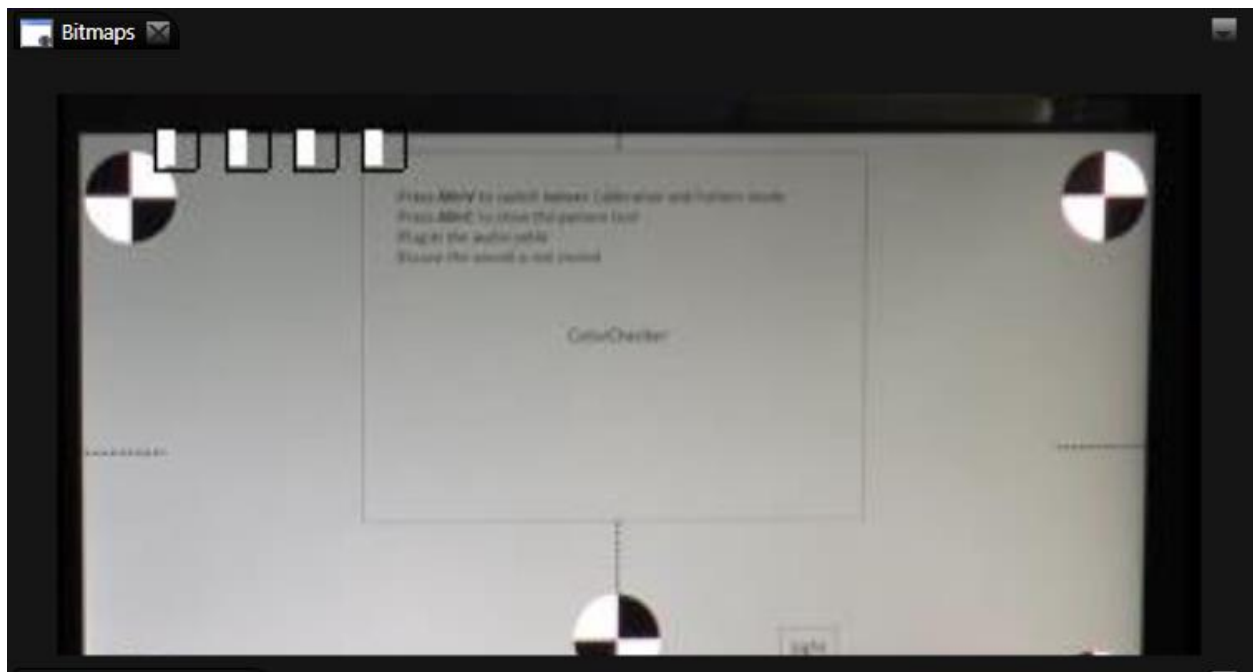
## BITMAP

The Bitmap data viewer is used to visualize video frames in a media clip. This is commonly used for Miracast performance analysis where there is a video capture of the source and sink.

To view video frames in the bitmap data viewer:

1. Load a media clip in MXA (eg. xa.exe -i 1080p.mp4)
2. Open the bitmap data viewer: View > Data viewers > Bitmap
3. Drag the video frames into a panel
4. Hover the mouse over a frame





