Windows 10, version 1703 basic level Windows diagnostic events and fields

12/7/2017 • 157 min to read • Edit Online

Applies to

- Windows 10, version 1703

The Basic level gathers a limited set of information that is critical for understanding the device and its configuration including: basic device information, quality-related information, app compatibility, and Microsoft Store. When the level is set to Basic, it also includes the Security level information. The Basic level also helps to identify problems that can occur on a particular device hardware or software configuration. For example, it can help determine if crashes are more frequent on devices with a specific amount of memory or that are running a particular driver version. This helps Microsoft fix operating system or app problems.

Use this article to learn about diagnostic events, grouped by event area, and the fields within each event. A brief description is provided for each field. Every event generated includes common data, which collects device data. You can learn more about Windows functional and diagnostic data through these articles:

- Manage connections from Windows operating system components to Microsoft services
- Configure Windows telemetry in your organization

> **NOTE**
>
> Updated November 2017 to document new and modified events. We've added some new events and also added new fields to existing events to prepare for upgrades to the next release of Windows.

Common data extensions

Common Data Extensions.App

The following fields are available:

- **expId** Associates a flight, such as an OS flight, or an experiment, such as a web site UX experiment, with an event.
- **userId** The userID as known by the application.
- **env** The environment from which the event was logged.
- **asId** An integer value that represents the app session. This value starts at 0 on the first app launch and increments after each subsequent app launch per boot session.

Common Data Extensions.CS

The following fields are available:

- **sig** A common schema signature that identifies new and modified event schemas.

Common Data Extensions.CUET

The following fields are available:

- **stId** Represents the Scenario Entry Point ID. This is a unique GUID for each event in a diagnostic scenario. This used to be Scenario Trigger ID.
- **aId** Represents the ETW ActivityId. Logged via TraceLogging or directly via ETW.

- **rald** Represents the ETW Related ActivityId. Logged via TraceLogging or directly via ETW.
- **op** Represents the ETW Op Code.
- **cat** Represents a bitmask of the ETW Keywords associated with the event.
- **flags** Represents the bitmap that captures various Windows specific flags.
- **cpId** The composer ID, such as Reference, Desktop, Phone, Holographic, Hub, IoT Composer.
- **tickets** A list of strings that represent entries in the HTTP header of the web request that includes this event.
- **bseq** Upload buffer sequence number in the format <buffer identifier>:<sequence number>
- **mon** Combined monitor and event sequence numbers in the format <monitor sequence>:<event sequence>

Common Data Extensions.Device

The following fields are available:

- **ver** Represents the major and minor version of the extension.
- **localId** Represents a locally defined unique ID for the device, not the human readable device name. Most likely equal to the value stored at HKLM\Software\Microsoft\SQMClient\MachineId
- **deviceClass** Represents the classification of the device, the device "family". For example, Desktop, Server, or Mobile.

Common Data Extensions.Envelope

The following fields are available:

- **ver** Represents the major and minor version of the extension.
- **name** Represents the uniquely qualified name for the event.
- **time** Represents the event date time in Coordinated Universal Time (UTC) when the event was generated on the client. This should be in ISO 8601 format.
- **popSample** Represents the effective sample rate for this event at the time it was generated by a client.
- **epoch** Represents the epoch and seqNum fields, which help track how many events were fired and how many events were uploaded, and enables identification of data lost during upload and de-duplication of events on the ingress server.
- **seqNum** Represents the sequence field used to track absolute order of uploaded events. It is an incrementing identifier for each event added to the upload queue. The Sequence helps track how many events were fired and how many events were uploaded and enables identification of data lost during upload and de-duplication of events on the ingress server.
- **iKey** Represents an ID for applications or other logical groupings of events.
- **flags** Represents a collection of bits that describe how the event should be processed by the Connected User Experience and Telemetry component pipeline. The lowest-order byte is the event persistence. The next byte is the event latency.
- **os** Represents the operating system name.
- **osVer** Represents the OS version, and its format is OS dependent.
- **appId** Represents a unique identifier of the client application currently loaded in the process producing the event; and is used to group events together and understand usage pattern, errors by application.
- **appVer** Represents the version number of the application. Used to understand errors by Version, Usage by Version across an app.
- **cV** Represents the Correlation Vector: A single field for tracking partial order of related telemetry events across component boundaries.

Common Data Extensions.OS

The following fields are available:

- **ver** Represents the major and minor version of the extension.
- **expId** Represents the experiment ID. The standard for associating a flight, such as an OS flight (pre-release

build), or an experiment, such as a web site UX experiment, with an event is to record the flight / experiment IDs in Part A of the common schema.

- **locale** Represents the locale of the operating system.
- **bootId** An integer value that represents the boot session. This value starts at 0 on first boot after OS install and increments after every reboot.

Common Data Extensions.User

The following fields are available:

- **ver** Represents the major and minor version of the extension.
- **localId** Represents a unique user identity that is created locally and added by the client. This is not the user's account ID.

Common Data Extensions.XBL

The following fields are available:

- **nbf** Not before time
- **expId** Expiration time
- **sbx** XBOX sandbox identifier
- **dty** XBOX device type
- **did** XBOX device ID
- **xid** A list of base10-encoded XBOX User IDs.
- **uts** A bit field, with 2 bits being assigned to each user ID listed in xid. This field is omitted if all users are retail accounts.

Common Data Extensions.Consent UI Event

This User Account Control (UAC) telemetry point collects information on elevations that originate from low integrity levels. This occurs when a process running at low integrity level (IL) requires higher (administrator) privileges, and therefore requests for elevation via UAC (consent.exe). By better understanding the processes requesting these elevations, Microsoft can in turn improve the detection and handling of potentially malicious behavior in this path.

The following fields are available:

- **eventType** Represents the type of elevation: If it succeeded, was cancelled, or was auto-approved.
- **splitToken** Represents the flag used to distinguish between administrators and standard users.
- **friendlyName** Represents the name of the file requesting elevation from low IL.
- **elevationReason** Represents the distinction between various elevation requests sources (appcompat, installer, COM, MSI and so on).
- **exeName** Represents the name of the file requesting elevation from low IL.
- **signatureState** Represents the state of the signature, if it signed, unsigned, OS signed and so on.
- **publisherName** Represents the name of the publisher of the file requesting elevation from low IL.
- **cmdLine** Represents the full command line arguments being used to elevate.
- **Hash.Length** Represents the length of the hash of the file requesting elevation from low IL.
- **Hash** Represents the hash of the file requesting elevation from low IL.
- **HashAlgId** Represents the algorithm ID of the hash of the file requesting elevation from low IL.
- **telemetryFlags** Represents the details about the elevation prompt for CEIP data.
- **timeStamp** Represents the time stamp on the file requesting elevation.
- **fileVersionMS** Represents the major version of the file requesting elevation.
- **fileVersionLS** Represents the minor version of the file requesting elevation.

Common data fields

Common Data Fields.MS.Device.DeviceInventory.Change

These fields are added whenever Ms.Device.DeviceInventoryChange is included in the event.

The following fields are available:

- **syncId** A string used to group StartSync, EndSync, Add, and Remove operations that belong together. This field is unique by Sync period and is used to disambiguate in situations where multiple agents perform overlapping inventories for the same object.
- **objectType** Indicates the object type that the event applies to.
- **Action** The change that was invoked on a device inventory object.
- **inventoryId** Device ID used for Compatibility testing

Common Data Fields.TelClientSynthetic.PrivacySettingsAfterCreatorsUpdate.PreUpgradeSettings

These fields are added whenever PreUpgradeSettings is included in the event.

The following fields are available:

- **HKLM_SensorPermissionState.SensorPermissionState** The state of the Location service before the feature update completed.
- **HKLM_SensorPermissionState.HRESULT** The error code returned when trying to query the Location service for the device.
- **HKCU_SensorPermissionState.SensorPermissionState** The state of the Location service when a user signs on before the feature update completed.
- **HKCU_SensorPermissionState.HRESULT** The error code returned when trying to query the Location service for the current user.
- **HKLM_LocationPlatform.Status** The state of the location platform after the feature update has completed.
- **HKLM_LocationPlatform.HRESULT** The error code returned when trying to query the location platform for the device.
- **HKLM_LocationSyncEnabled.AcceptedPrivacyPolicy** The speech recognition state for the device before the feature update completed.
- **HKLM_LocationSyncEnabled.HRESULT** The error code returned when trying to query the Find My Device service for the device.
- **HKCU_LocationSyncEnabled.AcceptedPrivacyPolicy** The speech recognition state for the current user before the feature update completed.
- **HKCU_LocationSyncEnabled.HRESULT** The error code returned when trying to query the Find My Device service for the current user.
- **HKLM_AllowTelemetry.AllowTelemetry** The state of the Connected User Experiences and Telemetry component for the device before the feature update.
- **HKLM_AllowTelemetry.HRESULT** The error code returned when trying to query the Connected User Experiences and Telemetry conponent for the device.
- **HKLM_TIPC.Enabled** The state of TIPC for the device.
- **HKLM_TIPC.HRESULT** The error code returned when trying to query TIPC for the device.
- **HKCU_TIPC.Enabled** The state of TIPC for the current user.
- **HKCU_TIPC.HRESULT** The error code returned when trying to query TIPC for the current user.
- **HKLM_FlipAhead.FPEnabled** Is Flip Ahead enabled for the device before the feature update was completed?
- **HKLM_FlipAhead.HRESULT** The error code returned when trying to query Flip Ahead for the device.
- **HKCU_FlipAhead.FPEnabled** Is Flip Ahead enabled for the current user before the feature update was completed?
- **HKCU_FlipAhead.HRESULT** The error code returned when trying to query Flip Ahead for the current user.

- **HKLM_TailoredExperiences.TailoredExperiencesWithDiagnosticDataEnabled** Is Tailored Experiences with Diagnostics Data enabled for the current user after the feature update had completed?
- **HKCU_TailoredExperiences.HRESULT** The error code returned when trying to query Tailored Experiences with Diagnostics Data for the current user.
- **HKLM_AdvertisingID.Enabled** Is the adveristing ID enabled for the device?
- **HKLM_AdvertisingID.HRESULT** The error code returned when trying to query the state of the advertising ID for the device.
- **HKCU_AdvertisingID.Enabled** Is the adveristing ID enabled for the current user?
- **HKCU_AdvertisingID.HRESULT** The error code returned when trying to query the state of the advertising ID for the user.

Common Data Fields.TelClientSynthetic.PrivacySettingsAfterCreatorsUpdate.PostUpgradeSettings

These fields are added whenever PostUpgradeSettings is included in the event.

The following fields are available:

- **HKLM_SensorPermissionState.SensorPermissionState** The state of the Location service after the feature update has completed.
- **HKLM_SensorPermissionState.HRESULT** The error code returned when trying to query the Location service for the device.
- **HKCU_SensorPermissionState.SensorPermissionState** The state of the Location service when a user signs on after a feature update has completed.
- **HKCU_SensorPermissionState.HRESULT** The error code returned when trying to query the Location service for the current user.
- **HKLM_LocationPlatform.Status** The state of the location platform after the feature update has completed.
- **HKLM_LocationPlatform.HRESULT** The error code returned when trying to query the location platform for the device.
- **HKLM_LocationSyncEnabled.AcceptedPrivacyPolicy** The speech recognition state for the device after the feature update has completed.
- **HKLM_LocationSyncEnabled.HRESULT** The error code returned when trying to query the Find My Device service for the device.
- **HKCU_LocationSyncEnabled.AcceptedPrivacyPolicy** The speech recognition state for the current user after the feature update has completed.
- **HKCU_LocationSyncEnabled.HRESULT** The error code returned when trying to query the Find My Device service for the current user.
- **HKLM_AllowTelemetry.AllowTelemetry** The state of the Connected User Experiences and Telemetry component for the device after the feature update.
- **HKLM_AllowTelemetry.HRESULT** The error code returned when trying to query the Connected User Experiences and Telemetry conponent for the device.
- **HKLM_TIPC.Enabled** The state of TIPC for the device.
- **HKLM_TIPC.HRESULT** The error code returned when trying to query TIPC for the device.
- **HKCU_TIPC.Enabled** The state of TIPC for the current user.
- **HKCU_TIPC.HRESULT** The error code returned when trying to query TIPC for the current user.
- **HKLM_FlipAhead.FPEnabled** Is Flip Ahead enabled for the device after the feature update has completed?
- **HKLM_FlipAhead.HRESULT** The error code returned when trying to query Flip Ahead for the device.
- **HKCU_FlipAhead.FPEnabled** Is Flip Ahead enabled for the current user after the feature update has completed?
- **HKCU_FlipAhead.HRESULT** The error code returned when trying to query Flip Ahead for the current user.
- **HKLM_TailoredExperiences.TailoredExperiencesWithDiagnosticDataEnabled** Is Tailored Experiences with Diagnostics Data enabled for the current user after the feature update had completed?

- **HKCU_TailoredExperiences.HRESULT** The error code returned when trying to query Tailored Experiences with Diagnostics Data for the current user.
- **HKLM_AdvertisingID.Enabled** Is the adveristing ID enabled for the device?
- **HKLM_AdvertisingID.HRESULT** The error code returned when trying to query the state of the advertising ID for the device.
- **HKCU_AdvertisingID.Enabled** Is the adveristing ID enabled for the current user?
- **HKCU_AdvertisingID.HRESULT** The error code returned when trying to query the state of the advertising ID for the user.

Appraiser events

Microsoft.Windows.Appraiser.General.ChecksumTotalPictureCount

This event lists the types of objects and how many of each exist on the client device. This allows for a quick way to ensure that the records present on the server match what is present on the client.

The following fields are available:

- **DatasourceApplicationFile_RS3** The total DecisionApplicationFile objects targeting the next release of Windows on this device. on this device.
- **DatasourceDevicePnp_RS3** The total DatasourceDevicePnp objects targeting the next release of Windows on this device.
- **DatasourceDriverPackage_RS3** The total DatasourceDriverPackage objects targeting the next release of Windows on this device.
- **DataSourceMatchingInfoBlock_RS3** The total DataSourceMatchingInfoBlock objects targeting the next release of Windows on this device.
- **DataSourceMatchingInfoPassive_RS3** The total DataSourceMatchingInfoPassive objects targeting the next release of Windows on this device.
- **DataSourceMatchingInfoPostUpgrade_RS3** The total DataSourceMatchingInfoPostUpgrade objects targeting the next release of Windows on this device.
- **DatasourceSystemBios_RS3** The total DatasourceSystemBios objects targeting the next release of Windows on this device.
- **DecisionApplicationFile_RS3** The total DecisionApplicationFile objects targeting the next release of Windows on this device.
- **DecisionDevicePnp_RS3** The total DecisionDevicePnp objects targeting the next release of Windows on this device.
- **DecisionDriverPackage_RS3** The total DecisionDriverPackage objects targeting the next release of Windows on this device.
- **DecisionMatchingInfoBlock_RS3** The total DecisionMatchingInfoBlock objects targeting the next release of Windows on this device.
- **DecisionMatchingInfoPassive_RS3** The total DataSourceMatchingInfoPostUpgrade objects targeting the next release of Windows on this device.
- **DecisionMatchingInfoPostUpgrade_RS3** The total DecisionMatchingInfoPostUpgrade objects targeting the next release of Windows on this device.
- **DecisionMediaCenter_RS3** The total DecisionMediaCenter objects targeting the next release of Windows on this device.
- **DecisionSystemBios_RS3** The total DecisionSystemBios objects targeting the next release of Windows on this device.
- **PCFP** An ID for the system that is calculated by hashing hardware identifiers.
- **InventoryApplicationFile** The total InventoryApplicationFile objects that are present on this device.
- **InventoryMediaCenter** The total InventoryMediaCenter objects that are present on this device.

- **InventoryLanguagePack** The total InventoryLanguagePack objects that are present on this device.
- **InventoryUplevelDriverPackage** The total InventoryUplevelDriverPackage objects that are present on this device.
- **InventorySystemBios** The total InventorySystemBios objects that are present on this device.
- **SystemProcessorCompareExchange** The total SystemProcessorCompareExchange objects that are present on this device.
- **SystemProcessorLahfSahf** The total SystemProcessorLahfSahf objects that are present on this device.
- **SystemMemory** The total SystemMemory objects that are present on this device.
- **SystemProcessorPrefetchW** The total SystemProcessorPrefetchW objects that are present on this device.
- **SystemProcessorSse2** The total SystemProcessorSse2 objects that are present on this device.
- **SystemProcessorNx** The total SystemProcessorNx objects that are present on this device.
- **SystemWlan** The total SystemWlan objects that are present on this device.
- **SystemWim** The total SystemWim objects that are present on this device
- **SystemTouch** The total SystemTouch objects that are present on this device.
- **SystemWindowsActivationStatus** The total SystemWindowsActivationStatus objects that are present on this device.
- **Wmdrm_RS3** The total Wmdrm objects targeting the next release of Windows on this device.

Microsoft.Windows.Appraiser.General.ChecksumTotalPictureIdHashSha256

This event lists the types of objects and the hashed values of all the identifiers for each one. This allows for a more in-depth way to ensure that the records present on the server match what is present on the client.

The following fields are available:

- **DatasourceApplicationFile_RS3** The total DatasourceApplicationFile objects targeting the next release of Windows on this device.
- **DatasourceDevicePnp_RS3** The total DatasourceDevicePnp objects targeting the next release of Windows on this device.
- **DatasourceDriverPackage_RS3** The total DatasourceDriverPackage objects targeting the next release of Windows on this device.
- **DataSourceMatchingInfoBlock_RS3** The total DataSourceMatchingInfoBlock objects targeting the next release of Windows on this device.
- **DataSourceMatchingInfoPassive_RS3** The total DataSourceMatchingInfoPassive objects targeting the next release of Windows on this device.
- **DataSourceMatchingInfoPostUpgrade_RS3** The total DataSourceMatchingInfoPostUpgrade objects targeting the next release of Windows on this device.
- **DatasourceSystemBios_RS3** The total DatasourceSystemBios objects targeting the next release of Windows on this device.
- **DecisionApplicationFile_RS3** The total DecisionApplicationFile objects targeting the next release of Windows on this device.
- **DecisionDevicePnp_RS3** The total DecisionDevicePnp objects targeting the next release of Windows on this device.
- **DecisionDriverPackage_RS3** The total DecisionDriverPackage objects targeting the next release of Windows on this device.
- **DecisionMatchingInfoBlock_RS3** The total DecisionMatchingInfoBlock objects targeting the next release of Windows on this device.
- **DecisionMatchingInfoPassive_RS3** The total DataSourceMatchingInfoPostUpgrade objects targeting the next release of Windows on this device.
- **DecisionMatchingInfoPostUpgrade_RS3** The total DecisionMatchingInfoPostUpgrade objects targeting the next release of Windows on this device.

