Windows 10, version 1709 enhanced telemetry events and fields used by Windows Analytics

12/7/2017 • 9 min to read • Edit Online

Applies to

- Windows 10, version 1709 and later

Windows Analytics Device Health reports are powered by diagnostic data not included in the Basic level. This includes crash reports and certain OS telemetry events. Organizations sending Enhanced or Full level diagnostic data were able to participate in Device Health, but some organizations which required detailed event and field level documentation were unable to move from Basic to Enhanced.

In Windows 10, version 1709, we introduce a new feature: "Limit Enhanced diagnostic data to the minimum required by Windows Analytics". When enabled, this feature limits the operating system telemetry events included in the Enhanced level to only those described below. Note that the Enhanced level also includes limited crash reports, which are not described below. For more information on the Enhanced level, see Configure Windows telemetry in your organization.

KernelProcess.AppStateChangeSummary

This event summarizes application usage and performance characteristics to help Microsoft improve performance and reliability. Organizations can use this event with Windows Analytics to gain insights into application reliability.

The following fields are available:

- **CommitChargeAtExit_Sum:** Total memory commit charge for a process when it exits
- **CommitChargePeakAtExit_Sum**: Total peak memory commit charge for a process when it exits
- **ContainerId:** Server Silo Container ID
- **CrashCount:** Number of crashes for a process instance
- **CycleCountAtExit_Sum:** Total processor cycles for a process when it exited
- **ExtraInfoFlags:** Flags indicating internal states of the logging
- **GhostCount_Sum:** Total number of instances where the application stopped responding
- **HandleCountAtExit_Sum:** Total handle count for a process when it exits
- **HangCount_Max:** Maximum number of hangs detected
- **HangCount_Sum:** Total number of application hangs detected
- **HardFaultCountAtExit_Sum:** Total number of hard page faults detected for a process when it exits
- **HeartbeatCount:** Heartbeats logged for this summary
- **HeartbeatSuspendedCount:** Heartbeats logged for this summary where the process was suspended
- **LaunchCount:** Number of process instances started
- **LicenseType:** Reserved for future use
- **ProcessDurationMS_Sum:** Total duration of wall clock process instances
- **ReadCountAtExit_Sum:** Total IO reads for a process when it exited
- **ReadSizeInKBAtExit_Sum:**Total IO read size for a process when it exited
- **ResumeCount:** Number of times a process instance has resumed
- **RunningDurationMS_Sum:** Total uptime
- **SuspendCount:** Number of times a process instance was suspended

- **TargetAppId:** Application identifier
- **TargetAppType:** Application type
- **TargetAppVer:** Application version
- **TerminateCount:** Number of times a process terminated
- **WriteCountAtExit_Sum:** Total number of IO writes for a process when it exited
- **WriteSizeInKBAtExit_Sum:** Total size of IO writes for a process when it exited

Microsoft.OSG.OSS.CredProvFramework.ReportResultStop

This event indicates the result of an attempt to authenticate a user with a credential provider. It helps Microsoft to improve logon reliability. Using this event with Windows Analytics can help organizations monitor and improve logon success for different methods (for example, biometric) on managed devices.

The following fields are available:

- **CredTileProviderId:** ID of the Credential Provider
- **IsConnectedUser:** Flag indicating whether a user is connected or not
- **IsPLAPTile:** Flag indicating whether this credential tile is a pre-logon access provider or not
- **IsRemoteSession:** Flag indicating whether the session is remote or not
- **IsV2CredProv:** Flag indicating whether the credential provider of V2 or not
- **OpitonalStatusText:** Status text
- **ProcessImage:** Image path to the process
- **ProviderId:** Credential provider ID
- **ProviderStatusIcon:** Indicates which status icon should be displayed
- **ReturnCode:** Output of the ReportResult function
- **SessionId:** Session identifier
- **Sign-in error status:** The sign-in error status
- **SubStatus:** Sign-in error sub-status
- **UserTag:** Count of the number of times a user has selected a provider

Microsoft.Windows.Kernel.Power.OSStateChange

This event denotes the transition between operating system states (e.g., On, Off, Sleep, etc.). By using this event with Windows Analytics, organizations can use this to monitor reliability and performance of managed devices

The following fields are available:

- **AcPowerOnline:** If "TRUE," the device is using AC power. If "FALSE," the device is using battery power.
- **ActualTransitions:** The number of transitions between operating system states since the last system boot
- **BatteryCapacity:** Maximum battery capacity in mWh
- **BatteryCharge:** Current battery charge as a percentage of total capacity
- **BatteryDischarging:** Flag indicating whether the battery is discharging or charging
- **BootId:** Total boot count since the operating system was installed
- **BootTimeUTC:** Date and time of a particular boot event (identified by BootId)
- **EnergyChangeV2:** A snapshot value in mWh reflecting a change in power usage
- **EnergyChangeV2Flags:** Flags for disambiguating EnergyChangeV2 context
- **EventSequence:** A sequential number used to evaluate the completeness of the data
- **LastStateTransition:** ID of the last operating system state transition
- **LastStateTransitionSub:** ID of the last operating system sub-state transition
- **StateDurationMS:** Number of milliseconds spent in the last operating system state