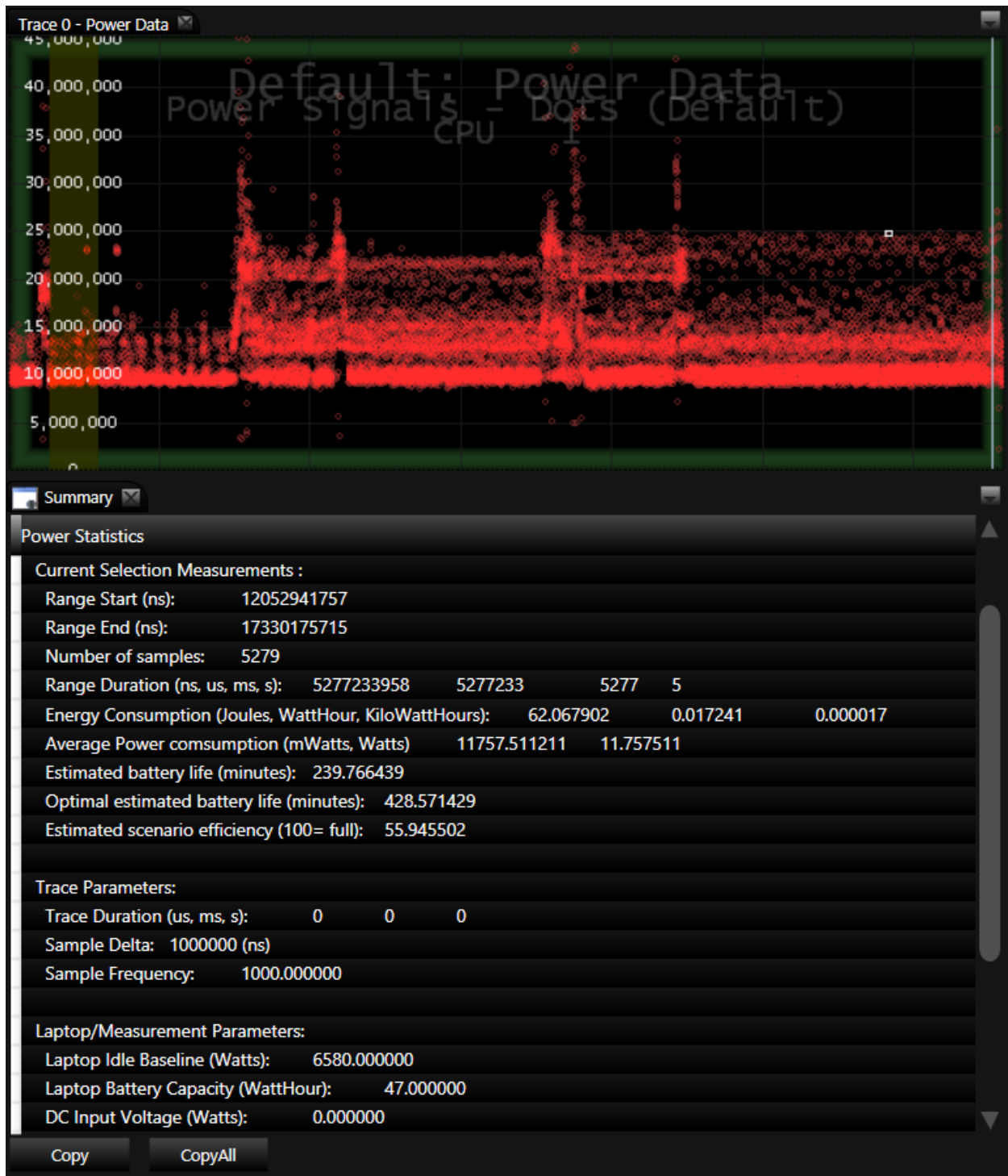
#### NOTE

The video frames are stored as images in the users temporary folder (%TEMP%) to speed up scrubbing time. MXA deletes these temp files when the app is closed. Example path to temp files:

C:\Users\<user\_name>\AppData\Local\Temp\XA\_TMP

#### SUMMARY

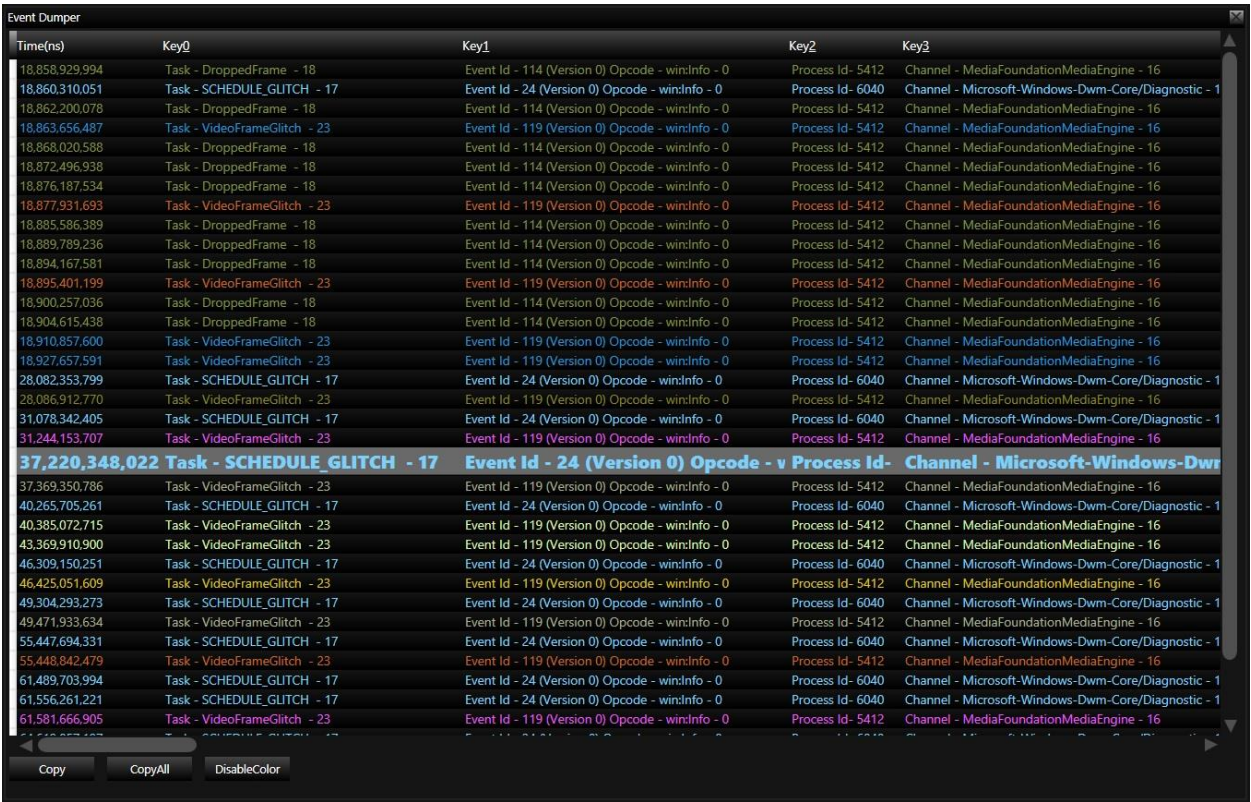
The summary dataviewer displays output from custom MXA plugins. For example, the custom power parser plugin generates power statistics and displays them in the summary dataviewer. The following screenshot shows that for a selected range of the power trace (ALT + left mouse click + drag and drop mouse), the system consumed an average of 11.75 Watts and the estimated battery life for this workload is 239 minutes.



## EVENT DUMPER

The event dumper data viewer displays a text version of the events the mouse is currently hovering over. The row that is displayed in larger font size is the event that is currently selected. The event

dumper is commonly used to view a small number of events at one time. It’s also helpful when you need to export a text version of the events by pressing the “Copy” or “Copy All” buttons.



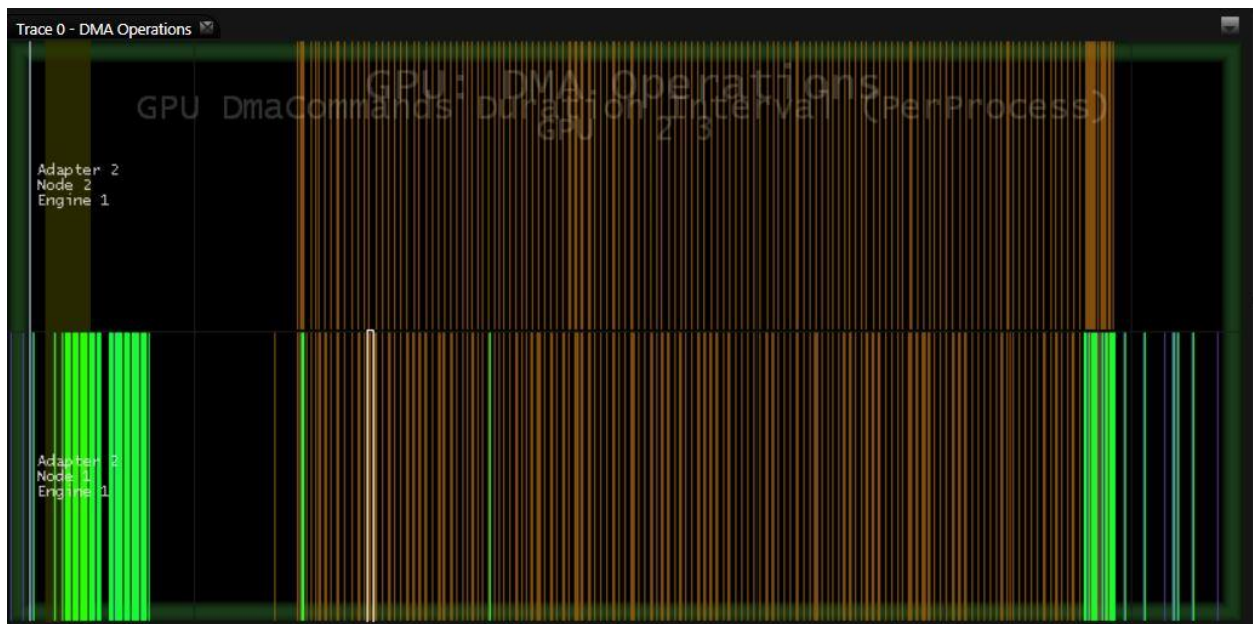
The “Disable Color” button changes the text to all of the same color instead of using the color that was assigned to the element.