- **DecisionMediaCenter_RS3** The total DecisionMediaCenter objects targeting the next release of Windows on this device.
- **DecisionSystemBios_RS3** The total DecisionSystemBios objects targeting the next release of Windows on this device.
- **PCFP** An ID for the system that is calculated by hashing hardware identifiers.
- **InventoryApplicationFile** The SHA256 hash of InventoryApplicationFile objects that are present on this device.
- **InventoryMediaCenter** The SHA256 hash of InventoryMediaCenter objects that are present on this device.
- **InventoryLanguagePack** The SHA256 hash of InventoryLanguagePack objects that are present on this device.
- **InventoryUplevelDriverPackage** The SHA256 hash of InventoryUplevelDriverPackage objects that are present on this device.
- **InventorySystemBios** The SHA256 hash of InventorySystemBios objects that are present on this device.
- **SystemProcessorCompareExchange** The SHA256 hash of SystemProcessorCompareExchange objects that are present on this device.
- **SystemProcessorLahfSahf** The SHA256 hash of SystemProcessorLahfSahf objects that are present on this device.
- **SystemMemory** The SHA256 hash of SystemMemory objects that are present on this device.
- **SystemProcessorPrefetchW** The SHA256 hash of SystemProcessorPrefetchW objects that are present on this device.
- **SystemProcessorSse2** The SHA256 hash of SystemProcessorSse2 objects that are present on this device.
- **SystemProcessorNx** The SHA256 hash of SystemProcessorNx objects that are present on this device.
- **SystemWlan** The SHA256 hash of SystemWlan objects that are present on this device.
- **SystemWim** The SHA256 hash of SystemWim objects that are present on this device.
- **SystemTouch** The SHA256 hash of SystemTouch objects that are present on this device.
- **SystemWindowsActivationStatus** The SHA256 hash of SystemWindowsActivationStatus objects that are present on this device.
- **Wmdrm_RS3** The total Wmdrm objects targeting the next release of Windows on this device.

Microsoft.Windows.Appraiser.General.DatasourceApplicationFileAdd

This event sends compatibility information about a file to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file that is generating the events.
- **AvDisplayName** If it is an anti-virus app, this is its display name.
- **CompatModelIndex** The compatibility prediction for this file.
- **HasCitData** Is the file present in CIT data?
- **HasUpgradeExe** Does the anti-virus app have an upgrade.exe file?
- **IsAv** Is the file an anti-virus reporting EXE?
- **ResolveAttempted** This will always be an empty string when sending telemetry.
- **SdbEntries** An array of fields that indicates the SDB entries that apply to this file.

Microsoft.Windows.Appraiser.General.DatasourceApplicationFileRemove

This event indicates that the DatasourceApplicationFile object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DatasourceApplicationFileStartSync

This event indicates that a new set of DatasourceApplicationFileAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DatasourceDevicePnpAdd

This event sends compatibility data for a PNP device, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file generating the events.
- **ActiveNetworkConnection** Is the device an active network device?
- **IsBootCritical** Is the device boot critical?
- **SdbEntries** An array of fields indicating the SDB entries that apply to this device.
- **WuDriverCoverage** Is there a driver uplevel for this device according to Windows Update?
- **WuDriverUpdateID** The Windows Update ID of the applicable uplevel driver.
- **WuPopulatedFromID** The expected uplevel driver matching ID based on driver coverage from Windows Update.

Microsoft.Windows.Appraiser.General.DatasourceDevicePnpRemove

This event indicates that the DatasourceDevicePnp object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DatasourceDevicePnpStartSync

This event indicates that a new set of DatasourceDevicePnpAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DatasourceDriverPackageAdd

This event sends compatibility database data about driver packages to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file generating the events.
- **SdbEntries** An array of fields indicating the SDB entries that apply to this driver package.

Microsoft.Windows.Appraiser.General.DatasourceDriverPackageRemove

This event indicates that the DatasourceDriverPackage object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DatasourceDriverPackageStartSync

This event indicates that a new set of DatasourceDriverPackageAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoBlockAdd

This event sends blocking data about any compatibility blocking entries hit on the system that are not directly related to specific applications or devices, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file generating the events.
- **SdbEntries** An array of fields indicating the SDB entries that apply to this file.

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoBlockRemove

This event indicates that the DataSourceMatchingInfoBlock object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoBlockStartSync

This event indicates that a full set of DataSourceMatchingInfoBlockStAdd events have been sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoPassiveAdd

This event sends compatibility database information about non-blocking compatibility entries on the system that are not keyed by either applications or devices, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file generating the events.
- **SdbEntries** An array of fields indicating the SDB entries that apply to this file.

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoPassiveRemove

This event indicates that the DataSourceMatchingInfoPassive object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoPassiveStartSync

This event indicates that a new set of DataSourceMatchingInfoPassiveAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoPostUpgradeAdd

This event sends compatibility database information about entries requiring reinstallation after an upgrade on the system that are not keyed by either applications or devices, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file generating the events.
- **SdbEntries** An array of fields indicating the SDB entries that apply to this file.

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoPostUpgradeRemove

This event indicates that the DataSourceMatchingInfoPostUpgrade object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoPostUpgradeStartSync

This event indicates that a new set of DataSourceMatchingInfoPostUpgradeAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DatasourceSystemBiosAdd

This event sends compatibility database information about the BIOS to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file generating the events.
- **SdbEntries** An array of fields indicating the SDB entries that apply to this BIOS.

Microsoft.Windows.Appraiser.General.DatasourceSystemBiosRemove

This event indicates that the DatasourceSystemBios object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DatasourceSystemBiosStartSync

This event indicates that a new set of DatasourceSystemBiosAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionApplicationFileAdd

This event sends compatibility decision data about a file to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file generating the events.
- **BlockAlreadyInbox** The uplevel runtime block on the file already existed on the current OS.
- **BlockingApplication** Are there any application issues that interfere with upgrade due to the file in question?
- **DisplayGenericMessage** Will be a generic message be shown for this file?
- **HardBlock** This file is blocked in the SDB.
- **HasUxBlockOverride** Does the file have a block that is overridden by a tag in the SDB?
- **MigApplication** Does the file have a MigXML from the SDB associated with it that applies to the current upgrade mode?
- **MigRemoval** Does the file have a MigXML from the SDB that will cause the app to be removed on upgrade?
- **NeedsDismissAction** Will the file cause an action that can be dimissed?
- **NeedsInstallPostUpgradeData** After upgrade, the file will have a post-upgrade notification to install a replacement for the app.
- **NeedsNotifyPostUpgradeData** Does the file have a notification that should be shown after upgrade?
- **NeedsReinstallPostUpgradeData** After upgrade, this file will have a post-upgrade notification to reinstall the app.
- **NeedsUninstallAction** The file must be uninstalled to complete the upgrade.
- **SdbBlockUpgrade** The file is tagged as blocking upgrade in the SDB,
- **SdbBlockUpgradeCanReinstall** The file is tagged as blocking upgrade in the SDB. It can be reinstalled after upgrade.
- **SdbBlockUpgradeUntilUpdate** The file is tagged as blocking upgrade in the SDB. If the app is updated, the upgrade can proceed.
- **SdbReinstallUpgrade** The file is tagged as needing to be reinstalled after upgrade in the SDB. It does not block upgrade.
- **SdbReinstallUpgradeWarn** The file is tagged as needing to be reinstalled after upgrade with a warning in the SDB. It does not block upgrade.

- **SoftBlock** The file is softblocked in the SDB and has a warning.

Microsoft.Windows.Appraiser.General.DecisionApplicationFileRemove

This event indicates Indicates that the DecisionApplicationFile object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionApplicationFileStartSync

This event indicates that a new set of DecisionApplicationFileAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionDevicePnpAdd

This event sends compatibility decision data about a PNP device to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file generating the events.
- **AssociatedDriverIsBlocked** Is the driver associated with this PNP device blocked?
- **BlockAssociatedDriver** Should the driver associated with this PNP device be blocked?
- **BlockUpgradeIfDriverBlocked** Is the PNP device both boot critical and does not have a driver included with the OS?
- **BlockUpgradeIfDriverBlockedAndOnlyActiveNetwork** Is this PNP device the only active network device?
- **BlockingDevice** Is this PNP device blocking upgrade?
- **DisplayGenericMessage** Will a generic message be shown during Setup for this PNP device?
- **DriverAvailableInbox** Is a driver included with the operating system for this PNP device?
- **DriverAvailableOnline** Is there a driver for this PNP device on Windows Update?
- **DriverAvailableUplevel** Is there a driver on Windows Update or included with the operating system for this PNP device?
- **DriverBlockOverridden** Is there is a driver block on the device that has been overridden?
- **NeedsDismissAction** Will the user would need to dismiss a warning during Setup for this device?
- **NotRegressed** Does the device have a problem code on the source OS that is no better than the one it would have on the target OS?
- **SdbDeviceBlockUpgrade** Is there an SDB block on the PNP device that blocks upgrade?
- **SdbDriverBlockOverridden** Is there an SDB block on the PNP device that blocks upgrade, but that block was overridden?
- **AssociatedDriverWillNotMigrate** Will the driver associated with this plug-and-play device migrate?

Microsoft.Windows.Appraiser.General.DecisionDevicePnpRemove

This event indicates that the DecisionDevicePnp object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionDevicePnpStartSync

This event indicates that the DecisionDevicePnp object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionDriverPackageAdd

This event sends decision data about driver package compatibility to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file generating the events.
- **DriverBlockOverridden** Does the driver package have an SDB block that blocks it from migrating, but that block has been overridden?
- **DriverIsDeviceBlocked** Was the driver package was blocked because of a device block?
- **DriverIsDriverBlocked** Is the driver package blocked because of a driver block?
- **DriverShouldNotMigrate** Should the driver package be migrated during upgrade?
- **SdbDriverBlockOverridden** Does the driver package have an SDB block that blocks it from migrating, but that block has been overridden?

Microsoft.Windows.Appraiser.General.DecisionDriverPackageRemove

This event indicates that the DecisionDriverPackage object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionDriverPackageStartSync

This event indicates that a new set of DecisionDriverPackageAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionMatchingInfoBlockAdd

This event sends compatibility decision data about blocking entries on the system that are not keyed by either applications or devices, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file generating the events.
- **BlockingApplication** Are there are any application issues that interfere with upgrade due to matching info blocks?
- **DisplayGenericMessage** Will a generic message be shown for this block?
- **NeedsUninstallAction** Does the user need to take an action in setup due to a matching info block?
- **SdbBlockUpgrade** Is a matching info block blocking upgrade?
- **SdbBlockUpgradeCanReinstall** Is a matching info block blocking upgrade, but has the can reinstall tag?
- **SdbBlockUpgradeUntilUpdate** Is a matching info block blocking upgrade but has the until update tag?

Microsoft.Windows.Appraiser.General.DecisionMatchingInfoBlockRemove

This event indicates that the DecisionMatchingInfoBlock object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionMatchingInfoBlockStartSync

This event indicates that a new set of DecisionMatchingInfoBlockAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionMatchingInfoPassiveAdd

This event sends compatibility decision data about non-blocking entries on the system that are not keyed by either applications or devices, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **BlockingApplication** Are there any application issues that interfere with upgrade due to matching info blocks?
- **MigApplication** Is there a matching info block with a mig for the current mode of upgrade?

Microsoft.Windows.Appraiser.General.DecisionMatchingInfoPassiveRemove

This event Indicates that the DecisionMatchingInfoPassive object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionMatchingInfoPassiveStartSync

This event indicates that a new set of DecisionMatchingInfoPassiveAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionMatchingInfoPostUpgradeAdd

This event sends compatibility decision data about entries that require reinstall after upgrade. It's used to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **NeedsInstallPostUpgradeData** Will the file have a notification after upgrade to install a replacement for the app?
- **NeedsNotifyPostUpgradeData** Should a notification be shown for this file after upgrade?
- **NeedsReinstallPostUpgradeData** Will the file have a notification after upgrade to reinstall the app?
- **SdbReinstallUpgrade** The file is tagged as needing to be reinstalled after upgrade in the compatibility database (but is not blocking upgrade).

Microsoft.Windows.Appraiser.General.DecisionMatchingInfoPostUpgradeRemove

This event indicates that the DecisionMatchingInfoPostUpgrade object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionMatchingInfoPostUpgradeStartSync

This event indicates that a new set of DecisionMatchingInfoPostUpgradeAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionMediaCenterAdd

This event sends decision data about the presence of Windows Media Center, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file generating the events.

- **BlockingApplication** Is there any application issues that interfere with upgrade due to Windows Media Center?
- **MediaCenterActivelyUsed** If Windows Media Center is supported on the edition, has it been run at least once and are the MediaCenterIndicators are true?
- **MediaCenterInUse** Is Windows Media Center actively being used?
- **MediaCenterIndicators** Do any indicators imply that Windows Media Center is in active use?
- **MediaCenterPaidOrActivelyUsed** Is Windows Media Center actively being used or is it running on a supported edition?
- **NeedsDismissAction** Are there any actions that can be dismissed coming from Windows Media Center?

Microsoft.Windows.Appraiser.General.DecisionMediaCenterRemove

This event indicates that the DecisionMediaCenter object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionMediaCenterStartSync

This event indicates that a new set of DecisionMediaCenterAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionSystemBiosAdd

This event sends compatibility decision data about the BIOS to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file generating the events.
- **Blocking** Is the device blocked from upgrade due to a BIOS block?
- **HasBiosBlock** Does the device have a BIOS block?

Microsoft.Windows.Appraiser.General.DecisionSystemBiosRemove

This event indicates that the DecisionSystemBios object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.DecisionSystemBiosStartSync

This event indicates that a new set of DecisionSystemBiosAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.EnterpriseScenarioWithDiagTrackServiceRunning

The event that indicates that Appraiser has been triggered to run an enterprise scenario while the DiagTrack service is installed. This event can only be sent if a special flag is used to trigger the enterprise scenario.

The following fields are available:

- **Time** The client time of the event.
- **PCFP** An ID for the system calculated by hashing hardware identifiers.

Microsoft.Windows.Appraiser.General.GatedRegChange

This event sends data about the results of running a set of quick-blocking instructions, to help keep Windows up

to date.

The following fields are available:

- **Time** The client time of the event.
- **PCFP** An ID for the system calculated by hashing hardware identifiers.
- **RegKey** The registry key name for which a result is being sent.
- **RegValue** The registry value for which a result is being sent.
- **OldData** The previous data in the registry value before the scan ran.
- **NewData** The data in the registry value after the scan completed.

Microsoft.Windows.Appraiser.General.InventoryApplicationFileAdd

This event represents the basic metadata about a file on the system. The file must be part of an app and either have a block in the compatibility database or are part of an anti-virus program.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file generating the events.
- **BinFileVersion** An attempt to clean up FileVersion at the client that tries to place the version into 4 octets.
- **BinProductVersion** An attempt to clean up ProductVersion at the client that tries to place the version into 4 octets.
- **BinaryType** A binary type. Example: UNINITIALIZED, ZERO_BYTE, DATA_ONLY, DOS_MODULE, NE16_MODULE, PE32_UNKNOWN, PE32_I386, PE32_ARM, PE64_UNKNOWN, PE64_AMD64, PE64_ARM64, PE64_IA64, PE32_CLR_32, PE32_CLR_IL, PE32_CLR_IL_PREFER32, PE64_CLR_64
- **BoeProgramId** If there is no entry in Add/Remove Programs, this is the ProgramID that is generated from the file metadata.
- **CompanyName** The company name of the vendor who developed this file.
- **FileId** A hash that uniquely identifies a file.
- **FileVersion** The File version field from the file metadata under Properties -> Details.
- **LinkDate** The date and time that this file was linked on.
- **LowerCaseLongPath** The full file path to the file that was inventoried on the device.
- **Name** The name of the file that was inventoried.
- **ProductName** The Product name field from the file metadata under Properties -> Details.
- **ProductVersion** The Product version field from the file metadata under Properties -> Details.
- **ProgramId** A hash of the Name, Version, Publisher, and Language of an application used to identify it.
- **Size** The size of the file (in hexadecimal bytes).

Microsoft.Windows.Appraiser.General.InventoryApplicationFileRemove

This event indicates that the InventoryApplicationFile object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.InventoryApplicationFileStartSync

This event indicates indicates that a new set of InventoryApplicationFileAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.InventoryLanguagePackAdd

This event sends data about the number of language packs installed on the system, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **HasLanguagePack** Does this device have 2 or more language packs?
- **LanguagePackCount** How many language packs are installed?

Microsoft.Windows.Appraiser.General.InventoryLanguagePackRemove

This event indicates that the InventoryLanguagePack object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.InventoryLanguagePackStartSync

This event indicates that a new set of InventoryLanguagePackAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.InventoryMediaCenterAdd

This event sends true/false data about decision points used to understand whether Windows Media Center is used on the system, to help keep Windows up to date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file generating the events.
- **EverLaunched** Has Windows Media Center ever been launched?
- **HasConfiguredTv** Has the user configured a TV tuner through Windows Media Center?
- **HasExtendedUserAccounts** Are any Windows Media Center Extender user accounts configured?
- **HasWatchedFolders** Are any folders configured for Windows Media Center to watch?
- **IsDefaultLauncher** Is Windows Media Center the default app for opening music or video files?
- **IsPaid** Is the user running a Windows Media Center edition that implies they paid for Windows Media Center?
- **IsSupported** Does the running OS support Windows Media Center?

Microsoft.Windows.Appraiser.General.InventoryMediaCenterRemove

This event indicates that the InventoryMediaCenter object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.InventoryMediaCenterStartSync

This event indicates that a new set of InventoryMediaCenterAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.InventorySystemBiosAdd

This event sends basic metadata about the BIOS to determine whether it has a compatibility block.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **BiosDate** The release date of the BIOS in UTC format.
- **BiosName** The name field from Win32_BIOS.
- **Manufacturer** The manufacturer field from Win32_ComputerSystem.

- **Model** The model field from Win32_ComputerSystem.

Microsoft.Windows.Appraiser.General.InventorySystemBiosRemove

This event indicates that the InventorySystemBios object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.InventorySystemBiosStartSync

This event indicates that a new set of InventorySystemBiosAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.InventoryUplevelDriverPackageAdd

This event is only runs during setup. It provides a listing of the uplevel driver packages that were downloaded before the upgrade. Is critical to understanding if failures in setup can be traced to not having sufficient uplevel drivers before the upgrade.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **BootCritical** Is the driver package marked as boot critical?
- **Build** The build value from the driver package.
- **CatalogFile** The name of the catalog file within the driver package.
- **ClassGuid** The device class GUID from the driver package.
- **Class** The device class from the driver package.
- **Date** The date from the driver package.
- **SignatureStatus** Indicates if the driver package is signed. Unknown:0, Unsigned:1, Signed: 2
- **Inbox** Is the driver package of a driver that is included with Windows?
- **VersionMajor** The major version of the driver package.
- **VersionMinor** The minor version of the driver package.
- **OriginalName** The original name of the INF file before it was renamed. Generally a path under $WINDOWS.~BT\Drivers\DU
- **Provider** The provider of the driver package.
- **PublishedName** The name of the INF file, post-rename.
- **Revision** The revision of the driver package.

Microsoft.Windows.Appraiser.General.InventoryUplevelDriverPackageRemove

This event indicates that the InventoryUplevelDriverPackage object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.InventoryUplevelDriverPackageStartSync

This event indicates that a new set of InventoryUplevelDriverPackageAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.IsOnlineTelemetryOutputter

This event indicates if Appraiser was able to connect successfully to Windows Update to get driver availability

information.

The following fields are available:

- **Time** The client time of the event.
- **PCFP** A unique hardware identifier that is calculated by hashing hardware identifiers.
- **IsOnlineRun** Was the device able to connect to Windows Update to get driver availability information?

Microsoft.Windows.Appraiser.General.IsOnlineWuDriverDataSource

This event indicates if Appraiser was able to connect to Windows Update to gather driver coverage information.

The following fields are available:

- **Time** The client time of the event.
- **PCFP** A unique hardware identifier that is calculated by hashing hardware identifiers.
- **IsOnlineRun** Was the device able to connect to Windows Update to get driver availability information?
- **TargetVersion** The abbreviated name for the OS version against which Windows Update was queried.

Microsoft.Windows.Appraiser.General.RunContext

This event indicates what should be expected in the data payload.

The following fields are available:

- **AppraiserBranch** The source branch in which the currently running version of Appraiser was built.
- **AppraiserVersion** The version of the Appraiser file generating the events.
- **Context** Indicates what mode Appraiser is running in. Example: Setup or Telemetry.
- **Time** The client time of the event.
- **AppraiserProcess** The name of the process that launched Appraiser.
- **PCFP** An ID for the system calculated by hashing hardware identifiers.