- **StateTransition:** ID of the operating system state the system is transitioning to
- **StateTransitionSub:** ID of the operating system sub-state the system is transitioning to
- **TotalDurationMS:** Total time (in milliseconds) spent in all states since the last boot
- **TotalUptimeMS:** Total time (in milliseconds) the device was in Up or Running states since the last boot
- **TransitionsToOn:** Number of transitions to the Powered On state since the last boot
- **UptimeDeltaMS:** Total time (in milliseconds) added to Uptime since the last event

Microsoft.Windows.LogonController.LogonAndUnlockSubmit

Sends details of the user attempting to sign into or unlock the device.

The following fields are available:

- **isSystemManagedAccount:** Indicates if the user's account is System Managed
- **isUnlockScenario:** Flag indicating whether the event is a Logon or an Unlock
- **PartA_UserSid:** The security identifier of the user
- **userType:** Indicates the user type: 0 = unknown; 1 = local; 2 = Active Directory domain user; 3 = Microsoft Account; 4 = Azure Active Directory user

Microsoft.Windows.LogonController.SignInFailure

Sends details about any error codes detected during a failed sign-in.

The following fields are available:

- **ntsStatus:** The NTSTATUS error code status returned from an attempted sign-in
- **ntsSubstatus:** The NTSTATUS error code sub-status returned from an attempted sign-in

Microsoft.Windows.Security.Biometrics.Service.BioServiceActivityCapture

Indicates that a biometric capture was compared to known templates

The following fields are available:

- **captureDetail:** Result of biometric capture, either matched to an enrollment or an error
- **captureSuccessful:** Indicates whether a biometric capture was successfully matched or not
- **hardwareId:** ID of the sensor that collected the biometric capture
- **isSecureSensor:** Flag indicating whether a biometric sensor was in enhanced security mode
- **isTrustletRunning:** Indicates whether an enhanced security component is currently running
- **isVsmCfg:** Flag indicating whether virtual secure mode is configured or not

Microsoft.Windows.Security.Certificates.PinRulesCaCertUsedAnalytics

The Microsoft.Windows.Security.Certificates.Pin*Analytics events summarize which server certificates the client encounters. By using this event with Windows Analytics, organizations can use this to determine potential scope and impact of pending certificate revocations or expirations.

The following fields are available:

- **certBinary:** Binary blob of public certificate as presented to the client (does not include any private keys)
- **certThumbprint:** Certificate thumbprint

Microsoft.Windows.Security.Certificates.PinRulesCheckedAnalytics

The Microsoft.Windows.Security.Certificates.Pin*Analytics events summarize which server certificates the client encounters. By using this event with Windows Analytics, organizations can use this to determine potential scope and impact of pending certificate revocations or expirations.

The following fields are available:

- **caThumbprints:** Intermediate certificate thumbprints
- **rootThumbprint:** Root certificate thumbprint
- **serverName:** Server name associated with the certificate
- **serverThumbprint:** Server certificate thumbprint
- **statusBits:** Certificate status

Microsoft.Windows.Security.Certificates.PinRulesServerCertUsedAnalytics

The Microsoft.Windows.Security.Certificates.Pin*Analytics events summarize which server certificates the client encounters. By using this event with Windows Analytics, organizations can use this to determine potential scope and impact of pending certificate revocations or expirations.

The following fields are available:

- **certBinary:** Binary blob of public certificate as presented to the client (does not include any private keys)
- **certThumbprint:** Certificate thumbprint

Microsoft.Windows.Security.Winlogon.SystemBootStop

System boot has completed.

The following field is available:

- **ticksSinceBoot:** Duration of boot event (milliseconds)

Microsoft.Windows.Shell.Desktop.LogonFramework.AllLogonTasks

This event summarizes the logon procedure to help Microsoft improve performance and reliability. By using this event with Windows Analytics organizations can help identify logon problems on managed devices.

The following fields are available:

- **isAadUser:** Indicates whether the current logon is for an Azure Active Directory account
- **isDomainUser:** Indicates whether the current logon is for a domain account
- **isMSA:** Indicates whether the current logon is for a Microsoft Account
- **logonOptimizationFlags:** Flags indicating optimization settings for this logon session
- **logonTypeFlags:** Flags indicating logon type (first logon vs. a later logon)
- **systemManufacturer:** Device manufacturer
- **systemProductName:** Device product name
- **wilActivity:** Indicates errors in the task to help Microsoft improve reliability.