## DATASETS

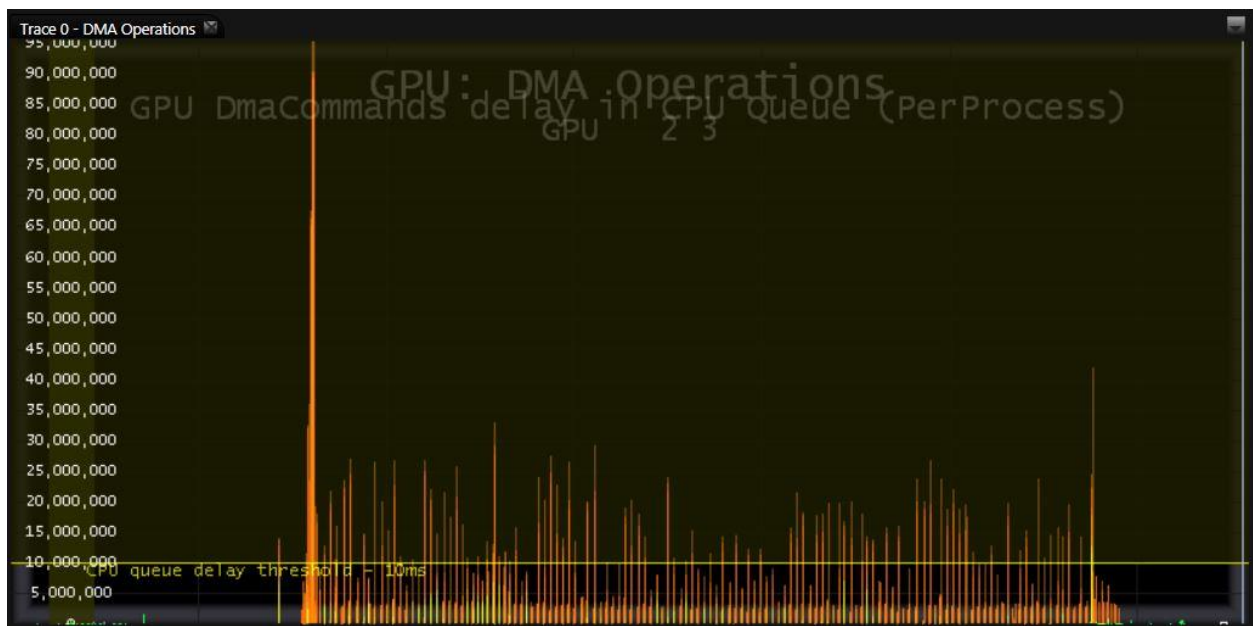
### GPU

#### DMA Operations

The DMA Operations dataset displays all DMA operations on the GPU. The Y axis in the default pivot provides activity per GPU adapter, node, and engine.