Microsoft.Windows.Appraiser.General.SetupAdlStatus

This event indicates if Appraiser used data files from the setup image or more up-to-date data files downloaded from a Microsoft server.

The following fields are available:

- **Time** The client time of the event.
- **PCFP** An ID for the system calculated by hashing hardware identifiers.
- **Result** The last result of the operation to determine if there is a data file to download.
- **OneSettingsInitialized** Was the query to OneSettings, where the information is stored on if there is a data file to download, initialized?
- **Url** The URL of the data file to download. This will be an empty string if there is no data file to download.
- **UsingAlternateData** Is the client using alternate data file or using the data file in the setup image?

Microsoft.Windows.Appraiser.General.SystemMemoryAdd

This event sends data on the amount of memory on the system and whether it meets requirements, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file generating the events.
- **Blocking** Is the device from upgrade due to memory restrictions?
- **MemoryRequirementViolated** Was a memory requirement violated?
- **pageFile** The current committed memory limit for the system or the current process, whichever is smaller (in bytes).

- **ram** The amount of memory on the device.
- **ramKB** The amount of memory (in KB).
- **virtual** The size of the user-mode portion of the virtual address space of the calling process (in bytes).
- **virtualKB** The amount of virtual memory (in KB).

Microsoft.Windows.Appraiser.General.SystemMemoryRemove

This event that the SystemMemory object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemMemoryStartSync

This event indicates that a new set of SystemMemoryAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemProcessorCompareExchangeAdd

This event sends data indicating whether the system supports the CompareExchange128 CPU requirement, to help keep Windows up to date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file generating the events.
- **Blocking** Is the upgrade blocked due to the processor?
- **CompareExchange128Support** Does the CPU support CompareExchange128?

Microsoft.Windows.Appraiser.General.SystemProcessorCompareExchangeRemove

This event indicates that the SystemProcessorCompareExchange object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemProcessorCompareExchangeStartSync

This event indicates that a new set of SystemProcessorCompareExchangeAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemProcessorLahfSahfAdd

This event sends data indicating whether the system supports the LahfSahf CPU requirement, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file generating the events.
- **Blocking** Is the upgrade blocked due to the processor?
- **LahfSahfSupport** Does the CPU support LAHF/SAHF?

Microsoft.Windows.Appraiser.General.SystemProcessorLahfSahfRemove

This event indicates that the SystemProcessorLahfSahf object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemProcessorLahfSahfStartSync

This event indicates that a new set of SystemProcessorLahfSahfAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemProcessorNxAdd

This event sends data indicating whether the system supports the NX CPU requirement, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **Blocking** Is the upgrade blocked due to the processor?
- **NXDriverResult** The result of the driver used to do a non-deterministic check for NX support.
- **NXProcessorSupport** Does the processor support NX?

Microsoft.Windows.Appraiser.General.SystemProcessorNxRemove

This event indicates that the SystemProcessorNx object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemProcessorNxStartSync

This event indicates that a new set of SystemProcessorNxAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemProcessorPrefetchWAdd

This event sends data indicating whether the system supports the PrefetchW CPU requirement, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **Blocking** Is the upgrade blocked due to the processor?
- **PrefetchWSupport** Does the processor support PrefetchW?

Microsoft.Windows.Appraiser.General.SystemProcessorPrefetchWRemove

This event indicates that the SystemProcessorPrefetchW object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemProcessorPrefetchWStartSync

This event indicates that a new set of SystemProcessorPrefetchWAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemProcessorSse2Add

This event sends data indicating whether the system supports the SSE2 CPU requirement, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **Blocking** Is the upgrade blocked due to the processor?
- **SSE2ProcessorSupport** Does the processor support SSE2?

Microsoft.Windows.Appraiser.General.SystemProcessorSse2Remove

This event indicates that the SystemProcessorSse2 object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemProcessorSse2StartSync

This event indicates that a new set of SystemProcessorSse2Add events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemTouchAdd

This event sends data indicating whether the system supports touch, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **IntegratedTouchDigitizerPresent** Is there an integrated touch digitizer?
- **MaximumTouches** The maximum number of touch points supported by the device hardware.

Microsoft.Windows.Appraiser.General.SystemTouchRemove

This event indicates that the SystemTouch object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemTouchStartSync

This event indicates that a new set of SystemTouchAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemWimAdd

This event sends data indicating whether the operating system is running from a compressed WIM file, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **IsWimBoot** Is the current operating system running from a compressed WIM file?
- **RegistryWimBootValue** The raw value from the registry that is used to indicate if the device is running from a WIM.

Microsoft.Windows.Appraiser.General.SystemWimRemove

This event indicates that the SystemWim object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemWimStartSync

This event indicates that a new set of SystemWimAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemWindowsActivationStatusAdd

This event sends data indicating whether the current operating system is activated, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **WindowsIsLicensedApiValue** The result from the API that's used to indicate if operating system is activated.
- **WindowsNotActivatedDecision** Is the current operating system activated?

Microsoft.Windows.Appraiser.General.SystemWindowsActivationStatusRemove

This event indicates that the SystemWindowsActivationStatus object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemWindowsActivationStatusStartSync

This event indicates that a new set of SystemWindowsActivationStatusAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemWlanAdd

This event sends data indicating whether the system has WLAN, and if so, whether it uses an emulated driver that could block an upgrade, to help keep Windows up-to-date.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **Blocking** Is the upgrade blocked because of an emulated WLAN driver?
- **HasWlanBlock** Does the emulated WLAN driver have an upgrade block?
- **WlanEmulatedDriver** Does the device have an emulated WLAN driver?
- **WlanExists** Does the device support WLAN at all?
- **WlanModulePresent** Are any WLAN modules present?
- **WlanNativeDriver** Does the device have a non-emulated WLAN driver?

Microsoft.Windows.Appraiser.General.SystemWlanRemove

This event indicates that the SystemWlan object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.SystemWlanStartSync

This event indicates that a new set of SystemWlanAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.TelemetryRunHealth

A summary event indicating the parameters and result of a telemetry run. This allows the rest of the data sent over the course of the run to be properly contextualized and understood, which is then used to keep Windows up-to-date.

The following fields are available:

- **PerfBackoff** Indicates if the run was invoked with logic to stop running when a user is present. Helps to understand why a run may have a longer elapsed time than normal.
- **RunAppraiser** Indicates if Appraiser was set to run at all. If this if false, it is understood that data events will not be received from this device.
- **ThrottlingUtc** Indicates if the Appraiser client is throttling its output of CUET events to avoid being disabled. This increases runtime but also telemetry reliability.
- **AuxInitial** Obsolete, indicates if Appraiser is writing data files to be read by the Get Windows 10 app.
- **Time** The client time of the event.
- **RunDate** The date that the telemetry run was stated, expressed as a filetime.
- **AppraiserProcess** The name of the process that launched Appraiser.
- **AppraiserVersion** The file version (major, minor and build) of the Appraiser DLL, concatenated without dots.
- **SendingUtc** Indicates if the Appraiser client is sending events during the current telemetry run.
- **DeadlineDate** A timestamp representing the deadline date, which is the time until which appraiser will wait to do a full scan.
- **AppraiserBranch** The source branch in which the version of Appraiser that is running was built.
- **EnterpriseRun** Indicates if the telemetry run is an enterprise run, which means appraiser was run from the command line with an extra enterprise parameter.
- **RunGeneralTel** Indicates if the generaltel.dll component was run. Generaltel collects additional telemetry on an infrequent schedule and only from machines at telemetry levels higher than Basic.
- **PerfBackoffInsurance** Indicates if appraiser is running without performance backoff because it has run with perf backoff and failed to complete several times in a row.
- **AuxFinal** Obsolete, always set to false
- **StoreHandleIsNotNull** Obsolete, always set to false
- **VerboseMode** Indicates if appraiser ran in Verbose mode, which is a test-only mode with extra logging.
- **AppraiserDataVersion** The version of the data files being used by the Appraiser telemetry run.
- **FullSync** Indicates if Appraiser is performing a full sync, which means that full set of events representing the state of the machine are sent. Otherwise, only the changes from the previous run are sent.
- **InventoryFullSync** Indicates if inventory is performing a full sync, which means that the full set of events representing the inventory of machine are sent.
- **PCFP** An ID for the system calculated by hashing hardware identifiers.
- **RunOnline** Indicates if appraiser was able to connect to Windows Update and theefore is making decisions using up-to-date driver coverage information.
- **TelementrySent** Indicates if telemetry was successfully sent.
- **WhyFullSyncWithoutTablePrefix** Indicates the reason or reasons that a full sync was generated.
- **RunResult** The hresult of the Appraiser telemetry run.

Microsoft.Windows.Appraiser.General.WmdrmAdd

This event sends data about the usage of older digital rights management on the system, to help keep Windows up to date. This data does not indicate the details of the media using the digital rights management, only whether any such files exist. Collecting this data was critical to ensuring the correct mitigation for customers, and should be able to be removed once all mitigations are in place.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **WmdrmCdRipped** Indicates if the system has any files encrypted with personal DRM, which was used for ripped CDs.
- **WmdrmNonPermanent** Indicates if the system has any files with non-permanent licenses.
- **WmdrmPurchased** Indicates if the system has any files with permanent licenses.
- **WmdrmApiResult** Raw value of the API used to gather DRM state.
- **WmdrmInUse** WmdrmIndicators AND dismissible block in setup was not dismissed.
- **WmdrmIndicators** WmdrmCdRipped OR WmdrmPurchased
- **NeedsDismissAction** Indicates if a dismissible message is needed to warn the user about a potential loss of data due to DRM deprecation.
- **BlockingApplication** Same as NeedsDismissAction

Microsoft.Windows.Appraiser.General.WmdrmRemove

This event indicates that the Wmdrm object is no longer present.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Microsoft.Windows.Appraiser.General.WmdrmStartSync

This event indicates that a new set of WmdrmAdd events will be sent.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.

Census events

Census.App

This event sends version data about the Apps running on this device, to help keep Windows up to date.

The following fields are available:

- **IEVersion** Retrieves which version of Internet Explorer is running on this device.
- **CensusVersion** The version of Census that generated the current data for this device.

Census.Battery

This event sends type and capacity data about the battery on the device, as well as the number of connected standby devices in use, type to help keep Windows up to date.

The following fields are available:

- **InternalBatteryCapablities** Represents information about what the battery is capable of doing.
- **InternalBatteryCapacityCurrent** Represents the battery's current fully charged capacity in mWh (or relative). Compare this value to DesignedCapacity to estimate the battery's wear.
- **InternalBatteryCapacityDesign** Represents the theoretical capacity of the battery when new, in mWh.
- **IsAlwaysOnAlwaysConnectedCapable** Represents whether the battery enables the device to be AlwaysOnAlwaysConnected . Boolean value.
- **InternalBatteryNumberOfCharges** Provides the number of battery charges. This is used when creating new products and validating that existing products meets targeted functionality performance.

Census.Camera

This event sends data about the resolution of cameras on the device, to help keep Windows up to date.

The following fields are available:

- **FrontFacingCameraResolution** Represents the resolution of the front facing camera in megapixels. If a front facing camera does not exist, then the value is 0.
- **RearFacingCameraResolution** Represents the resolution of the rear facing camera in megapixels. If a rear facing camera does not exist, then the value is 0.

Census.Enterprise

This event sends data about Azure presence, type, and cloud domain use in order to provide an understanding of the use and integration of devices in an enterprise, cloud, and server environment.

The following fields are available:

- **IsCloudDomainJoined** Is this device joined to an Azure Active Directory (AAD) tenant? true/false
- **IsMDMEnrolled** Whether the device has been MDM Enrolled or not.
- **ServerFeatures** Represents the features installed on a Windows Server. This can be used by developers and administrators who need to automate the process of determining the features installed on a set of server computers.
- **CommercialId** Represents the GUID for the commercial entity which the device is a member of. Will be used to reflect insights back to customers.
- **AzureVMType** Represents whether the instance is Azure VM PAAS, Azure VM IAAS or any other VMs.
- **AzureOSIDPresent** Represents the field used to identify an Azure machine.
- **IsDomainJoined** Indicates whether a machine is joined to a domain.
- **HashedDomain** The hashed representation of the user domain used for login.
- **SystemCenterID** The SCCM ID is an anonymized one-way hash of the Active Directory Organization identifier
- **MPNId** Returns the Partner ID/MPN ID from Regkey. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\DeployID
- **SCCMClientId** This ID correlate systems that send data to Compat Analytics (OMS) and other OMS based systems with systems in an Enterprise SCCM environment.
- **CDJType** Represents the type of cloud domain joined for the machine.
- **IsDeviceProtected** Represents if Device protected by BitLocker/Device Encryption
- **IsDERequirementMet** Represents if the device can do device encryption.
- **IsEDPEnabled** Represents if Enterprise data protected on the device.
- **ContainerType** The type of container, such as process or virtual machine hosted.
- **EnrollmentType** Represents the type of enrollment, such as MDM or Intune, for a particular device.

Census.Firmware

This event sends data about the BIOS and startup embedded in the device, to help keep Windows up to date.

The following fields are available:

- **FirmwareManufacturer** Represents the manufacturer of the device's firmware (BIOS).
- **FirmwareReleaseDate** Represents the date the current firmware was released.
- **FirmwareType** Represents the firmware type. The various types can be unknown, BIOS, UEFI.
- **FirmwareVersion** Represents the version of the current firmware.

Census.Flighting

This event sends Windows Insider data from customers participating in improvement testing and feedback programs, to help keep Windows up-to-date.

The following fields are available:

- **FlightIds** A list of the different Windows Insider builds on this device.

- **MSA_Accounts** Represents a list of hashed IDs of the Microsoft Accounts that are flighting (pre-release builds) on this device.
- **IsFlightsDisabled** Represents if the device is participating in the Windows Insider program.
- **FlightingBranchName** The name of the Windows Insider branch currently used by the device.
- **DeviceSampleRate** The telemetry sample rate assigned to the device.
- **EnablePreviewBuilds** Used to enable Windows Insider builds on a device.
- **SSRK** Retrieves the mobile targeting settings.

Census.Hardware

This event sends data about the device, including hardware type, OEM brand, model line, model, telemetry level setting, and TPM support, to help keep Windows up-to-date.

The following fields are available:

- **ChassisType** Represents the type of device chassis, such as desktop or low profile desktop. The possible values can range between 1 - 36.
- **ComputerHardwareID** Identifies a device class that is represented by a hash of different SMBIOS fields.
- **DeviceColor** Indicates a color of the device.
- **DeviceName** The device name that is set by the user.
- **OEMDigitalMarkerFileName** The name of the file placed in the \Windows\system32\drivers directory that specifies the OEM and model name of the device.
- **OEMManufacturerName** The device manufacturer name. The OEMName for an inactive device is not reprocessed even if the clean OEM name is changed at a later date.
- **OEMModelNumber** The device model number.
- **OEMModelName** The device model name.
- **OEMModelSKU** The device edition that is defined by the manufacturer.
- **OEMOptionalIdentifier** A Microsoft assigned value that represents a specific OEM subsidiary.
- **OEMSerialNumber** The serial number of the device that is set by the manufacturer.
- **PhoneManufacturer** The friendly name of the phone manufacturer.
- **SoCName** The firmware manufacturer of the device.
- **DUID** The device unique ID.
- **InventoryId** The device ID used for compatibility testing.
- **VoiceSupported** Does the device have a cellular radio capable of making voice calls?
- **PowerPlatformRole** The OEM preferred power management profile. It's used to help to identify the basic form factor of the device.
- **TPMVersion** The supported Trusted Platform Module (TPM) on the device. If no TPM is present, the value is 0.
- **StudyID** Used to identify retail and non-retail device.
- **TelemetryLevel** The telemetry level the user has opted into, such as Basic or Enhanced.
- **TelemetrySettingAuthority** Determines who set the telemetry level, such as GP, MDM, or the user.
- **DeviceForm** Indicates the form as per the device classification.
- **DigitizerSupport** Is a digitizer supported?
- **OEMModelBaseBoard** The baseboard model used by the OEM.
- **OEMModelSystemFamily** The system family set on the device by an OEM.
- **OEMModelBaseBoardVersion** Differentiates between developer and retail devices.
- **ActiveMicCount** The number of active microphones attached to the device.
- **OEMModelSystemVersion** The system model version set on the device by the OEM.
- **D3DMaxFeatureLevel** The supported Direct3D version.
- **Gyroscope** Indicates whether the device has a gyroscope.
- **Magnetometer** Indicates whether the device has a magnetometer.

- **NFCProximity** Indicates whether the device supports NFC.
- **TelemetryLevelLimitEnhanced** The telemetry level for Windows Analytics-based solutions.

Census.Memory

This event sends data about the memory on the device, including ROM and RAM, to help keep Windows up to date.

The following fields are available:

- **TotalPhysicalRAM** Represents the physical memory (in MB).
- **TotalVisibleMemory** Represents the memory that is not reserved by the system.

Census.Network

This event sends data about the mobile and cellular network used by the device (mobile service provider, network, device ID, and service cost factors), to help keep Windows up to date.

The following fields are available:

- **MobileOperatorBilling** Represents the telephone company that provides services for mobile phone users.
- **MobileOperatorCommercialized** Represents which reseller and geography the phone is commercialized for. This is the set of values on the phone for who and where it was intended to be used. For example, the commercialized mobile operator code AT&T in the US would be ATT-US.
- **NetworkCost** Represents the network cost associated with a connection.
- **IMEI0** Represents the International Mobile Station Equipment Identity. This number is usually unique and used by the mobile operator to distinguish different phone hardware. Microsoft does not have access to mobile operator billing data so collecting this data does not expose or identify the user. The two fields represent phone with dual sim coverage.
- **SPN0** Retrieves the Service Provider Name (SPN). For example, these might be AT&T, Sprint, T-Mobile, or Verizon. The two fields represent phone with dual sim coverage.
- **MobileOperatorNetwork0** Represents the operator of the current mobile network that the device is used on. (AT&T, T-Mobile, Vodafone). The two fields represent phone with dual sim coverage.
- **MCC0** Represents the Mobile Country Code (MCC). It used with the Mobile Network Code (MNC) to uniquely identify a mobile network operator. The two fields represent phone with dual sim coverage.
- **MNC0** Retrieves the Mobile Network Code (MNC). It used with the Mobile Country Code (MCC) to uniquely identify a mobile network operator. The two fields represent phone with dual sim coverage.
- **IMEI1** Represents the International Mobile Station Equipment Identity. This number is usually unique and used by the mobile operator to distinguish different phone hardware. Microsoft does not have access to mobile operator billing data so collecting this data does not expose or identify the user. The two fields represent phone with dual sim coverage.
- **SPN1** Retrieves the Service Provider Name (SPN). For example, these might be AT&T, Sprint, T-Mobile, or Verizon. The two fields represent phone with dual sim coverage.
- **MobileOperatorNetwork1** Represents the operator of the current mobile network that the device is used on. (AT&T, T-Mobile, Vodafone). The two fields represent phone with dual sim coverage.
- **MCC1** Represents the Mobile Country Code (MCC). It used with the Mobile Network Code (MNC) to uniquely identify a mobile network operator. The two fields represent phone with dual sim coverage.
- **MNC1** Retrieves the Mobile Network Code (MNC). It used with the Mobile Country Code (MCC) to uniquely identify a mobile network operator. The two fields represent phone with dual sim coverage.
- **MEID** Represents the Mobile Equipment Identity (MEID). MEID is a worldwide unique phone ID assigned to CDMA phones. MEID replaces electronic serial number (ESN), and is equivalent to IMEI for GSM and WCDMA phones. Microsoft does not have access to mobile operator billing data so collecting this data does not expose or identify the user.
- **NetworkAdapterGUID** The GUID of the primary network adapter.