Microsoft.Windows.Shell.Desktop.LogonFramework.LogonTask

This event describes system tasks which are part of the user logon sequence and helps Microsoft to improve reliability.

The following fields are available:

- **isStartWaitTask:** Flag indicating whether the task starts a background task
- **isWaitMethod:** Flag indicating the task is waiting on a background task
- **logonTask:** Indicates which logon step is currently occurring
- **wilActivity:** Indicates errors in the task to help Microsoft improve reliability.

Microsoft.Windows.Shell.Explorer.DesktopReady

Initialization of Explorer is complete.

Microsoft-Windows-Security-EFS-EDPAudit-ApplicationLearning.EdpAuditLogApplicationLearning

For a device subject to Windows Information Protection policy, learning events are generated when an app encounters a policy boundary (for example, trying to open a work document from a personal app). These events help the WIP administrator tune policy rules and prevent unnecessary user disruption.

The following fields are available:

- **actiontype:** Indicates what type of resource access the app was attempting (for example, opening a local document vs. a network resource) when it encountered a policy boundary. Useful for Windows Information Protection administrators to tune policy rules.
- **appIdType:** Based on the type of application, this indicates what type of app rule a Windows Information Protection administrator would need to create for this app.
- **appname:** App that triggered the event
- **status:** Indicates whether errors occurred during WIP learning events

Win32kTraceLogging.AppInteractivitySummary

Summarizes which app windows are being used (for example, have focus) to help Microsoft improve compatibility and user experience. Also helps organizations (by using Windows Analytics) to understand and improve application reliability on managed devices.

The following fields are available:

- **AggregationDurationMS:** Actual duration of aggregation period (in milliseconds)
- **AggregationFlags:** Flags denoting aggregation settings
- **AggregationPeriodMS:** Intended duration of aggregation period (in milliseconds)
- **AggregationStartTime:** Start date and time of AppInteractivity aggregation
- **AppId:** Application ID for usage
- **AppSessionId:** GUID identifying the application's usage session
- **AppVersion:** Version of the application that produced this event
- **AudioInMS:** Audio capture duration (in milliseconds)
- **AudioOutMS:** Audio playback duration (in milliseconds)
- **BackgroundMouseSec:** Indicates that there was a mouse hover event while the app was in the background
- **BitPeriodMS:** Length of the period represented by InFocusBitmap
- **CommandLineHash:** A hash of the command line
- **CompositionDirtyGeneratedSec:** Represents the amount of time (in seconds) during which the active app reported that it had an update
- **CompositionDirtyPropagatedSec:** Total time (in seconds) that a separate process with visuals hosted in an app signaled updates
- **CompositionRenderedSec:** Time (in seconds) that an app's contents were rendered

- **EventSequence:** [need more info]
- **FocusLostCount:** Number of times that an app lost focus during the aggregation period
- **GameInputSec:** Time (in seconds) there was user input using a game controller
- **HidInputSec:** Time (in seconds) there was user input using devices other than a game controller
- **InFocusBitmap:** Series of bits representing application having and losing focus
- **InFocusDurationMS:** Total time (in milliseconds) the application had focus
- **InputSec:** Total number of seconds during which there was any user input
- **InteractiveTimeoutPeriodMS:** Total time (in milliseconds) that inactivity expired interactivity sessions
- **KeyboardInputSec:** Total number of seconds during which there was keyboard input
- **MonitorFlags:** Flags indicating app use of individual monitor(s)
- **MonitorHeight:** Number of vertical pixels in the application host monitor resolution
- **MonitorWidth:** Number of horizontal pixels in the application host monitor resolution
- **MouseInputSec:** Total number of seconds during which there was mouse input
- **NewProcessCount:** Number of new processes contributing to the aggregate
- **PartATransform_AppSessionGuidToUserSid:** Flag which influences how other parts of the event are constructed
- **PenInputSec:** Total number of seconds during which there was pen input
- **SpeechRecognitionSec:** Total number of seconds of speech recognition
- **SummaryRound:** Incrementing number indicating the round (batch) being summarized
- **TargetAsId:** Flag which influences how other parts of the event are constructed
- **TotalUserOrDisplayActiveDurationMS:** Total time the user or the display was active (in milliseconds)
- **TouchInputSec:** Total number of seconds during which there was touch input
- **UserActiveDurationMS:** Total time that the user was active including all input methods
- **UserActiveTransitionCount:** Number of transitions in and out of user activity
- **UserOrDisplayActiveDurationMS:** Total time the user was using the display
- **ViewFlags:** Flags denoting properties of an app view (for example, special VR view or not)
- **WindowFlags:** Flags denoting runtime properties of an app window
- **WindowHeight:** Number of vertical pixels in the application window
- **WindowWidth:** Number of horizontal pixels in the application window