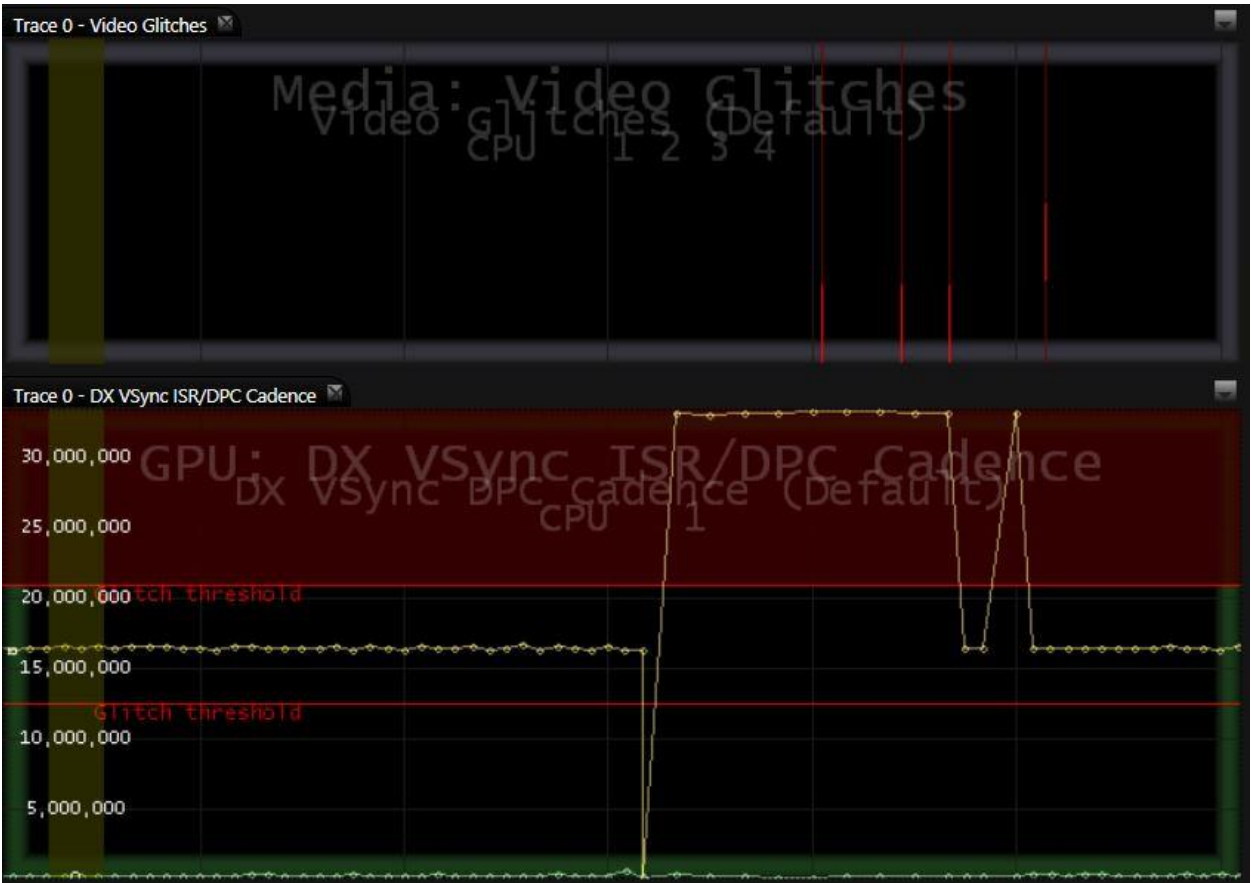
Other pivots provide graphs that display the amount of time the DMA packets spend in the CPU and GPU queues.



## DX VSync ISR/DPC Cadence

This dataset graphs the time between Vsync interrupts. For example, when a Windows device is rendering at 60hz during full screen video playback, the expected time between DX VSync interrupts is 16.6ms. As shown in the following screenshot, when the VSync cadence jumps far from the expected

value, video glitch events are logged. See the [Media](#) video glitch dataset for more information on video glitches.



MEDIA

Video Glitches

The video glitches dataset in MXA is a collection of events that indicate video frames were dropped or late. The following events are included in this list.

Event Provider & Task	Description
Microsoft-Windows-MediaEngine Task - VideoFrameGlitch - 23	Indicates a video frame was rendered late. The FramesLate event property value indicates how many frames were late, or the severity of the glitch.

Microsoft-Windows-MediaEngine Task - DroppedFrame - 18	Indicates the media source dropped a frame. This condition is typically caused by network or disk starvation.
Microsoft-Windows-Dwm-Core Task - SCHEDULE_GLITCH - 17	Indicates the DWM (Desktop Window Manager) rendered a frame late. The tGlitch event property value indicates how late the frame was (eg. 38.77ms which is about 2 vsyncs on a 60hz display).