Census.OS

This event sends data about the operating system such as the version, locale, update service configuration, when and how it was originally installed, and whether it is a virtual device, to help keep Windows up to date.

The following fields are available:

- **GenuineState** Retrieves the ID Value specifying the OS Genuine check.
- **IsPortableOperatingSystem** Retrieves whether OS is running Windows-To-Go
- **IsSecureBootEnabled** Retrieves whether Boot chain is signed under UEFI.
- **InstallationType** Retrieves the type of OS installation. (Clean, Upgrade, Reset, Refresh, Update).
- **OSInstallType** Retrieves a numeric description of what install was used on the device i.e. clean, upgrade, refresh, reset, etc
- **OSOOBEDateTime** Retrieves Out of Box Experience (OOBE) Date in Coordinated Universal Time (UTC).
- **OSSKU** Retrieves the Friendly Name of OS Edition.
- **OSTimeZoneBiasInMins** Retrieves the time zone set on machine.
- **OSUILocale** Retrieves the locale of the UI that is currently used by the OS.
- **RACw7Id** Retrieves the Microsoft Reliability Analysis Component (RAC) Win7 Identifier. RAC is used to monitor and analyze system usage and reliability.
- **CompactOS** Indicates if the Compact OS feature from Win10 is enabled.
- **Signature** Retrieves if it is a signature machine sold by Microsoft store.
- **IsDeviceRetailDemo** Retrieves if the device is running in demo mode.
- **ActivationChannel** Retrieves the retail license key or Volume license key for a machine.
- **LicenseStateReason** Retrieves why (or how) a system is licensed or unlicensed. The HRESULT may indicate an error code that indicates a key blocked error, or it may indicate that we are running an OS License granted by the MS store.
- **OA3xOriginalProductKey** Retrieves the License key stamped by the OEM to the machine.
- **ProductKeyID2** Retrieves the License key if the machine is updated with a new license key.
- **ServiceMachineIP** Retrieves the IP address of the KMS host used for anti-piracy.
- **ServiceProductKeyID** Retrieves the License key of the KMS
- **LanguagePacks** The list of language packages installed on the device.
- **InstallLanguage** The first language installed on the user machine.
- **IsEduData** Returns Boolean if the education data policy is enabled.
- **SharedPCMode** Returns Boolean for education devices used as shared cart
- **SLICVersion** Returns OS type/version from SLIC table.
- **SLICStatus** Whether a SLIC table exists on the device.
- **OSEdition** Retrieves the version of the current OS.
- **ProductActivationTime** Returns the OS Activation time for tracking piracy issues.
- **ProductActivationResult** Returns Boolean if the OS Activation was successful.
- **OSSubscriptionTypeId** Returns boolean for enterprise subscription feature for selected PRO machines.
- **OSSubscriptionStatus** Represents the existing status for enterprise subscription feature for PRO machines.
- **ServiceMachinePort** Retrieves the port of the KMS host used for anti-piracy.
- **DeviceTimeZone** The time zone that is set on the device. Example: Pacific Standard Time
- **DeveloperUnlockStatus** Represents if a device has been developer unlocked by the user or Group Policy.
- **AssignedAccessStatus** The kiosk configuration mode.

Census.Processor

This event sends data about the processor (architecture, speed, number of cores, manufacturer, and model number), to help keep Windows up to date.

The following fields are available:

- **ProcessorCores** Retrieves the number of cores in the processor.
- **ProcessorPhysicalCores** Number of physical cores in the processor.
- **ProcessorArchitecture** Retrieves the processor architecture of the installed operating system. The complete list of values can be found in DimProcessorArchitecture.
- **ProcessorClockSpeed** Retrieves the clock speed of the processor in MHz.
- **ProcessorManufacturer** Retrieves the name of the processor's manufacturer.
- **ProcessorModel** Retrieves the name of the processor model.
- **SocketCount** Number of physical CPU sockets of the machine.
- **ProcessorIdentifier** The processor identifier of a manufacturer.
- **ProcessorUpdateRevision** The microcode version.

Census.Speech

This event is used to gather basic speech settings on the device.

The following fields are available:

- **AboveLockEnabled** Cortana setting that represents if Cortana can be invoked when the device is locked.
- **GPAllowInputPersonalization** Indicates if a Group Policy setting has enabled speech functionalities.
- **HolographicSpeechInputDisabled** Holographic setting that represents if the attached HMD devices have speech functionality disabled by the user.
- **HolographicSpeechInputDisabledRemote** Indicates if a remote policy has disabled speech functionalities for the HMD devices.
- **KWSEnabled** Cortana setting that represents if a user has enabled the "Hey Cortana" keyword spotter (KWS).
- **MDMAllowInputPersonalization** Indicates if an MDM policy has enabled speech functionalities.
- **RemotelyManaged** Indicates if the device is being controlled by a remote administrator (MDM or Group Policy) in the context of speech functionalities.
- **SpeakerIdEnabled** Cortana setting that represents if keyword detection has been trained to try to respond to a single user's voice.
- **SpeechServicesEnabled** Windows setting that represents whether a user is opted-in for speech services on the device.

Census.Storage

This event sends data about the total capacity of the system volume and primary disk, to help keep Windows up to date.

The following fields are available:

- **PrimaryDiskTotalCapacity** Retrieves the amount of disk space on the primary disk of the device in MB.
- **SystemVolumeTotalCapacity** Retrieves the size of the partition that the System volume is installed on in MB.
- **PrimaryDiskType** Retrieves an enumerator value of type STORAGE_BUS_TYPE that indicates the type of bus to which the device is connected. This should be used to interpret the raw device properties at the end of this structure (if any).

Census.Userdefault

This event sends data about the current user's default preferences for browser and several of the most popular extensions and protocols, to help keep Windows up to date.

The following fields are available:

- **DefaultBrowserProgId** The ProgramId of the current user's default browser
- **DefaultApp** The current uer's default program selected for the following extension or protocol:

.html,.htm,.jpg,.jpeg,.png,.mp3,.mp4, .mov,.pdf

Census.UserDisplay

This event sends data about the logical/physical display size, resolution and number of internal/external displays, and VRAM on the system, to help keep Windows up to date.

The following fields are available:

- **InternalPrimaryDisplayLogicalDPIX** Retrieves the logical DPI in the x-direction of the internal display.
- **InternalPrimaryDisplayLogicalDPIY** Retrieves the logical DPI in the y-direction of the internal display.
- **InternalPrimaryDisplayPhysicalDPIX** Retrieves the physical DPI in the x-direction of the internal display.
- **InternalPrimaryDisplayPhysicalDPIY** Retrieves the physical DPI in the y-direction of the internal display.
- **InternalPrimaryDisplayResolutionHorizontal** Retrieves the number of pixels in the horizontal direction of the internal display.
- **InternalPrimaryDisplayResolutionVertical** Retrieves the number of pixels in the vertical direction of the internal display.
- **InternalPrimaryDisplaySizePhysicalH** Retrieves the physical horizontal length of the display in mm. Used for calculating the diagonal length in inches .
- **InternalPrimaryDisplaySizePhysicalY** Retrieves the physical vertical length of the display in mm. Used for calculating the diagonal length in inches
- **NumberofInternalDisplays** Retrieves the number of internal displays in a machine.
- **NumberofExternalDisplays** Retrieves the number of external displays connected to the machine
- **VRAMDedicated** Retrieves the video RAM in MB.
- **VRAMDedicatedSystem** Retrieves the amount of memory on the dedicated video card.
- **VRAMSharedSystem** Retrieves the amount of RAM memory that the video card can use.

Census.UserNLS

This event sends data about the default app language, input, and display language preferences set by the user, to help keep Windows up to date.

The following fields are available:

- **DefaultAppLanguage** The current user Default App Language.
- **HomeLocation** The current user location, which is populated using GetUserGeoId() function.
- **DisplayLanguage** The current user preferred Windows Display Language.
- **SpeechInputLanguages** The Speech Input languages installed on the device.
- **KeyboardInputLanguages** The Keyboard input languages installed on the device.

Census.VM

This event sends data indicating whether virtualization is enabled on the device, and its various characteristics, to help keep Windows up to date.

The following fields are available:

- **VirtualizationFirmwareEnabled** Represents whether virtualization is enabled in the firmware.
- **SLATSupported** Represents whether Second Level Address Translation (SLAT) is supported by the hardware.
- **IOMMUPresent** Represents if an input/output memory management unit (IOMMU) is present.
- **IsVirtualDevice** Retrieves that when the Hypervisor is Microsoft's Hyper-V Hypervisor or other Hv#1 Hypervisor, this field will be set to FALSE for the Hyper-V host OS and TRUE for any guest OS's. This field should not be relied upon for non-Hv#1 Hypervisors.
- **HyperVisor** Retrieves whether the current OS is running on top of a Hypervisor.
- **CloudService** Indicates which cloud service, if any, that this virtual machine is running within.
- **isVDI** Is the device using Virtual Desktop Infrastructure?

Census.WU

This event sends data about the Windows update server and other App store policies, to help keep Windows up to date.

The following fields are available:

- **WUMachineId** Retrieves the Windows Update (WU) Machine Identifier.
- **WUServer** Retrieves the HTTP(S) URL of the WSUS server that is used by Automatic Updates and API callers (by default).
- **WUDODownloadMode** Retrieves whether DO is turned on and how to acquire/distribute updates Delivery Optimization (DO) allows users to deploy previously downloaded WU updates to other devices on the same network.
- **OSWUAutoUpdateOptions** Retrieves the auto update settings on the device.
- **AppStoreAutoUpdate** Retrieves the Appstore settings for auto upgrade. (Enable/Disabled).
- **AppStoreAutoUpdatePolicy** Retrieves the Microsoft Store App Auto Update group policy setting
- **AppStoreAutoUpdateMDM** Retrieves the App Auto Update value for MDM: 0 - Disallowed. 1 - Allowed. 2 - Not configured. Default: [2] Not configured
- **DelayUpgrade** Retrieves the Windows upgrade flag for delaying upgrades.
- **UpdateServiceURLConfigured** Retrieves if the device is managed by Windows Server Update Services (WSUS).
- **WUDeferUpgradePeriod** Retrieves if deferral is set for Upgrades
- **WUDeferUpdatePeriod** Retrieves if deferral is set for Updates
- **WUPauseState** Retrieves WU setting to determine if updates are paused
- **OSUninstalled** A flag that represents when a feature update is uninstalled on a device .
- **OSRolledBack** A flag that represents when a feature update has rolled back during setup.
- **OSRollbackCount** The number of times feature updates have rolled back on the device.
- **UninstallActive** A flag that represents when a device has uninstalled a previous upgrade recently.
- **AppraiserGatedStatus** Indicates whether a device has been gated for upgrading.
- **OSAssessmentFeatureOutOfDate** How many days has it been since a the last feature update was released but the device did not install it?
- **OSAssessmentForFeatureUpdate** Is the device is on the latest feature update?
- **OSAssessmentForQualityUpdate** Is the device on the latest quality update?
- **OSAssessmentForSecurityUpdate** Is the device on the latest security update?
- **OSAssessmentQualityOutOfDate** How many days has it been since a the last quality update was released but the device did not install it?
- **OSAssessmentReleaseInfoTime** The freshness of release information used to perform an assessment.

Census.Xbox

This event sends data about the Xbox Console, such as Serial Number and DeviceId, to help keep Windows up to date.

The following fields are available:

- **XboxLiveDeviceId** Retrieves the unique device id of the console.
- **XboxConsoleSerialNumber** Retrieves the serial number of the Xbox console.
- **XboxLiveSandboxId** Retrieves the developer sandbox id if the device is internal to MS.
- **XboxConsolePreferredLanguage** Retrieves the preferred language selected by the user on Xbox console.

Census.Security

This event provides information on about security settings used to help keep Windows up-to-date and secure.

- **AvailableSecurityProperties** Enumerates and reports state on the relevant security properties for Device Guard.
- **CGRunning** Is Credential Guard running?
- **DGState** A summary of the Device Guard state.
- **HVCIRunning** Is HVCI running?
- **RequiredSecurityProperties** Describes the required security properties to enable virtualization-based security.
- **SecureBootCapable** Is this device capable of running Secure Boot?
- **VBSState** Is virtualization-based security enabled, disabled, or running?

Diagnostic data events

TelClientSynthetic.AuthorizationInfo_RuntimeTransition

This event sends data indicating that a device has undergone a change of telemetry opt-in level during the runtime of the device (not at UTC boot or offline), to help keep Windows up to date.

The following fields are available:

- **CanAddMsaToMsTelemetry** True if UTC is allowed to add MSA user identity onto telemetry from the OS provider groups.
- **CanCollectAnyTelemetry** True if UTC is allowed to collect non-OS telemetry. Non-OS telemetry is responsible for providing its own opt-in mechanism.
- **CanCollectCoreTelemetry** True if UTC is allowed to collect data which is tagged with both MICROSOFT_KEYWORD_CRITICAL_DATA and MICROSOFT_EVENTTAG_CORE_DATA.
- **CanCollectHeartbeats** True if UTC is allowed to collect heartbeats.
- **CanCollectOsTelemetry** True if UTC is allowed to collect telemetry from the OS provider groups (often called Microsoft Telemetry).
- **CanPerformDiagnosticEscalations** True if UTC is allowed to perform all scenario escalations.
- **CanPerformScripting** True if UTC is allowed to perform scripting.
- **CanPerformTraceEscalations** True if UTC is allowed to perform scenario escalations with tracing actions.
- **CanReportScenarios** True if UTC is allowed to load and report scenario completion, failure, and cancellation events.
- **TransitionFromEverythingOff** True if this transition is moving from not allowing core telemetry to allowing core telemetry.
- **PreviousPermissions** Bitmask representing the previously configured permissions since the telemetry opt-in level was last changed.

TelClientSynthetic.AuthorizationInfo_Startup

This event sends data indicating that a device has undergone a change of telemetry opt-in level detected at UTC startup, to help keep Windows up to date.

The following fields are available:

- **TransitionFromEverythingOff** True if this transition is moving from not allowing core telemetry to allowing core telemetry.
- **CanCollectAnyTelemetry** True if UTC is allowed to collect non-OS telemetry. Non-OS telemetry is responsible for providing its own opt-in mechanism.
- **CanCollectHeartbeats** True if UTC is allowed to collect heartbeats.
- **CanCollectCoreTelemetry** True if UTC is allowed to collect data which is tagged with both MICROSOFT_KEYWORD_CRITICAL_DATA and MICROSOFT_EVENTTAG_CORE_DATA.
- **CanCollectOsTelemetry** True if UTC is allowed to collect telemetry from the OS provider groups (often called

Microsoft Telemetry).

- **CanReportScenarios** True if UTC is allowed to load and report scenario completion, failure, and cancellation events.
- **CanAddMsaToMsTelemetry** True if UTC is allowed to add MSA user identity onto telemetry from the OS provider groups.
- **CanPerformTraceEscalations** True if UTC is allowed to perform scenario escalations with tracing actions.
- **CanPerformDiagnosticEscalations** True if UTC is allowed to perform all scenario escalations.
- **CanPerformScripting** True if UTC is allowed to perform scripting.
- **PreviousPermissions** Bitmask representing the previously configured permissions since the telemetry client was last started.

TelClientSynthetic.ConnectivityHeartBeat_0

This event sends data about the connectivity status of the Connected User Experience and Telemetry component that uploads telemetry events. If an unrestricted free network (such as Wi-Fi) is available, this event updates the last successful upload time. Otherwise, it checks whether a Connectivity Heartbeat event was fired in the past 24 hours, and if not, it fires an event. A Connectivity Heartbeat event also fires when a device recovers from costed network to free network.

The following fields are available:

- **CensusExitCode** Returns last execution codes from census client run.
- **CensusStartTime** Returns timestamp corresponding to last successful census run.
- **CensusTaskEnabled** Returns Boolean value for the census task (Enable/Disable) on client machine.
- **LastConnectivityLossTime** Retrieves the last time the device lost free network.
- **NetworkState** Retrieves the network state: 0 = No network. 1 = Restricted network. 2 = Free network.
- **NoNetworkTime** Retrieves the time spent with no network (since the last time) in seconds.
- **RestrictedNetworkTime** Retrieves the time spent on a metered (cost restricted) network in seconds.
- **LastConntectivityLossTime** Retrieves the last time the device lost free network.

TelClientSynthetic.HeartBeat_5

This event sends data about the health and quality of the telemetry data from the given device, to help keep Windows up to date. It also enables data analysts to determine how 'trusted' the data is from a given device.

The following fields are available:

- **PreviousHeartBeatTime** The time of last heartbeat event. This allows chaining of events.
- **EtwDroppedCount** The number of events dropped by the ETW layer of the telemetry client.
- **ConsumerDroppedCount** The number of events dropped by the consumer layer of the telemetry client.
- **DecodingDroppedCount** The number of events dropped because of decoding failures.
- **ThrottledDroppedCount** The number of events dropped due to throttling of noisy providers.
- **DbDroppedCount** The number of events that were dropped because the database was full.
- **EventSubStoreResetCounter** The number of times the event database was reset.
- **EventSubStoreResetSizeSum** The total size of the event database across all resets reports in this instance.
- **CriticalOverflowEntersCounter** The number of times a critical overflow mode was entered into the event database.
- **EnteringCriticalOverflowDroppedCounter** The number of events that was dropped because a critical overflow mode was initiated.
- **UploaderDroppedCount** The number of events dropped by the uploader layer of the telemetry client.
- **InvalidHttpCodeCount** The number of invalid HTTP codes received from Vortex.
- **LastInvalidHttpCode** The last invalid HTTP code received from Vortex.
- **MaxInUseScenarioCounter** The soft maximum number of scenarios loaded by the Connected User

Experience and Telemetry component.

- **LastEventSizeOffender** The name of the last event that exceeded the maximum event size.
- **SettingsHttpAttempts** The number of attempts to contact the OneSettings service.
- **SettingsHttpFailures** The number of failures from contacting the OneSettings service.
- **VortexHttpAttempts** The number of attempts to contact the Vortex service.
- **EventsUploaded** The number of events that have been uploaded.
- **DbCriticalDroppedCount** The total number of dropped critical events in the event database.
- **VortexHttpFailures4xx** The number of 400-499 error codes received from Vortex.
- **VortexHttpFailures5xx** The number of 500-599 error codes received from Vortex.
- **VortexFailuresTimeout** The number of timeout failures received from Vortex.
- **HeartBeatSequenceNumber** A monotonically increasing heartbeat counter.
- **EtwDroppedBufferCount** The number of buffers dropped in the CUET ETW session.
- **FullTriggerBufferDroppedCount** The number of events that were dropped because the trigger buffer was full.
- **CriticalDataThrottleDroppedCount** The number of critical data sampled events that were dropped because of throttling.
- **CriticalDataDbDroppedCount** The number of critical data sampled events that were dropped at the database layer.
- **MaxActiveAgentConnectionCount** The maximum number of active agents during this heartbeat timeframe.
- **AgentConnectionErrorsCount** The number of non-timeout errors associated with the host/agent channel.
- **LastAgentConnectionError** The last non-timeout error that happened in the host/agent channel.
- **Flags** Flags that indicate device state, such as network, battery, and opt-in state.
- **CensusTaskEnabled** Indicates whether Census is enabled.
- **CensusExitCode** The last exit code of the Census task.
- **CensusStartTime** The time of the last Census run.

TelClientSynthetic.PrivacySettingsAfterCreatorsUpdate

This event sends basic data on privacy settings before and after a feature update. This is used to ensure that customer privacy settings are correctly migrated across feature updates.