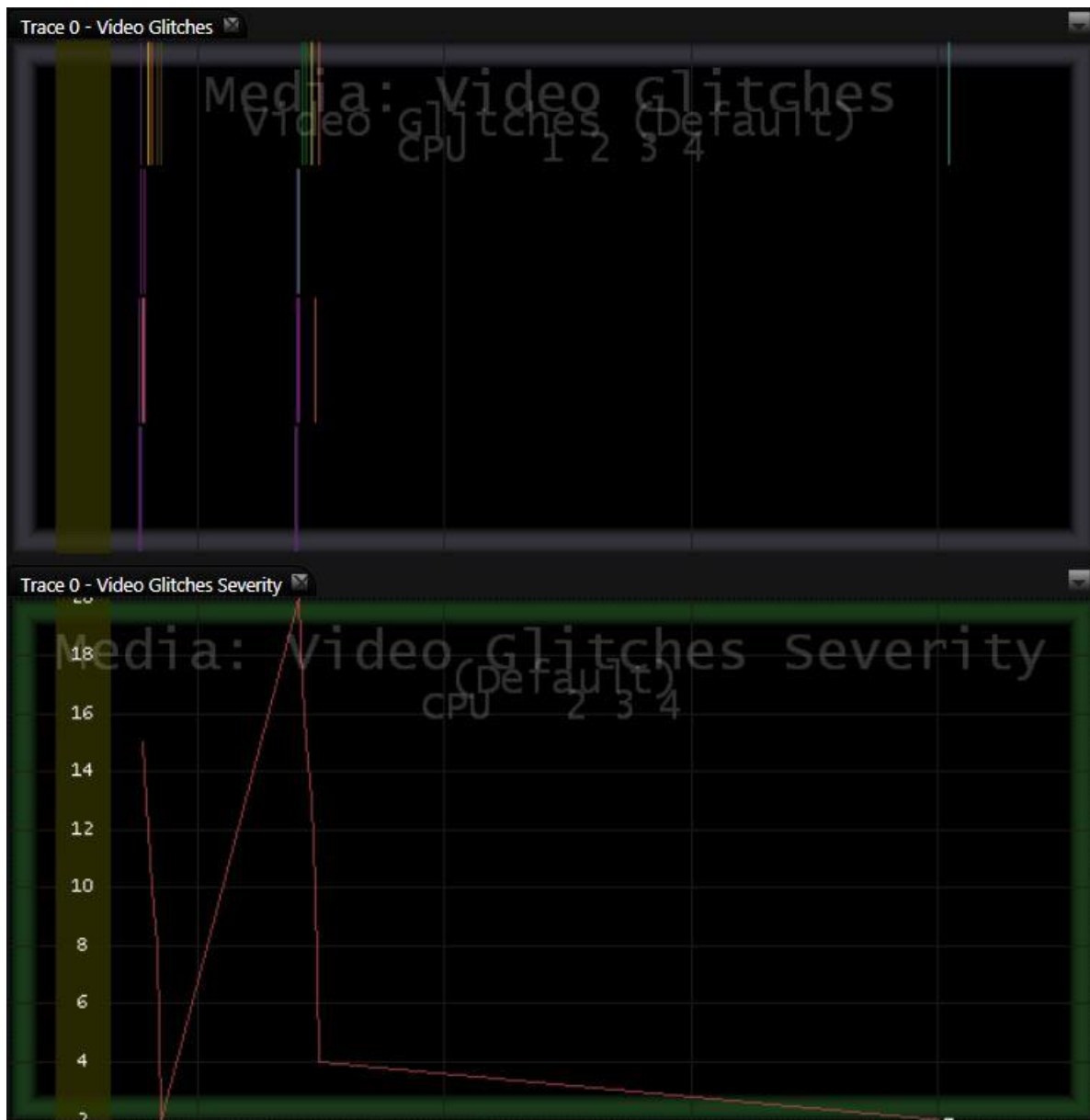
The Y axis contains a row for each core on the system so if the thread that fired a video glitch event was running on core 0, the glitch event appears on the bottom row.

### Video Glitches Severity

The video glitches severity dataset provides a graph of how severity of the Media Engine VideoFrameGitch events. The Y axis is the FramesLate value of the Media Engine Video glitch event. This pivot enables engineers to focus on the most severe video glitches. The higher the FramesLate value, the more severe the video glitch is. In the screenshot below there are 2 panels.

- **Top panel** - The lines in the top panel represent VideoFrameGitch events logged by Media Engine.
- **Bottom panel** - Provides a pivot on the FramesLate event property of the VideoFrameGitch event. The glitches that have FramesLate values of 14 – 20 are significantly more severe than the glitches that were 1 or 2 frames late.