The following fields are available:

- **PostUpgradeSettings** The privacy settings after a feature update.
- **PreUpgradeSettings** The privacy settings before a feature update.

DxgKernelTelemetry events

DxgKrnlTelemetry.GPUAdapterInventoryV2

This event sends basic GPU and display driver information to keep Windows and display drivers up-to-date.

The following fields are available:

- **version** The event version.
- **bootId** The system boot ID.
- **aiSeqId** The event sequence ID.
- **MeasureEnabled** Is the device listening to MICROSOFT_KEYWORD_MEASURES?
- **TelemetryEnabled** Is the device listening to MICROSOFT_KEYWORD_TELEMETRY?
- **InterfaceId** The GPU interface ID.
- **GPUVendorID** The GPU vendor ID.
- **GPUDeviceID** The GPU device ID.

- **SubVendorID** The GPU sub vendor ID.
- **SubSystemID** The subsystem ID.
- **GPURevisionID** The GPU revision ID.
- **DriverVersion** The display driver version.
- **DriverDate** The date of the display driver.
- **DriverRank** The rank of the display driver.
- **IsMiracastSupported** Does the GPU support Miracast?
- **IsMsMiracastSupported** Are the GPU Miracast capabilities driven by a Microsoft solution?
- **IsHybridDiscrete** Does the GPU have discrete GPU capabilities in a hybrid device?
- **IsHybridIntegrated** Does the GPU have integrated GPU capabilities in a hybrid device?
- **IsMPOSupported** Does the GPU support Multi-Plane Overlays?
- **IsLDA** Is the GPU comprised of Linked Display Adapters?
- **IsMismatchLDA** Is at least one device in the Linked Display Adapters chain from a different vendor?
- **IsPostAdapter** Is this GPU the POST GPU in the device?
- **IsSoftwareDevice** Is this a software implementation of the GPU?
- **IsRenderDevice** Does the GPU have rendering capabilities?
- **IsDisplayDevice** Does the GPU have displaying capabilities?
- **WDDMVersion** The Windows Display Driver Model version.
- **DisplayAdapterLuid** The display adapter LUID.
- **GPUPreemptionLevel** The maximum preemption level supported by GPU for graphics payload.
- **ComputePreemptionLevel** The maximum preemption level supported by GPU for compute payload.
- **TelInvEvntTrigger** What triggered this event to be logged? Example: 0 (GPU enumeration) or 1 (DxgKrnlTelemetry provider toggling)
- **DedicatedVideoMemoryB** The amount of dedicated VRAM of the GPU (in bytes).
- **DedicatedSystemMemoryB** The amount of system memory dedicated for GPU use (in bytes).
- **SharedSystemMemoryB** The amount of system memory shared by GPU and CPU (in bytes).
- **NumVidPnSources** The number of supported display output sources.
- **NumVidPnTargets** The number of supported display output targets.

Fault Reporting events

Microsoft.Windows.FaultReporting.AppCrashEvent

This event sends data about crashes for both native and managed applications, to help keep Windows up to date. The data includes information about the crashing process and a summary of its exception record. It does not contain any Watson bucketing information. The bucketing information is recorded in a Windows Error Reporting (WER) event that is generated when the WER client reports the crash to the Watson service, and the WER event will contain the same ReportID (see field 14 of crash event, field 19 of WER event) as the crash event for the crash being reported. AppCrash is emitted once for each crash handled by WER (e.g. from an unhandled exception or FailFast or ReportException). Note that Generic Watson event types (e.g. from PLM) that may be considered crashes" by a user DO NOT emit this event.

The following fields are available:

- **ProcessId** The ID of the process that has crashed.
- **ProcessCreateTime** The time of creation of the process that has crashed.
- **ExceptionCode** The exception code returned by the process that has crashed.
- **ExceptionOffset** The address where the exception had occurred.
- **AppName** The name of the app that has crashed.
- **AppVersion** The version of the app that has crashed.

- **AppTimeStamp** The date/time stamp of the app.
- **ModName** Exception module name (e.g. bar.dll).
- **ModVersion** The version of the module that has crashed.
- **ModTimeStamp** The date/time stamp of the module.
- **PackageFullName** Store application identity.
- **PackageRelativeAppId** Store application identity.
- **ProcessArchitecture** Architecture of the crashing process, as one of the PROCESSOR_ARCHITECTURE_* constants: 0: PROCESSOR_ARCHITECTURE_INTEL. 5: PROCESSOR_ARCHITECTURE_ARM. 9: PROCESSOR_ARCHITECTURE_AMD64. 12: PROCESSOR_ARCHITECTURE_ARM64.
- **ReportId** A GUID used to identify the report. This can used to track the report across Watson.
- **Flags** Flags indicating how reporting is done. For example, queue the report, do not offer JIT debugging, or do not terminate the process after reporting.
- **AppSessionGuid** GUID made up of process ID and is used as a correlation vector for process instances in the telemetry backend.
- **TargetAppId** The kernel reported AppId of the application being reported.
- **TargetAppVer** The specific version of the application being reported
- **TargetAsId** The sequence number for the hanging process.

Hang Reporting events

Microsoft.Windows.HangReporting.AppHangEvent

This event sends data about hangs for both native and managed applications, to help keep Windows up to date. It does not contain any Watson bucketing information. The bucketing information is recorded in a Windows Error Reporting (WER) event that is generated when the WER client reports the hang to the Watson service, and the WER event will contain the same ReportID (see field 13 of hang event, field 19 of WER event) as the hang event for the hang being reported. AppHang is reported only on PC devices. It handles classic Win32 hangs and is emitted only once per report. Some behaviors that may be perceived by a user as a hang are reported by app managers (e.g. PLM/RM/EM) as Watson Generics and will not produce AppHang events.

The following fields are available:

- **AppName** The name of the app that has hung.
- **TypeCode** Bitmap describing the hang type.
- **ProcessId** The ID of the process that has hung.
- **UTCReplace_TargetAppId** The kernel reported AppId of the application being reported.
- **ProcessCreateTime** The time of creation of the process that has hung.
- **UTCReplace_TargetAppVer** The specific version of the application being reported.
- **WaitingOnAppName** If this is a cross process hang waiting for an application, this has the name of the application.
- **PackageRelativeAppId** Store application identity.
- **ProcessArchitecture** Architecture of the hung process, as one of the PROCESSOR_ARCHITECTURE_* constants: 0: PROCESSOR_ARCHITECTURE_INTEL. 5: PROCESSOR_ARCHITECTURE_ARM. 9: PROCESSOR_ARCHITECTURE_AMD64. 12: PROCESSOR_ARCHITECTURE_ARM64.
- **WaitingOnPackageRelativeAppId** If this is a cross process hang waiting for a package, this has the relative application id of the package.
- **WaitingOnAppVersion** If this is a cross process hang, this has the version of the application for which it is waiting.
- **AppSessionGuid** GUID made up of process id used as a correlation vector for process instances in the telemetry backend.
- **WaitingOnPackageFullName** If this is a cross process hang waiting for a package, this has the full name of

the package for which it is waiting.

- **PackageFullName** Store application identity.
- **AppVersion** The version of the app that has hung.
- **ReportId** A GUID used to identify the report. This can used to track the report across Watson.
- **TargetAppId** The kernel reported AppId of the application being reported.
- **TargetAppVer** The specific version of the application being reported.
- **TargetAsId** The sequence number for the hanging process.

Inventory events

Microsoft.Windows.Inventory.Core.AmiTelCacheChecksum

This event captures basic checksum data about the device inventory items stored in the cache for use in validating data completeness for Microsoft.Windows.Inventory.Core events. The fields in this event may change over time, but they will always represent a count of a given object.

The following fields are available:

- **Device** A count of device objects in cache
- **DeviceCensus** A count of devicecensus objects in cache
- **DriverPackageExtended** A count of driverpackageextended objects in cache
- **File** A count of file objects in cache
- **Generic** A count of generic objects in cache
- **HwItem** A count of hwitem objects in cache
- **InventoryApplication** A count of application objects in cache
- **InventoryApplicationFile** A count of application file objects in cache
- **InventoryDeviceContainer** A count of device container objects in cache
- **InventoryDeviceMediaClass** A count of device media objects in cache
- **InventoryDevicePnp** A count of devicepnp objects in cache
- **InventoryDriverBinary** A count of driver binary objects in cache
- **InventoryDriverPackage** A count of device objects in cache
- **Metadata** A count of metadata objects in cache
- **Orphan** A count of orphan file objects in cache
- **Programs** A count of program objects in cache
- **FileSigningInfo** A count of file signing info objects in cache.
- **InventoryDeviceInterface** A count of inventory device interface objects in cache.

Microsoft.Windows.Inventory.Core.AmiTelCacheVersions

This event sends inventory component versions for the Device Inventory data.

The following fields are available:

- **aeinv** The version of the App inventory component.
- **devinv** The file version of the Device inventory component.

Microsoft.Windows.Inventory.Core.InventoryDeviceUsbHubClassStartSync

This event indicates that a new set of InventoryDeviceUsbHubClassAdd events will be sent

The following fields are available:

- **InventoryVersion** The version of the inventory file generating the events
- ### Microsoft.Windows.Inventory.Core.InventoryDeviceUsbHubClassAdd

This event sends basic metadata about the USB hubs on the device

The following fields are available:

- **InventoryVersion** The version of the inventory file generating the events
- **TotalUserConnectablePorts** Total number of connectable USB ports
- **TotalUserConnectableTypeCPorts** Total number of connectable USB Type C ports
- ### Microsoft.Windows.Inventory.Core.InventoryApplicationAdd

This event sends basic metadata about an application on the system to help keep Windows up to date.

The following fields are available:

- **ProgramInstanceId** A hash of the file IDs in an app.
- **Name** The name of the application. Location pulled from depends on 'Source' field.
- **Type** One of ("Application", "Hotfix", "BOE", "Service", "Unknown"). Application indicates Win32 or Appx app, Hotfix indicates app updates (KBs), BOE indicates it's an app with no ARP or MSI entry, Service indicates that it is a service. Application and BOE are the ones most likely seen.
- **Publisher** The Publisher of the application. Location pulled from depends on the 'Source' field.
- **Version** The version number of the program.
- **Language** The language code of the program.
- **Source** How the program was installed (ARP, MSI, Appx, etc...)
- **MsiProductCode** A GUID that describe the MSI Product.
- **MsiPackageCode** A GUID that describes the MSI Package. Multiple 'Products' (apps) can make up an MsiPackage.
- **HiddenArp** Indicates whether a program hides itself from showing up in ARP.
- **OSVersionAtInstallTime** The four octets from the OS version at the time of the application's install.
- **RootDirPath** The path to the root directory where the program was installed.
- **InstallDate** The date the application was installed (a best guess based on folder creation date heuristics)
- **InstallDateMsi** The install date if the application was installed via MSI. Passed as an array.
- **InstallDateFromLinkFile** The estimated date of install based on the links to the files. Passed as an array.
- **InstallDateArpLastModified** The date of the registry ARP key for a given application. Hints at install date but not always accurate. Passed as an array.
- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **objectInstanceId** ProgramId (a hash of Name, Version, Publisher, and Language of an application used to identify it).
- **PackageFullName** The package full name for a Store application.
- **InventoryVersion** The version of the inventory file generating the events.
- **StoreAppType** A sub-classification for the type of Microsoft Store app, such as UWP or Win8StoreApp.

Microsoft.Windows.Inventory.Core.InventoryApplicationRemove

This event indicates that a new set of InventoryDevicePnpAdd events will be sent.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryApplicationStartSync

This event indicates that a new set of InventoryApplicationAdd events will be sent.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDeviceContainerAdd

This event sends basic metadata about a device container (such as a monitor or printer as opposed to a PNP device) to help keep Windows up-to-date.

The following fields are available:

- **ModelName** The model name.
- **ModelId** A model GUID.
- **PrimaryCategory** The primary category for the device container.
- **Categories** A comma separated list of functional categories in which the container belongs.
- **IsConnected** For a physically attached device, this value is the same as IsPresent. For wireless a device, this value represents a communication link.
- **IsActive** Is the device connected, or has it been seen in the last 14 days?
- **IsPaired** Does the device container require pairing?
- **IsNetworked** Is this a networked device?
- **IsMachineContainer** Is the container the root device itself?
- **FriendlyName** The name of the device container.
- **DiscoveryMethod** The discovery method for the device container.
- **ModelNumber** The model number for the device container.
- **Manufacturer** The manufacturer name for the device container.
- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **objectInstanceId** ContainerId
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDeviceContainerRemove

This event indicates that the InventoryDeviceContainer object is no longer present.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDeviceContainerStartSync

This event indicates that a new set of InventoryDeviceContainerAdd events will be sent.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDeviceInterfaceAdd

This event retrieves information about what sensor interfaces are available on the device.

The following fields are available:

- **InventoryVersion** The version of the inventory file generating the events.
- **Accelerometer3D** Indicates if an Accelerator3D sensor is found.
- **ActivityDetection** Indicates if an Activity Detection sensor is found.
- **AmbientLight** Indicates if an Ambient Light sensor is found.
- **Barometer** Indicates if a Barometer sensor is found.
- **Custom** Indicates if a Custom sensor is found.

- **FloorElevation** Indicates if a Floor Elevation sensor is found.
- **GeomagneticOrientation** Indicates if a Geo Magnetic Orientation sensor is found.
- **GravityVector** Indicates if a Gravity Detector sensor is found.
- **Gyrometer3D** Indicates if a Gyrometer3D sensor is found.
- **Humidity** Indicates if a Humidity sensor is found.
- **LinearAccelerometer** Indicates if a Linear Accelerometer sensor is found.
- **Magnetometer3D** Indicates if a Magnetometer3D sensor is found.
- **Orientation** Indicates if an Orientation sensor is found.
- **Pedometer** Indicates if a Pedometer sensor is found.
- **Proximity** Indicates if a Proximity sensor is found.
- **RelativeOrientation** Indicates if a Relative Orientation sensor is found.
- **SimpleDeviceOrientation** Indicates if a Simple Device Orientation sensor is found.
- **Temperature** Indicates if a Temperature sensor is found.
- **EnergyMeter** Indicates if an Energy sensor is found.

Microsoft.Windows.Inventory.Core.InventoryDeviceInterfaceStartSync

This event indicates that a new set of InventoryDeviceInterfaceAdd events will be sent.

The following fields are available:

- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDeviceMediaClassAdd

This event sends additional metadata about a PNP device that is specific to a particular class of devices to help keep Windows up to date while reducing overall size of data payload.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.
- **Audio_CaptureDriver** The Audio device capture driver endpoint.
- **Audio_RenderDriver** The Audio device render driver endpoint.

Microsoft.Windows.Inventory.Core.InventoryDeviceMediaClassRemove

This event indicates that the InventoryDeviceMediaClassRemove object is no longer present.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDeviceMediaClassStartSync

This event indicates that a new set of InventoryDeviceMediaClassSAdd events will be sent.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDevicePnpAdd

This event sends basic metadata about a PNP device and its associated driver to help keep Windows up-to-date.

The following fields are available:

- **HWID** A JSON array that provides the value and order of the HWID tree for the device.

- **COMPID** A JSON array the provides the value and order of the compatible ID tree for the device.
- **InstallState** The device installation state. One of these values: https://msdn.microsoft.com/en-us/library/windows/hardware/ff543130.aspx
- **Enumerator** The bus that enumerated the device.
- **ContainerId** A system-supplied GUID that uniquely groups the functional devices associated with a single-function or multifunction device installed in the device.
- **DeviceState** DeviceState is a bitmask of the following: DEVICE_IS_CONNECTED 0x0001 (currently only for container). DEVICE_IS_NETWORK_DEVICE 0x0002 (currently only for container). DEVICE_IS_PAIRED 0x0004 (currently only for container). DEVICE_IS_ACTIVE 0x0008 (currently never set). DEVICE_IS_MACHINE 0x0010 (currently only for container). DEVICE_IS_PRESENT 0x0020 (currently always set). DEVICE_IS_HIDDEN 0x0040. DEVICE_IS_PRINTER 0x0080 (currently only for container). DEVICE_IS_WIRELESS 0x0100. DEVICE_IS_WIRELESS_FAT 0x0200. The most common values are therefore: 32 (0x20)= device is present. 96 (0x60)= device is present but hidden. 288 (0x120)= device is a wireless device that is present.
- **ParentId** Device instance id of the parent of the device.
- **STACKID** A JSON array that provides the value and order of the STACKID tree for the device.
- **Description** The device description.
- **MatchingID** Represents the hardware ID or compatible ID that Windows uses to install a device instance.
- **Class** The device setup class of the driver loaded for the device.
- **ClassGuid** The device setup class guid of the driver loaded for the device.
- **Manufacturer** The device manufacturer.
- **Model** The device model.
- **Inf** The INF file name.
- **DriverVerVersion** The version of the driver loaded for the device.
- **DriverVerDate** The date of the driver loaded for the device.
- **Provider** The device provider.
- **DriverPackageStrongName** The immediate parent directory name in the Directory field of InventoryDriverPackage.
- **Service** The device service name.
- **LowerClassFilters** Lower filter class drivers IDs installed for the device.
- **LowerFilters** Lower filter drivers IDs installed for the device.
- **UpperClassFilters** Upper filter class drivers IDs installed for the device.
- **UpperFilters** Upper filter drivers IDs installed for the device.
- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **DriverId** A unique identifier for the installed device.
- **DriverName** The name of the driver image file.
- **InventoryVersion** The version of the inventory file generating the events.
- **ProblemCode** The current error code for the device.

Microsoft.Windows.Inventory.Core.InventoryDevicePnpRemove

This event indicates that the InventoryDevicePnpRemove object is no longer present.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDevicePnpStartSync

This event indicates that a new set of InventoryDevicePnpAdd events will be sent.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDriverBinaryAdd

This event sends basic metadata about driver files running on the system to help keep Windows up-to-date.

The following fields are available:

- **DriverName** The file name of the driver.
- **Inf** The name of the INF file.
- **DriverPackageStrongName** The strong name of the driver package.
- **DriverCompany** The company name that developed the driver.
- **DriverCheckSum** The checksum of the driver file.
- **DriverTimeStamp** The low 32 bits of the time stamp of the driver file.
- **DriverType** A bitfield of driver attributes: 1. define DRIVER_MAP_DRIVER_TYPE_PRINTER 0x0001. 2. define DRIVER_MAP_DRIVER_TYPE_KERNEL 0x0002. 3. define DRIVER_MAP_DRIVER_TYPE_USER 0x0004. 4. define DRIVER_MAP_DRIVER_IS_SIGNED 0x0008. 5. define DRIVER_MAP_DRIVER_IS_INBOX 0x0010. 6. define DRIVER_MAP_DRIVER_IS_WINQUAL 0x0040. 7. define DRIVER_MAP_DRIVER_IS_SELF_SIGNED 0x0020. 8. define DRIVER_MAP_DRIVER_IS_CI_SIGNED 0x0080. 9. define DRIVER_MAP_DRIVER_HAS_BOOT_SERVICE 0x0100. 10. define DRIVER_MAP_DRIVER_TYPE_I386 0x10000. 11. define DRIVER_MAP_DRIVER_TYPE_IA64 0x20000. 12. define DRIVER_MAP_DRIVER_TYPE_AMD64 0x40000. 13. define DRIVER_MAP_DRIVER_TYPE_ARM 0x100000. 14. define DRIVER_MAP_DRIVER_TYPE_THUMB 0x200000. 15. define DRIVER_MAP_DRIVER_TYPE_ARMNT 0x400000. 16. define DRIVER_MAP_DRIVER_IS_TIME_STAMPED 0x800000.
- **DriverInBox** Is the driver included with the operating system?
- **DriverSigned** Is the driver signed?
- **DriverIsKernelMode** Is it a kernel mode driver?
- **DriverVersion** The version of the driver file.
- **ImageSize** The size of the driver file.
- **Product** The product name that is included in the driver file.
- **ProductVersion** The product version that is included in the driver file.
- **WdfVersion** The Windows Driver Framework version.
- **Service** The name of the service that is installed for the device.
- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDriverBinaryRemove

This event indicates that the InventoryDriverBinary object is no longer present.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDriverBinaryStartSync

This event indicates that a new set of InventoryDriverBinaryAdd events will be sent.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDriverPackageAdd

This event sends basic metadata about drive packages installed on the system to help keep Windows up-to-date.

The following fields are available:

- **Inf** The INF name of the driver package.
- **ClassGuid** The class GUID for the device driver.
- **Class** The class name for the device driver.
- **Directory** The path to the driver package.
- **Date** The driver package date.
- **Version** The version of the driver package.
- **Provider** The provider for the driver package.
- **SubmissionId** The HLK submission ID for the driver package.
- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.
- **DriverInBox** Is the driver included with the operating system?

Microsoft.Windows.Inventory.Core.InventoryDriverPackageRemove

This event indicates that the InventoryDriverPackageRemove object is no longer present.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDriverPackageStartSync

This event indicates that a new set of InventoryDriverPackageAdd events will be sent.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Indicators.Checksum

This event summarizes the counts for the InventoryMiscellaneousUexIndicatorAdd events.

The following fields are available:

- **ChecksumDictionary** A count of each operating system indicator.
- **PCFP** Equivalent to the InventoryId field that is found in other core events.

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeVBAAdd

This event provides a summary rollup count of conditions encountered while performing a local scan of Office files, analyzing for known VBA programmability compatibility issues between legacy office version and ProPlus, and between 32 and 64-bit versions

The following fields are available:

- **Design** Count of files with design issues found
- **Design_x64** Count of files with 64 bit design issues found
- **DuplicateVBA** Count of files with duplicate VBA code
- **HasVBA** Count of files with VBA code
- **Inaccessible** Count of files that were inaccessible for scanning
- **Issues** Count of files with issues detected
- **Issues_x64** Count of files with 64-bit issues detected
- **IssuesNone** Count of files with no issues detected

- **IssuesNone_x64** Count of files with no 64-bit issues detected
- **Locked** Count of files that were locked, preventing scanning
- **NoVBA** Count of files with no VBA inside
- **Protected** Count of files that were password protected, preventing scanning
- **RemLimited** Count of files that require limited remediation changes
- **RemLimited_x64** Count of files that require limited remediation changes for 64-bit issues
- **RemSignificant** Count of files that require significant remediation changes
- **RemSignificant_x64** Count of files that require significant remediation changes for 64-bit issues
- **Score** Overall compatibility score calculated for scanned content
- **Score_x64** Overall 64-bit compatibility score calculated for scanned content
- **Total** Total number of files scanned
- **Validation** Count of files that require additional manual validation
- **Validation_x64** Count of files that require additional manual validation for 64-bit issues

Microsoft.Windows.Inventory.Core.InventoryApplicationFrameworkStartSync

This event indicates that a new set of InventoryApplicationFrameworkAdd events will be sent

The following fields are available:

- **InventoryVersion** The version of the inventory file generating the events

Microsoft.Windows.Inventory.Core.InventoryApplicationFrameworkAdd

This event provides the basic metadata about the frameworks an application may depend on

The following fields are available:

- **FileId** A hash that uniquely identifies a file
- **Frameworks** The list of frameworks this file depends on
- **InventoryVersion** The version of the inventory file generating the events
- **ProgramId** A hash of the Name, Version, Publisher, and Language of an application used to identify it

Microsoft.Windows.Inventory.Indicators.InventoryMiscellaneousUexIndicatorAdd

These events represent the basic metadata about the OS indicators installed on the system which are used for keeping the device up-to-date.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.
- **IndicatorValue** The indicator value

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeVBARuleViolationsStartSync

This event indicates that a new sync is being generated for this object type.

There are no fields in this event.

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeVBAStartSync

This event indicates that a new sync is being generated for this object type.

There are no fields in this event.

Microsoft.Windows.Inventory.Indicators.InventoryMiscellaneousUexIndicatorRemove

This event is a counterpart to InventoryMiscellaneousUexIndicatorAdd, indicating that the item has been removed. There are no additional unique fields in this event.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.

Microsoft.Windows.Inventory.Indicators.InventoryMiscellaneousUexIndicatorStartSync

This event indicates that a new set of InventoryMiscellaneousUexIndicatorAdd events will be sent.

The following fields are available:

- **PartB_Ms.Device.DeviceInventoryChange** See the Common Data Fields section.

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeVBARuleViolationsAdd

This event provides data on Microsoft Office VBA rule violations, including a rollup count per violation type, giving an indication of remediation requirements for an organization. The event identifier is a unique GUID, associated with the validation rule

The following fields are available:

- **Count** Count of total Microsoft Office VBA rule violations

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeAddInAdd

This event provides data on the installed Office Add-ins.

- **AddInCLSID** The CLSID key office the Office addin.
- **AddInId** The ID of the Office addin.
- **BinFileTimestamp** The timestamp of the Office addin.
- **BinFileVersion** The version of the Office addin.
- **Description** The description of the Office addin.
- **FileId** The file ID of the Office addin.
- **FriendlyName** The friendly name of the Office addin.
- **FullPath** The full path to the Office addin.
- **LoadBehavior** A Uint32 that describes the load behavior.
- **LoadTime** The load time for the Office addin.
- **OfficeApplication** The OIffice application for this addin.
- **OfficeArchitecture** The architecture of the addin.
- **OfficeVersion** The Office version for this addin.
- **OutlookCrashingAddin** A boolean value that indicates if crashes have been found for this addin.
- **Provider** The provider name for this addin.

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeAddInStartSync

This event indicates that a new sync is being generated for this object type.

There are no fields in this event.

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeIdentifiersAdd

This event provides data on the installed Office identifiers.

- **OAudienceData** The Office Audience descriptor.
- **OAudienceId** The Office Audience ID.
- **OMID** The Office machine ID.
- **OPlatform** The Office architecture.
- **OVersion** The Office version
- **OTenantId** The Office 365 Tenant GUID.
- **OWowMID** The Office machine ID.

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeIdentifiersStartSync

This event indicates that a new sync is being generated for this object type.

There are no fields in this event.

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeIESettingsStartSync

This event indicates that a new sync is being generated for this object type.

There are no fields in this event.

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeProductsStartSync

This event indicates that a new sync is being generated for this object type.

There are no fields in this event.

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeIESettingsAdd

This event provides data on the installed Office-related Internet Explorer features.

- **OleFeatureAddon** For more information, see the Office-related Internet Feature Control Keys.
- **OleMachineLockdown** For more information, see the Office-related Internet Feature Control Keys.
- **OleMimeHandling** For more information, see the Office-related Internet Feature Control Keys.
- **OleMimeSniffing** For more information, see the Office-related Internet Feature Control Keys.
- **OleNoAxInstall** For more information, see the Office-related Internet Feature Control Keys.
- **OleNoDownload** For more information, see the Office-related Internet Feature Control Keys.
- **OleObjectCaching** For more information, see the Office-related Internet Feature Control Keys.
- **OlePasswordDisable** For more information, see the Office-related Internet Feature Control Keys.
- **OleSafeBind** For more information, see the Office-related Internet Feature Control Keys.
- **OleSecurityBand** For more information, see the Office-related Internet Feature Control Keys.
- **OleUncSaveCheck** For more information, see the Office-related Internet Feature Control Keys.
- **OleValidateUrl** For more information, see the Office-related Internet Feature Control Keys.
- **OleWebOcPopup** For more information, see the Office-related Internet Feature Control Keys.
- **OleWinRestrict** For more information, see the Office-related Internet Feature Control Keys.
- **OleZoneElevate** For more information, see the Office-related Internet Feature Control Keys.

Microsoft.Windows.Inventory.General.InventoryMiscellaneousOfficeProductsAdd

This event describes the Office products that are installed.

- **OC2rApps** The Office Click-to-Run apps.
- **OC2rSkus** The Office Click-to-Run products.
- **OMsiApps** The Office MSI apps.
- **OProductCodes** The Office MSI product code.

OneDrive events

Microsoft.OneDrive.Sync.Setup.APIOperation

This event includes basic data about install and uninstall OneDrive API operations.

The following fields are available:

- **APIName** The name of the API.
- **ScenarioName** The name of the scenario.
- **Duration** How long the operation took.
- **isSuccess** Was the operation successful?
- **ResultCode** The result code.

Microsoft.OneDrive.Sync.Setup.EndExperience

This event includes a success or failure summary of the installation.

The following fields are available:

- **APIName** The name of the API.
- **ScenarioName** The name of the scenario.
- **Hresult** The HResult of the operation.
- **isSuccess** Was the operation successful?

Microsoft.OneDrive.Sync.Setup.OSUpgradeInstallationOperation

This event is related to the OS version when the OS is upgraded with OneDrive installed.

The following fields are available:

- **HResult** The HResult of the operation.
- **SourceOSVersion** The source version of the operating system.
- **SourceOSBuildNumber** The source build number of the operating system.
- **SourceOSBuildBranch** The source branch of the operating system.
- **CurrentOSVersion** The current version of the operating system.
- **CurrentOSBuildNumber** The current build number of the operating system.
- **CurrentOSBuildBranch** The current branch of the operating system.
- **CurrentOneDriveVersion** The current version of OneDrive.

Microsoft.OneDrive.Sync.Setup.RegisterStandaloneUpdaterAPIOperation

This event is related to registering or unregistering the OneDrive update task.

The following fields are available:

- **APIName** The name of the API.
- **ScenarioName** The name of the scenario.
- **UnregisterOldTaskResult** The HResult of the UnregisterOldTask operation.
- **RegisterNewTaskResult** The HResult of the RegisterNewTask operation.
- **isSuccess** Was the operation successful?

Microsoft.OneDrive.Sync.Setup.SetupCommonData

This event contains basic OneDrive configuration data that helps to diagnose failures.

The following fields are available:

- **AppVersion** The version of the app.
- **OfficeVersion** The version of Office that is installed.
- **BuildArch** Is the architecture x86 or x64?
- **Market** Which market is this in?
- **OneDriveDeviceId** The OneDrive device ID.
- **MachineGuid** The CEIP machine ID.
- **IsMSFTInternal** Is this an internal Microsoft device?
- **OSDeviceName** Only if the device is internal to Microsoft, the device name.
- **OSUserName** Only if the device is internal to Microsoft, the user name.
- **Environment** Is the device on the production or int service?
- **OfficeVersionString** The version of Office that is installed.
- **BuildArchitecture** Is the architecture x86 or x64?
- **UserGuid** The CEIP user ID.

- **MSFTInternal** Is this an internal Microsoft device?

Microsoft.OneDrive.Sync.Updater.CommonData

This event contains basic OneDrive configuration data that helps to diagnose failures.

The following fields are available:

- **AppVersion** The version of the app.
- **OfficeVersion** The version of Office that is installed.
- **BuildArch** Is the architecture x86 or x64?
- **Market** Which market is this in?
- **OneDriveDeviceId** The OneDrive device ID.
- **MachineGuid** The CEIP machine ID.
- **IsMSFTInternal** Is this an internal Microsoft device?
- **OSDeviceName** Only if the device is internal to Microsoft, the device name.
- **OSUserName** Only if the device is internal to Microsoft, the user name.
- **Environment** Is the device on the production or int service?
- **UserGuid** A unique global user identifier.

Microsoft.OneDrive.Sync.Updater.ComponentInstallState

This event determines the installation state of dependent OneDrive components.

The following fields are available:

- **ComponentName** The name of the dependent component.
- **isInstalled** Is the dependent component installed?

Microsoft.OneDrive.Sync.Updater.OfficeRegistration

This event determines the status of the OneDrive integration with Microsoft Office.

The following fields are available:

- **isValid** Is the Microsoft Office registration valid?

Microsoft.OneDrive.Sync.Updater.OverlayIconStatus

This event indicates if the OneDrive overlay icon is working correctly. 0 = healthy; 1 = can be fixed; 2 = broken

The following fields are available:

- **32bit** The status of the OneDrive overlay icon on a 32-bit operating system.
- **64bit** The status of the OneDrive overlay icon on a 64-bit operating system.

Microsoft.OneDrive.Sync.Updater.RepairResult

The event determines the result of the installation repair.

The following fields are available:

- **hr** The HResult of the operation.

Microsoft.OneDrive.Sync.Updater.SetupBinaryDownloadHResult

This event indicates the status when downloading the OneDrive setup file.

The following fields are available:

- **hr** The HResult of the operation.

Microsoft.OneDrive.Sync.Updater.UpdateOverallResult

This event determines the outcome of the operation.

The following fields are available:

- **UpdaterVersion** The version of the updater.
- **IsLoggingEnabled** Is logging enabled?
- **hr** The HResult of the operation.

Microsoft.OneDrive.Sync.Updater.UpdateTierReg

This event determines status of the update tier registry values.

The following fields are available:

- **regReadEnterpriseHr** The HResult of the enterprise reg read value.
- **regReadTeamHr** The HResult of the team reg read value.

Microsoft.OneDrive.Sync.Updater.UpdateXmlDownloadHResult

This event determines the status when downloading the OneDrive update configuration file.

The following fields are available:

- **hr** The HResult of the operation.

Microsoft.OneDrive.Sync.Updater.WebConnectionStatus

This event determines the error code that was returned when verifying Internet connectivity.

The following fields are available:

- **winInetError** The HResult of the operation.

Setup events

SetupPlatformTel.SetupPlatformTelActivityEvent

This event sends a unique ID that can be used to bind Setup Platform events together, to help keep Windows up to date.

The following fields are available:

- **FieldName** Retrieves the event name/data point. Examples: InstallStartTime, InstallEndtime, OverallResult etc.
- **GroupName** Retrieves the groupname the event belongs to. Example: Install Information, DU Information, Disk Space Information etc.
- **Value** Retrieves the value associated with the corresponding event name. For example: For time-related events, this will include the system time.
- **ActivityId** Provides a unique Id to correlate events that occur between a activity start event, and a stop event
- **ActivityName** Provides a friendly name of the package type that belongs to the ActivityId (Setup, LanguagePack, GDR, Driver, etc.)

SetupPlatformTel.SetupPlatformTelActivityStarted

This event sends basic metadata about the update installation process generated by SetupPlatform to help keep Windows up to date.

The following fields are available:

- **Name** The name of the dynamic update type. Example: GDR driver

SetupPlatformTel.SetupPlatformTelActivityStopped

This event sends basic metadata about the update installation process generated by SetupPlatform to help keep

Windows up to date.

SetupPlatformTel.SetupPlatformTelEvent

This service retrieves events generated by SetupPlatform, the engine that drives the various deployment scenarios.

The following fields are available:

- **FieldName** Retrieves the event name/data point. Examples: InstallStartTime, InstallEndtime, OverallResult etc.
- **Value** Retrieves the value associated with the corresponding event name (Field Name). For example: For time related events this will include the system time.
- **GroupName** Retrieves the groupname the event belongs to. Example: Install Information, DU Information, Disk Space Information etc.

Shared PC events

Microsoft.Windows.SharedPC.AccountManager.DeleteUserAccount

Activity for deletion of a user account for devices set up for Shared PC mode as part of the Transient Account Manager to help keep Windows up to date. Deleting unused user accounts on shared devices frees up disk space to improve Windows Update success rates.

The following fields are available:

- **wilActivity** Windows Error Reporting data collected when there is a failure in deleting a user account with the Transient Account Manager.
- **userSid** The security identifier of the account.
- **accountType** The type of account that was deleted. Example: AD, AAD, or Local

Microsoft.Windows.SharedPC.AccountManager.SinglePolicyEvaluation

Activity for run of the Transient Account Manager that determines if any user accounts should be deleted for devices set up for Shared PC mode to help keep Windows up to date. Deleting unused user accounts on shared devices frees up disk space to improve Windows Update success rates

The following fields are available:

- **wilActivity** Windows Error Reporting data collected when there is a failure in evaluating accounts to be deleted with the Transient Account Manager.
- **totalAccountCount** The number of accounts on a device after running the Transient Account Manager policies.
- **evaluationTrigger** When was the Transient Account Manager policies ran? Example: At log off or during maintenance hours

Software update events

SoftwareUpdateClientTelemetry.CheckForUpdates

This event sends tracking data about the software distribution client check for content that is applicable to a device, to help keep Windows up to date

The following fields are available:

- **EventScenario** Indicates the purpose of sending this event - whether because the software distribution just started checking for content, or whether it was cancelled, succeeded, or failed.
- **EventInstanceID** A globally unique identifier for event instance.
- **DeviceModel** What is the device model.
- **BiosName** The name of the device BIOS.
- **BIOSVendor** The vendor of the BIOS.

- **BiosVersion** The version of the BIOS.
- **BiosReleaseDate** The release date of the device BIOS.
- **SystemBIOSMajorRelease** Major version of the BIOS.
- **SystemBIOSMinorRelease** Minor version of the BIOS.
- **BiosFamily** The family of the BIOS (Basic Input Output System).
- **BiosSKUNumber** The sku number of the device BIOS.
- **ClientVersion** The version number of the software distribution client.
- **WUDeviceID** The unique identifier of a specific device, used to identify how many devices are encountering success or a particular issue.
- **CallerApplicationName** The name provided by the caller who initiated API calls into the software distribution client.
- **ProcessName** The process name of the caller who initiated API calls, in the event where CallerApplicationName was not provided.
- **ServiceGuid** An ID which represents which service the software distribution client is checking for content (Windows Update, Microsoft Store, etc.).
- **StatusCode** Indicates the result of a CheckForUpdates event (success, cancellation, failure code HResult).
- **ExtendedStatusCode** Secondary error code for certain scenarios where StatusCode wasn't specific enough.
- **FlightRing** The ring (speed of getting builds) that a device is on if participating in flighting (pre-release builds).
- **FlightBranch** The branch that a device is on if participating in flighting (pre-release builds).
- **RelatedCV** The previous Correlation Vector that was used before swapping with a new one
- **IsWUfBEnabled** Indicates if Windows Update for Business is enabled on the device.
- **IsWUfBDualScanEnabled** Indicates if Windows Update for Business dual scan is enabled on the device.
- **ShippingMobileOperator** The mobile operator that a device shipped on.
- **CurrentMobileOperator** The mobile operator the device is currently connected to.
- **HomeMobileOperator** The mobile operator that the device was originally intended to work with.
- **PhonePreviewEnabled** Indicates whether a phone was getting preview build, prior to flighting (pre-release builds) being introduced.
- **ActivityMatchingId** Contains a unique ID identifying a single CheckForUpdates session from initialization to completion.
- **SyncType** Describes the type of scan the event was
- **IPVersion** Indicates whether the download took place over IPv4 or IPv6
- **NumberOfApplicationsCategoryScanEvaluated** The number of categories (apps) for which an app update scan checked
- **ScanDurationInSeconds** The number of seconds a scan took
- **ScanEnqueueTime** The number of seconds it took to initialize a scan
- **NumberOfLoop** The number of round trips the scan required
- **NumberOfUpdatesEvaluated** The total number of updates which were evaluated as a part of the scan
- **NumberOfNewUpdatesFromServiceSync** The number of updates which were seen for the first time in this scan
- **ServiceUrl** The environment URL a device is configured to scan with
- **Online** Indicates if this was an online scan.
- **AllowCachedResults** Indicates if the scan allowed using cached results.
- **MetadataIntegrityMode** The mode of the update transport metadata integrity check. 0-Unknown, 1-Ignoe, 2-Audit, 3-Enforce
- **TotalNumMetadataSignatures** The total number of metadata signatures checks done for new metadata that was synced down.
- **NumFailedMetadataSignatures** The number of metadata signatures checks which failed for new metadata synced down.