## QUICK REFERENCE

## KEYBOARD SHORTCUTS

	Action	Shortcut	Purpose
	Open a <b>new</b> panel	Ctrl + N	When you create a new panel, it provides more real-estate to drag other datasets into.
	Open the <b>Offset</b> Tab	Ctrl + O	This is used to align traces when loading multiple traces in the same timeline.
Data Viewers	Open the <b>CallStack</b> data viewer	Ctrl + C	The CallStack data viewer displays call stacks for events in the CPU Scheduler, CPU Sampled Profile, and general instrumentation datasets.
	Open the <b>Properties</b> data viewer	Ctrl + P	View properties of an event payload
	Open the <b>Summary</b> data viewer	Ctrl + S	View summary information of a dataset or output from a custom MXA plugin such as the power plugin.
	Open the <b>Event Dumper</b> data viewer	Ctrl + E	Displays events in a flat text based list.
Toolbars	Open the <b>Navigation</b> toolbar	Ctrl + V	Contains shortcuts for changing the zoom level and panning.
	Open the <b>Mode</b> Toolbar	Ctrl + M	Switch color and drawing modes.
	Open the <b>Control</b> Toolbar	Ctrl + L	Modify settings such as whether or not to display the vertical line across all panels or toggle the bottom ruler.
	Open the <b>CPU</b> Toolbar	Ctrl + U	Display or hide all activity in a panel that contains a CPU related dataset.



	Action	Shortcut	Purpose
Machine Info	Open the <b>General</b> Tab	Ctrl + G	Displays general hardware system information of the system that the trace was collected on.
	Open the <b>Display</b> Tab	Ctrl + D	Displays graphics hardware information of the system that the trace was collected on.
	Open the <b>Network</b> Tab	Ctrl + T	Displays networking hardware information of the system that the trace was collected on.

## NAVIGATION

Action	Shortcut
Zoom in on a specific part of the timeline	Click and drag the mouse cursor over the portion of the timeline that you want to zoom in on.
Zoom out to full timeline	Press Esc
Zoom in	Press < or z Use the scroll wheel on mouse Click “zoom in” on the Navigation toolbar
Zoom out	Press > or a Use the scroll wheel on mouse Click “zoom in” on the Navigation toolbar

Action	Shortcut
Auto zoom on the Y axis	Press `
Undo Zoom	Click “undo zoom” on the Navigation toolbar
Redo Zoom	Click “redo zoom” on the Navigation toolbar
Toggle between zooming in on full and partial part of Y axis	Press Y
Pan left, right, up, or down	<p>Press left, right, up, or down arrow keys</p> <p>Press SHIFT + Right-click on the mouse and drag the data in the panel</p>

## COMMON TASKS

Action	Shortcut
Change the pivot in a dataset	<p>Option 1: Set focus in the panel and press d or f.</p> <p>Option 2: Right-click the panel and select a data pivot.</p> <p>Option 3: Click on “Toggle Drawing Mode Backward” or “Toggle Drawing Mode Forward” in the Mode toolbar.</p>
Measure a distance on the timeline	Press and hold down shift while left clicking and drag the mouse cursor over the part of the timeline to be measured.

Action	Shortcut
View event payload information	Click <b>View &gt; Data Viewers &gt; Properties</b> to open the data viewer, and then hover over event.
Freeze information in the <b>Data Viewers</b>	Press and hold down Shift.

## FREQUENTLY ASKED QUESTIONS & TROUBLESHOOTING

**Q:** What is the difference between MXA, WPA, & GPUView?

**A:** MXA provides visualizations that are complimentary to those offered in WPA and GPUView. MXA visualizations are designed to help you analyze timing and quality issues that are often found in multimedia scenarios. MXA also supports loading non-ETW trace formats such as MP4 video files and power traces in text format.

**Q:** How to freeze data inside data viewers?

**A:** Press and hold SHIFT to freeze the data displayed in data viewers.

## RELEASE NOTES

Date	Change	Comments
<b>5/23/2014</b>	MXA Published	Initial release of MXA published.
<b>10/1/2015</b>	MXA bug fix release	Contains several dozen bug fixes since the initial release.

## FEEDBACK

Please send feedback to [MXAFeedback@microsoft.com](mailto:MXAFeedback@microsoft.com)