- **MSIError** The last error that was encountered during a scan for updates.

- **DriverError** The error code hit during a driver scan. This is 0 if no error was encountered.

- **FailedUpdatesCount** The number of updates that failed to be evaluated during the scan.

- **FailedUpdateGuids** The GUIDs for the updates that failed to be evaluated during the scan.

- **CapabilityDetectoidGuid** The GUID for a hardware applicability detectoid that could not be evaluated.

- **ExtendedMetadataCabUrl** Hostname that is used to download an update.

- **CDNId** The unique identifier of a specific device, used to identify how many devices are encountering success or a particular issue.

- **CDNCountryCode** Two letter country abbreviation for the CDN's location.

- **NetworkConnectivityDetected** Indicates the type of network connectivity that was detected. 0 - IPv4, 1 - IPv6

- **NumberOfApplicableUpdates** The number of updates which were ultimately deemed applicable to the system after the detection process is complete

- **ApplicableUpdateInfo** Metadata for the updates which were detected as applicable

- **WebServiceRetryMethods** Web service method requests that needed to be retried to complete operation.

- **DeferredUpdates** Update IDs which are currently being deferred until a later time

- **BranchReadinessLevel** The servicing branch configured on the device.

- **DeferralPolicySources** Sources for any update deferral policies defined (GPO = 0x10, MDM = 0x100, Flight = 0x1000, UX = 0x10000).

- **QualityUpdateDeferral** The deferral period configured for quality OS updates on the device (in days).

- **QualityUpdatePause** Indicates whether quality OS updates are paused on the device.

- **QualityUpdatePausePeriod** The pause duration configured for quality OS updates on the device (in days).

- **FeatureUpdateDeferral** The deferral period configured for feature OS updates on the device (in days).

- **FeatureUpdatePause** Indicates whether feature OS updates are paused on the device.

- **FeatureUpdatePausePeriod** The pause duration configured for feature OS updates on the device (in days).

- **DriverExclusionPolicy** Indicates if the policy for not including drivers with Windows Update is enabled.

- **TargetMetadataVersion** For self-initiated healing, this is the target version of the SIH engine to download (if needed). If not, the value is null.

- **CachedEngineVersion** For self-initiated healing, the version of the SIH engine that is cached on the device. If the SIH engine does not exist, the value is null.

- **SearchFilter** Contains information indicating filters applied while checking for content applicable to the device. For example, to filter out all content which may require a reboot.

- **IntentPFNs** Intended application-set metadata for atomic update scenarios.

- **PausedUpdates** A list of UpdateIds which that currently being paused.

- **PauseQualityUpdatesStartTime** If quality OS updates are paused on the device, this is the date and time for the beginning of the pause time window.

- **PauseQualityUpdatesEndTime** If quality OS updates are paused on the device, this is the date and time for the end of the pause time window.

- **PauseFeatureUpdatesStartTime** If feature OS updates are paused on the device, this is the date and time for the beginning of the pause time window.

- **PauseFeatureUpdatesEndTime** If feature OS updates are paused on the device, this is the date and time for the end of the pause time window.

- **Context** Gives context on where the error has occurred. Example: AutoEnable, GetSLSData, AddService, Misc, or Unknown

- **DriverSyncPassPerformed** Were drivers scanned this time?

SoftwareUpdateClientTelemetry.Commit

This event sends data on whether the Update Service has been called to execute an upgrade, to help keep Windows up to date.

The following fields are available:

- **EventScenario** State of call
- **EventInstanceID** A globally unique identifier for event instance.
- **DeviceModel** What is the device model.
- **BiosName** The name of the device BIOS.
- **BIOSVendor** The vendor of the BIOS.
- **BiosVersion** The version of the BIOS.
- **BiosReleaseDate** The release date of the device BIOS.
- **SystemBIOSMajorRelease** Major version of the BIOS.
- **SystemBIOSMinorRelease** Minor version of the BIOS.
- **BiosFamily** The family of the BIOS (Basic Input Output System).
- **BiosSKUNumber** The sku number of the device BIOS.
- **ClientVersion** The version number of the software distribution client.
- **WUDeviceID** UniqueDeviceID
- **ServerId** Identifier for the service to which the software distribution client is connecting, such as Windows Update and Microsoft Store.
- **EventType** Possible values are "Child", "Bundle", or "Driver".
- **UpdateId** Unique Update ID
- **BundleId** Identifier associated with the specific content bundle; should not be all zeros if the bundleID was found.
- **RevisionNumber** Unique revision number of Update
- **HandlerType** Indicates the kind of content (app, driver, windows patch, etc.)
- **BundleRevisionNumber** Identifies the revision number of the content bundle
- **FlightId** The specific id of the flight the device is getting
- **CallerApplicationName** The name provided by the caller who initiated API calls into the software distribution client

SoftwareUpdateClientTelemetry.Download

This event sends tracking data about the software distribution client download of the content for that update, to help keep Windows up to date.

The following fields are available:

- **EventScenario** Indicates the purpose of sending this event - whether because the software distribution just started downloading content, or whether it was cancelled, succeeded, or failed.
- **EventInstanceID** A globally unique identifier for event instance.
- **DeviceModel** What is the device model.
- **BiosName** The name of the device BIOS.
- **BIOSVendor** The vendor of the BIOS.
- **BiosVersion** The version of the BIOS.
- **BiosReleaseDate** The release date of the device BIOS.
- **SystemBIOSMajorRelease** Major version of the BIOS.
- **SystemBIOSMinorRelease** Minor version of the BIOS.
- **BiosFamily** The family of the BIOS (Basic Input Output System).
- **BiosSKUNumber** The sku number of the device BIOS.
- **ClientVersion** The version number of the software distribution client.
- **WUDeviceID** The unique identifier of a specific device, used to identify how many devices are encountering success or a particular issue.

- **CallerApplicationName** The name provided by the caller who initiated API calls into the software distribution client.
- **ProcessName** The process name of the caller who initiated API calls, in the event where CallerApplicationName was not provided.
- **ServiceGuid** An ID which represents which service the software distribution client is installing content for (Windows Update, Microsoft Store, etc.).
- **StatusCode** Indicates the result of a Download event (success, cancellation, failure code HResult).
- **ExtendedStatusCode** Secondary error code for certain scenarios where StatusCode wasn't specific enough.
- **FlightRing** The ring (speed of getting builds) that a device is on if participating in flighting (pre-release builds).
- **FlightBranch** The branch that a device is on if participating in flighting (pre-release builds).
- **RelatedCV** The previous Correlation Vector that was used before swapping with a new one
- **IsWUfBEnabled** Indicates if Windows Update for Business is enabled on the device.
- **IsWUfBDualScanEnabled** Indicates if Windows Update for Business dual scan is enabled on the device.
- **ShippingMobileOperator** The mobile operator that a device shipped on.
- **CurrentMobileOperator** The mobile operator the device is currently connected to.
- **HomeMobileOperator** The mobile operator that the device was originally intended to work with.
- **PhonePreviewEnabled** Indicates whether a phone was opted-in to getting preview builds, prior to flighting (pre-release builds) being introduced.
- **IPVersion** Indicates whether the download took place over IPv4 or IPv6.
- **NetworkCostBitMask** Indicates what kind of network the device is connected to (roaming, metered, over data cap, etc.)
- **NetworkRestrictionStatus** More general version of NetworkCostBitMask, specifying whether Windows considered the current network to be "metered."
- **TimeToEstablishConnection** Time (in ms) it took to establish the connection prior to beginning downloaded.
- **HostName** The hostname URL the content is downloading from.
- **CDNId** ID which defines which CDN the software distribution client downloaded the content from.
- **CDNCountryCode** Two letter country abbreviation for the CDN's location.
- **ActiveDownloadTime** How long the download took, in seconds, excluding time where the update wasn't actively being downloaded.
- **IsDependentSet** Indicates whether a driver is a part of a larger System Hardware/Firmware Update
- **TargetingVersion** For drivers targeted to a specific device model, this is the version number of the drivers being distributed to the device.
- **HardwareId** If this download was for a driver targeted to a particular device model, this ID indicates the model of the device.
- **UpdateImportance** Indicates whether a piece of content was marked as Important, Recommended, or Optional.
- **TargetGroupId** For drivers targeted to a specific device model, this ID indicates the distribution group of devices receiving that driver.
- **RepeatFailFlag** Indicates whether this specific piece of content had previously failed to download.
- **BytesDownloaded** How many bytes were downloaded for an individual piece of content (not the entire bundle).
- **TotalExpectedBytes** The total count of bytes that the download is expected to be.
- **ThrottlingServiceHResult** Result code (success/failure) while contacting a web service to determine whether this device should download content yet.
- **EventType** Possible values are Child, Bundle, or Driver.
- **UpdateId** An identifier associated with the specific piece of content.
- **RevisionNumber** Identifies the revision number of this specific piece of content.
- **BundleId** Identifier associated with the specific content bundle; should not be all zeros if the bundleID was

found.

- **BundleRevisionNumber** Identifies the revision number of the content bundle.
- **HandlerType** Indicates what kind of content is being downloaded (app, driver, windows patch, etc.).
- **DownloadPriority** Indicates whether a download happened at background, normal, or foreground priority.
- **FlightId** The specific id of the flight (pre-release build) the device is getting.
- **Setup360Phase** If the download is for an operating system upgrade, this datapoint indicates which phase of the upgrade is underway.
- **UsedDO** Whether the download used the delivery optimization service.
- **CbsDownloadMethod** Indicates whether the download was a full-file download or a partial/delta download.
- **UsedSystemVolume** Indicates whether the content was downloaded to the device's main system storage drive, or an alternate storage drive.
- **FlightBuildNumber** If this download was for a flight (pre-release build), this indicates the build number of that flight.
- **BundleBytesDownloaded** How many bytes were downloaded for the specific content bundle.
- **BundleRepeatFailFlag** Indicates whether this particular update bundle had previously failed to download.
- **DownloadScenarioId** A unique ID for a given download used to tie together WU and DO events.
- **PackageFullName** The package name of the content.
- **AppXBlockHashValidationFailureCount** A count of the number of blocks that have failed validation after being downloaded.
- **CachedEngineVersion** For self-initiated healing, the version of the SIH engine that is cached on the device. If the SIH engine does not exist, the value is null.
- **TargetMetadataVersion** For self-initiated healing, this is the target version of the SIH engine to download (if needed). If not, the value is null.
- **DownloadType** Differentiates the download type of SIH downloads between Metadata and Payload downloads.
- **WUSetting** Indicates the users' current updating settings.
- **ProcessorArchitecture** Processor architecture of the system (x86, AMD64, ARM).
- **PlatformRole** The PowerPlatformRole as defined on MSDN
- **IsAOACDevice** Is it Always On, Always Connected?
- **EventNamespaceID** Indicates whether the event succeeded or failed. Has the format EventType+Event where Event is Succeeded, Cancelled, Failed, etc.
- **Edition** Indicates the edition of Windows being used.
- **DeviceOEM** What OEM does this device belong to.
- **ClientManagedByWSUSServer** Indicates whether the client is managed by Windows Server Update Services (WSUS).
- **QualityUpdatePause** Indicates whether quality OS updates are paused on the device.
- **FeatureUpdatePause** Indicates whether feature OS updates are paused on the device.
- **AppXDownloadScope** Indicates the scope of the download for application content. For streaming install scenarios, AllContent - non-streaming download, RequiredOnly - streaming download requested content required for launch, AutomaticOnly - streaming download requested automatic streams for the app, and Unknown - for events sent before download scope is determined by the Windows Update client.

SoftwareUpdateClientTelemetry.Install

This event sends tracking data about the software distribution client installation of the content for that update, to help keep Windows up to date.

The following fields are available:

- **EventScenario** Indicates the purpose of sending this event - whether because the software distribution just started installing content, or whether it was cancelled, succeeded, or failed.

- **EventInstanceID** A globally unique identifier for event instance.
- **DeviceModel** What is the device model.
- **BiosName** The name of the device BIOS.
- **BIOSVendor** The vendor of the BIOS.
- **BiosVersion** The version of the BIOS.
- **BiosReleaseDate** The release date of the device BIOS.
- **SystemBIOSMajorRelease** Major version of the BIOS.
- **SystemBIOSMinorRelease** Minor version of the BIOS.
- **BiosFamily** The family of the BIOS (Basic Input Output System).
- **BiosSKUNumber** The sku number of the device BIOS.
- **ClientVersion** The version number of the software distribution client.
- **WUDeviceID** The unique identifier of a specific device, used to identify how many devices are encountering success or a particular issue.
- **CallerApplicationName** The name provided by the caller who initiated API calls into the software distribution client.
- **ProcessName** The process name of the caller who initiated API calls, in the event where CallerApplicationName was not provided.
- **ServiceGuid** An ID which represents which service the software distribution client is installing content for (Windows Update, Microsoft Store, etc.).
- **StatusCode** Indicates the result of an installation event (success, cancellation, failure code HResult).
- **ExtendedStatusCode** Secondary error code for certain scenarios where StatusCode wasn't specific enough.
- **FlightRing** The ring that a device is on if participating in the Windows Insider Program.
- **FlightBranch** The branch that a device is on if participating in the Windows Insider Program.
- **RelatedCV** The previous Correlation Vector that was used before swapping with a new one
- **IsWUfBEnabled** Is Windows Update for Business enabled on the device?
- **IsWUfBDualScanEnabled** Is Windows Update for Business dual scan enabled on the device?
- **ShippingMobileOperator** The mobile operator that a device shipped on.
- **CurrentMobileOperator** Mobile operator that device is currently connected to.
- **HomeMobileOperator** The mobile operator that the device was originally intended to work with.
- **PhonePreviewEnabled** Indicates whether a phone was getting preview build, prior to flighting being introduced.
- **TargetGroupId** For drivers targeted to a specific device model, this ID indicates the distribution group of devices receiving that driver.
- **RepeatFailFlag** Indicates whether this specific piece of content had previously failed to install.
- **EventType** Possible values are Child, Bundle, or Driver.
- **TargetingVersion** For drivers targeted to a specific device model, this is the version number of the drivers being distributed to the device.
- **UpdateImportance** Indicates whether a piece of content was marked as Important, Recommended, or Optional.
- **IsFirmware** Is this update a firmware update?
- **IsFinalOutcomeEvent** Does this event signal the end of the update/upgrade process?
- **IsDependentSet** Is the driver part of a larger System Hardware/Firmware update?
- **DriverPingBack** Contains information about the previous driver and system state.
- **ExtendedErrorCode** The extended error code.
- **CSIErrorType** The stage of CBS installation where it failed.
- **MsiAction** The stage of MSI installation where it failed.
- **MsiProductCode** The unique identifier of the MSI installer.

- **TransactionCode** The ID which represents a given MSI installation
- **HardwareId** If this install was for a driver targeted to a particular device model, this ID indicates the model of the device.
- **IsSuccessFailurePostReboot** Did it succeed and then fail after a restart?
- **UpdateId** Unique update ID
- **RevisionNumber** The revision number of this specific piece of content.
- **BundleId** Identifier associated with the specific content bundle; should not be all zeros if the bundleID was found.
- **BundleRevisionNumber** Identifies the revision number of the content bundle.
- **HandlerType** Indicates what kind of content is being installed. Example: app, driver, Windows update
- **FlightId** The specific ID of the Windows Insider build the device is getting.
- **Setup360Phase** If the install is for an operating system upgrade, indicates which phase of the upgrade is underway.
- **UsedSystemVolume** Indicates whether the content was downloaded and then installed from the device's main system storage drive, or an alternate storage drive.
- **FlightBuildNumber** If this installation was for a Windows Insider build, this is the build number of that build.
- **BundleRepeatFailFlag** Has this particular update bundle previously failed to install?
- **PackageFullName** The package name of the content being installed.
- **CachedEngineVersion** For self-initiated healing, the version of the SIH engine that is cached on the device. If the SIH engine does not exist, the value is null.
- **BundleBytesDownloaded** How many bytes were downloaded for the specific content bundle?
- **CbsDownloadMethod** Was the download a full download or a partial download?
- **ClientManagedByWSUSServer** Is the client managed by Windows Server Update Services (WSUS)?
- **DeviceOEM** What OEM does this device belong to.
- **DownloadPriority** The priority of the download activity.
- **DownloadScenarioId** A unique ID for a given download used to tie together WU and DO events.
- **Edition** Indicates the edition of Windows being used.
- **EventNamespaceID** Indicates whether the event succeeded or failed. Has the format EventType+Event where Event is Succeeded, Cancelled, Failed, etc.
- **IsAOACDevice** Is it Always On, Always Connected? (Mobile device usage model)
- **PlatformRole** The PowerPlatformRole as defined on MSDN.
- **ProcessorArchitecture** Processor architecture of the system (x86, AMD64, ARM).
- **RepeatSuccessInstallFlag** Indicates whether this specific piece of content had previously installed successful, for example if another user had already installed it.
- **WUSetting** Indicates the user's current updating settings.
- **IntentPFNs** Intended application-set metadata for atomic update scenarios.
- **QualityUpdatePause** Are quality OS updates paused on the device?
- **FeatureUpdatePause** Are feature OS updates paused on the device?
- **MergedUpdate** Was the OS update and a BSP update merged for installation?

SoftwareUpdateClientTelemetry.SLSDiscovery

This event sends data about the ability of Windows to discover the location of a backend server with which it must connect to perform updates or content acquisition, in order to determine disruptions in availability of update services and provide context for Windows Update errors.

The following fields are available:

- **EventScenario** Indicates the purpose of sending this event - whether because the software distribution just started checking for content, or whether it was cancelled, succeeded, or failed

- **SusClientId** The unique device ID controlled by the software distribution client

- **WUAVersion** The version number of the software distribution client

- **ServiceID** An ID which represents which service the software distribution client is connecting to (Windows Update, Microsoft Store, etc.)

- **UrlPath** Path to the SLS cab that was downloaded

- **HResult** Indicates the result code of the event (success, cancellation, failure code HResult)

- **IsBackground** Indicates whether the SLS discovery event took place in the foreground or background

- **NextExpirationTime** Indicates when the SLS cab expires

SoftwareUpdateClientTelemetry.UpdateDetected

This event sends data about an AppX app that has been updated from the Microsoft Store, including what app needs an update and what version/architecture is required, in order to understand and address problems with apps getting required updates.

The following fields are available:

- **CallerApplicationName** The name provided by the caller who initiated API calls into the software distribution client

- **ApplicableUpdateInfo** Metadata for the updates which were detected as applicable

- **NumberOfApplicableUpdates** The number of updates which were ultimately deemed applicable to the system after the detection process is complete

- **WUDeviceID** The unique device ID controlled by the software distribution client

- **RelatedCV** The previous Correlation Vector that was used before swapping with a new one

- **EventScenario** Indicates the purpose of sending this event - whether because the software distribution just started checking for content, or whether it was cancelled, succeeded, or failed

- **EventInstanceID** A globally unique identifier for event instance

- **DeviceModel** The device's model as defined in system bios

- **BiosName** The name of the device's system bios

- **BIOSVendor** The vendor of the device's system bios

- **BiosVersion** The version of the device's system bios

- **BiosReleaseDate** The release date of the device's system bios

- **SystemBIOSMajorRelease** The major release version of the device's system system

- **SystemBIOSMinorRelease** The minor release version of the device's system system

- **BiosFamily** The device's family as defined in system bios

- **BiosSKUNumber** The device's SKU as defined in system bios

- **ClientVersion** The version number of the software distribution client

- **ProcessName** The process name of the caller who initiated API calls, in the event where CallerApplicationName was not provided

- **ServiceGuid** An ID which represents which service the software distribution client is connecting to (Windows Update, Microsoft Store, etc.)

- **StatusCode** Indicates the result code of the event (success, cancellation, failure code HResult)

- **ExtendedStatusCode** Secondary status code for certain scenarios where StatusCode wasn't specific enough

- **FlightRing** The ring (speed of getting builds) that a device is on if participating in flighting (pre-release builds).

- **FlightBranch** The branch that a device is on if participating in flighting (pre-release builds).

- **ShippingMobileOperator** The mobile operator that a device shipped on.

- **CurrentMobileOperator** The mobile operator the device is currently connected to.

- **HomeMobileOperator** The mobile operator that the device was originally intended to work with

- **PhonePreviewEnabled** Indicates whether a phone was getting preview build, prior to flighting (pre-release builds) being introduced.

- **ActivityMatchingId** Contains a unique ID identifying a single CheckForUpdates session from initialization to completion
- **SyncType** Describes the type of scan the event was
- **IPVersion** Indicates whether the download took place over IPv4 or IPv6
- **NumberOfApplicationsCategoryScanEvaluated** The number of categories (apps) for which an app update scan checked
- **ScanDurationInSeconds** The number of seconds a scan took
- **ScanEnqueueTime** The number of seconds it took to initialize a scan
- **NumberOfLoop** The number of round trips the scan required
- **NumberOfUpdatesEvaluated** The total number of updates which were evaluated as a part of the scan
- **NumberOfNewUpdatesFromServiceSync** The number of updates which were seen for the first time in this scan
- **ServiceUrl** The environment URL a device is configured to scan with
- **IntentPFNs** Intended application-set metadata for atomic update scenarios.

SoftwareUpdateClientTelemetry.UpdateMetadataIntegrity

This event identifies whether updates have been tampered with and protects against man-in-the-middle attacks.

The following fields are available:

- **EventScenario** The purpose of this event, such as scan started, scan succeeded, or scan failed.
- **ServiceGuid** Identifies the service to which the software distribution client is connected, Example: Windows Update or Microsoft Store
- **MetadataIntegrityMode** The mode of the transport metadata integrity check. 0 = unknown; 1 = ignore; 2 = audit; 3 = enforce
- **StatusCode** The status code of the event.
- **ExtendedStatusCode** The secondary status code of the event.
- **RevisionId** The revision ID for a specific piece of content.
- **UpdateId** The update ID for a specific piece of content.
- **RevisionNumber** The revision number for a specific piece of content.
- **TimestampTokenId** The time this was created. It is encoded in a timestamp blob and will be zero if the token is malformed.
- **LeafCertId** Integral ID from the FragmentSigning data for certificate that failed.
- **SHA256OfLeafCertPublicKey** A base64 encoding of the hash of the Base64CertData in the FragmentSigning data of the leaf certificate.
- **MetadataSignature** A base64-encoded string of the signature associated with the update metadata (specified by revision ID).
- **SignatureAlgorithm** The hash algorithm for the metadata signature.
- **SHA256OfTimestampToken** A base64-encoded string of hash of the timestamp token blob.
- **ValidityWindowInDays** The validity window that's in effect when verifying the timestamp.
- **TimestampTokenCertThumbprint** The thumbprint of the encoded timestamp token.
- **RawMode** The raw unparsed mode string from the SLS response. This field is null if not applicable.
- **RawValidityWindowInDays** The raw unparsed validity window string in days of the timestamp token. This field is null if not applicable.
- **SHA256OfLeafCerData** A base64 encoding of the hash for the Base64CerData in the FragmentSigning data of the leaf certificate.
- **ListOfSHA256OfIntermediateCerData** A semicolon delimited list of base64 encoding of hashes for the Base64CerData in the FragmentSigning data of an intermediate certificate.
- **EndpointUrl** The endpoint URL where the device obtains update metadata. This is used to distinguish between

test, staging, and production environments.

- **SLSPrograms** A test program to which a device may have opted in. Example: Insider Fast

Update events

Update360Telemetry.UpdateAgent_DownloadRequest

This event sends data during the download request phase of updating Windows.

The following fields are available:

- **ErrorCode** The error code returned for the current download request phase.
- **PackageCountTotal** Total number of packages needed.
- **PackageCountRequired** Number of required packages requested.
- **PackageCountOptional** Number of optional packages requested.
- **ObjectId** Unique value for each Update Agent mode.
- **SessionId** Unique value for each Update Agent mode attempt.
- **ScenarioId** The scenario ID. Example: MobileUpdate, DesktopLanguagePack, DesktopFeatureOnDemand, or DesktopDriverUpdate
- **RelatedCV** Correlation vector value generated from the latest USO scan.
- **Result** Result of the download request phase of update.
- **PackageSizeCanonical** Size of canonical packages in bytes
- **PackageSizeDiff** Size of diff packages in bytes
- **PackageSizeExpress** Size of express packages in bytes
- **FlightId** Unique ID for each flight.
- **UpdateId** Unique ID for each update.
- **PackageCountTotalCanonical** Total number of canonical packages.
- **PackageCountTotalDiff** Total number of diff packages.
- **PackageCountTotalExpress** Total number of express packages.
- **RangeRequestState** Represents the state of the download range request.
- **DeletedCorruptFiles** Indicates if UpdateAgent found any corrupt payload files and whether the payload was deleted.

Update360Telemetry.UpdateAgent_Initialize

This event sends data during the initialize phase of updating Windows.

The following fields are available:

- **ErrorCode** The error code returned for the current initialize phase.
- **SessionData** Contains instructions to update agent for processing FODs and DUICs (Null for other scenarios).
- **UpdateId** Unique ID for each update.
- **FlightId** Unique ID for each flight.
- **FlightMetadata** Contains the FlightId and the build being flighted.
- **ObjectId** Unique value for each Update Agent mode.
- **SessionId** Unique value for each Update Agent mode attempt .
- **ScenarioId** The scenario ID. Example: MobileUpdate, DesktopLanguagePack, DesktopFeatureOnDemand, or DesktopDriverUpdate
- **RelatedCV** Correlation vector value generated from the latest USO scan.
- **Result** Result of the initialize phase of update. 0 = Succeeded, 1 = Failed, 2 = Cancelled, 3 = Blocked, 4 = BlockCancelled

Update360Telemetry.UpdateAgent_Install

This event sends data during the install phase of updating Windows.

The following fields are available:

- **ErrorCode** The error code returned for the current install phase.
- **ObjectId** Unique value for each Update Agent mode.
- **SessionId** Unique value for each Update Agent mode attempt.
- **ScenarioId** The scenario ID. Example: MobileUpdate, DesktopLanguagePack, DesktopFeatureOnDemand, or DesktopDriverUpdate
- **RelatedCV** Correlation vector value generated from the latest scan.
- **Result** Result of the install phase of update. 0 = Succeeded 1 = Failed, 2 = Cancelled, 3 = Blocked, 4 = BlockCancelled
- **FlightId** Unique ID for each flight.
- **UpdateId** Unique ID for each update.

Update360Telemetry.UpdateAgent_ModeStart

This event sends data for the start of each mode during the process of updating Windows.

The following fields are available:

- **Mode** Indicates that the Update Agent mode that has started. 1 = Initialize, 2 = DownloadRequest, 3 = Install, 4 = Commit
- **ObjectId** Unique value for each Update Agent mode.
- **SessionId** Unique value for each Update Agent mode attempt.
- **ScenarioId** The scenario ID. Example: MobileUpdate, DesktopLanguagePack, DesktopFeatureOnDemand, or DesktopDriverUpdate
- **RelatedCV** The correlation vector value generated from the latest scan.
- **FlightId** Unique ID for each flight.
- **UpdateId** Unique ID for each update.

Update360Telemetry.UpdateAgent_SetupBoxLaunch

This event sends data during the launching of the setup box when updating Windows.

The following fields are available:

- **Quiet** Indicates whether setup is running in quiet mode. 0 = false 1 = true
- **ObjectId** Unique value for each Update Agent mode.
- **SessionId** Unique value for each Update Agent mode attempt.
- **ScenarioId** The scenario ID. Example: MobileUpdate, DesktopLanguagePack, DesktopFeatureOnDemand, or DesktopDriverUpdate
- **RelatedCV** Correlation vector value generated from the latest scan.
- **FlightId** Unique ID for each flight.
- **UpdateId** Unique ID for each update.
- **SetupMode** Setup mode 1 = predownload, 2 = install, 3 = finalize
- **SandboxSize** The size of the sandbox folder on the device.

Upgrade events

Setup360Telemetry.Downlevel

This event sends data indicating that the device has invoked the downlevel phase of the upgrade. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **ClientId** If using Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, the default value is Media360, but it can be overwritten by the caller to a unique value.
- **InstanceId** A unique GUID that identifies each instance of setuphost.exe.
- **ReportId** In the Windows Update scenario, this is the updateID that is passed to Setup. In media setup, this is the GUID for the install.wim.
- **Wuld** This is the Windows Update Client ID. In the Windows Update scenario, this is the same as the clientId.
- **TestId** A string that uniquely identifies a group of events.
- **State** Exit state of given Setup360 run. Example: succeeded, failed, blocked, cancelled
- **HostOsSkuName** The operating system edition which is running Setup360 instance (downlevel OS).
- **HostOSBuildNumber** The build number of the downlevel OS.
- **Setup360Scenario** The Setup360 flow type. Example: Boot, Media, Update, MCT
- **Setup360Mode** The phase of Setup360. Example: Predownload, Install, Finalize, Rollback
- **Setup360Result** The result of Setup360. It's an HRESULT error code that can be used to diagnose errors.
- **Setup360Extended** Extension of result - more granular information about phase/action when the potential failure happened
- **SetupVersionBuildNumber** The build number of Setup360 (build number of the target OS).

Setup360Telemetry.Finalize

This event sends data indicating that the device has invoked the finalize phase of the upgrade, to help keep Windows up-to-date.

The following fields are available:

- **ClientId** With Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, default value is Media360, but can be overwritten by the caller to a unique value.
- **InstanceId** A unique GUID that identifies each instance of setuphost.exe
- **ReportId** With Windows Update, this is the updateID that is passed to Setup. In media setup, this is the GUID for the install.wim.
- **Wuld** This is the Windows Update Client ID. With Windows Update, this is the same as the clientId.
- **TestId** A string to uniquely identify a group of events.
- **State** The exit state of a Setup360 run. Example: succeeded, failed, blocked, cancelled
- **HostOsSkuName** The OS edition which is running Setup360 instance (previous OS).
- **HostOSBuildNumber** The build number of the previous OS.
- **Setup360Scenario** The Setup360 flow type. Example: Boot, Media, Update, MCT
- **Setup360Mode** The phase of Setup360. Example: Predownload, Install, Finalize, Rollback
- **Setup360Result** The result of Setup360. This is an HRESULT error code that is used to diagnose errors.
- **Setup360Extended** Extension of result - more granular information about phase/action when the potential failure happened
- **SetupVersionBuildNumber** The build number of Setup360 (build number of target OS).

Setup360Telemetry.OsUninstall

The event sends data regarding OS updates and upgrades from Windows 7, Windows 8, and Windows 10. Specifically, the Setup360Telemetry.OSUninstall indicates the outcome of an OS uninstall.

The following fields are available:

- **ClientId** For Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, default value is Media360, but can be overwritten by the caller to a unique value.
- **InstanceId** A unique GUID that identifies each instance of setuphost.exe.
- **ReportId** For Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, this is the GUID for the install.wim.

- **WuId** Windows Update client ID.
- **TestId** A string to uniquely identify a group of events.
- **State** Exit state of a Setup360 run. Example: succeeded, failed, blocked, cancelled
- **HostOsSkuName** The OS edition which is running the Setup360 instance (previous OS).
- **HostOSBuildNumber** The build number of the previous OS.
- **Setup360Scenario** The Setup360 flow type. Example: Boot, Media, Update, MCT
- **Setup360Mode** The phase of Setup360. Example: Predownload, Install, Finalize, Rollback
- **Setup360Result** The result of Setup360. This is an HRESULT error code that is used to diagnose errors.
- **Setup360Extended** Extension of result - more granular information about phase/action when the potential failure happened
- **SetupVersionBuildNumber** The build number of Setup360 (build number of target OS).

Setup360Telemetry.PostRebootInstall

This event sends data indicating that the device has invoked the postrebootinstall phase of the upgrade, to help keep Windows up-to-date.

The following fields are available:

- **ClientId** With Windows Update, this is the Windows Update client ID that is passed to Setup. In Media setup, the default value is Media360, but can be overwritten by the caller to a unique value.
- **InstanceId** A unique GUID that identifies each instance of setuphost.exe.
- **ReportId** With Windows Update, this is the updateID that is passed to Setup. In media setup, this is the GUID for the install.wim.
- **WuId** This is the Windows Update Client ID. With Windows Update, this is the same as ClientId.
- **TestId** A string to uniquely identify a group of events.
- **State** The exit state of a Setup360 run. Example: succeeded, failed, blocked, cancelled
- **HostOsSkuName** The OS edition which is running Setup360 instance (previous OS).
- **HostOSBuildNumber** The build number of the previous OS.
- **Setup360Scenario** The Setup360 flow type. Example: Boot, Media, Update, MCT
- **Setup360Mode** The phase of Setup360. Example: Predownload, Install, Finalize, Rollback
- **Setup360Result** The result of Setup360. This is an HRESULT error code that's used to diagnose errors.
- **Setup360Extended** Extension of result - more granular information about phase/action when the potential failure happened
- **SetupVersionBuildNumber** The build number of Setup360 (build number of target OS).

Setup360Telemetry.PreDownloadQuiet

This event sends data indicating that the device has invoked the predownload quiet phase of the upgrade, to help keep Windows up to date.

The following fields are available:

- **ClientId** Using Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, default value is Media360, but can be overwritten by the caller to a unique value.
- **InstanceId** A unique GUID that identifies each instance of setuphost.exe
- **ReportId** Using Windows Update, this is the updateID that is passed to Setup. In media setup, this is the GUID for the install.wim.
- **WuId** This is the Windows Update Client ID. Using Windows Update, this is the same as the clientId.
- **TestId** A string to uniquely identify a group of events.
- **State** The exit state of a Setup360 run. Example: succeeded, failed, blocked, canceled
- **HostOsSkuName** The OS edition which is running Setup360 instance (previous operating system).
- **HostOSBuildNumber** The build number of the previous OS.

- **Setup360Scenario** The Setup360 flow type. Example: Boot, Media, Update, MCT
- **Setup360Mode** The phase of Setup360. Example: Predownload, Install, Finalize, Rollback
- **Setup360Result** The result of Setup360. This is an HRESULT error code that is used to diagnose errors.
- **Setup360Extended** Extension of result - more granular information about phase/action when the potential failure happened
- **SetupVersionBuildNumber** The build number of Setup360 (build number of target OS).

Setup360Telemetry.PreDownloadUX

The event sends data regarding OS updates and upgrades from Windows 7, Windows 8, and Windows 10. Specifically, the Setup360Telemetry.PredownloadUX indicates the outcome of the PredownloadUX portion of the update process.

The following fields are available:

- **ClientId** For Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, default value is Media360, but can be overwritten by the caller to a unique value.
- **InstanceId** Unique GUID that identifies each instance of setuphost.exe.
- **ReportId** For Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, this is the GUID for the install.wim.
- **Wuld** Windows Update client ID.
- **TestId** A string to uniquely identify a group of events.
- **State** The exit state of the Setup360 run. Example: succeeded, failed, blocked, cancelled
- **HostOsSkuName** The OS edition which is running the Setup360 instance (previous operating system).
- **HostOSBuildNumber** The build number of the previous operating system.
- **Setup360Scenario** The Setup360 flow type. Examplle: Boot, Media, Update, MCT
- **Setup360Mode** The phase of Setup360. Example: Predownload, Install, Finalize, Rollback
- **Setup360Result** The result of Setup360. This is an HRESULT error code that can be used to diagnose errors.
- **Setup360Extended** Extension of result - more granular information about phase/action when the potential failure happened
- **SetupVersionBuildNumber** The build number of Setup360 (build number of the target OS).

Setup360Telemetry.PreInstallQuiet

This event sends data indicating that the device has invoked the preinstall quiet phase of the upgrade, to help keep Windows up to date.

The following fields are available:

- **ClientId** With Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, default value is Media360, but can be overwritten by the caller to a unique value.
- **InstanceId** A unique GUID that identifies each instance of setuphost.exe
- **ReportId** With Windows Update, this is the updateID that is passed to Setup. In media setup, this is the GUID for the install.wim.
- **Wuld** This is the Windows Update Client ID. With Windows Update, this is the same as the clientId.
- **TestId** A string to uniquely identify a group of events.
- **State** The exit state of a Setup360 run. Example: succeeded, failed, blocked, cancelled
- **HostOsSkuName** The OS edition which is running Setup360 instance (previous OS).
- **HostOSBuildNumber** The build number of the previous OS.
- **Setup360Scenario** Setup360 flow type (Boot, Media, Update, MCT)
- **Setup360Mode** The phase of Setup360. Example: Predownload, Install, Finalize, Rollback etc.
- **Setup360Result** The result of Setup360. This is an HRESULT error code that can be used to diagnose errors.
- **Setup360Extended** Extension of result - more granular information about phase/action when the potential

failure happened

- **SetupVersionBuildNumber** The build number of Setup360 (build number of target OS).

Setup360Telemetry.PreInstallUX

This event sends data regarding OS updates and upgrades from Windows 7, Windows 8, and Windows 10. Specifically, the Setup360Telemetry.PreinstallUX indicates the outcome of the PreinstallUX portion of the update process.

The following fields are available:

- **ClientId** For Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, default value is Media360, but can be overwritten by the caller to a unique value.
- **InstanceId** A unique GUID that identifies each instance of setuphost.exe.
- **ReportId** For Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, this is the GUID for the install.wim.
- **WuId** Windows Update client ID.
- **TestId** A string to uniquely identify a group of events.
- **State** The exit state of a Setup360 run. Example: succeeded, failed, blocked, cancelled
- **HostOsSkuName** The OS edition which is running the Setup360 instance (previous OS).
- **HostOSBuildNumber** The build number of the previous OS.
- **Setup360Scenario** The Setup360 flow type, Example: Boot, Media, Update, MCT
- **Setup360Mode** The phase of Setup360. Example: Predownload, Install, Finalize, Rollback
- **Setup360Result** The result of Setup360. This is an HRESULT error code that is used to diagnose errors.
- **Setup360Extended** Extension of result - more granular information about phase/action when the potential failure happened
- **SetupVersionBuildNumber** The build number of Setup360 (build number of target OS).

Setup360Telemetry.Setup360

This event sends data about OS deployment scenarios, to help keep Windows up-to-date.

The following fields are available:

- **InstanceId** Retrieves a unique identifier for each instance of a setup session.
- **ReportId** Retrieves the report ID.
- **FlightData** Specifies a unique identifier for each group of Windows Insider builds.
- **ScenarioId** Retrieves the deployment scenario.
- **FieldName** Retrieves the data point.
- **Value** Retrieves the value associated with the corresponding FieldName.
- **ClientId** Retrieves the upgrade ID: Upgrades via Windows Update - specifies the WU clientID. All other deployment - static string.

Setup360Telemetry.UnexpectedEvent

This event sends data indicating that the device has invoked the unexpected event phase of the upgrade, to help keep Windows up to date.

The following fields are available:

- **ClientId** With Windows Update, this will be the Windows Update client ID that is passed to Setup. In Media setup, default value is Media360, but can be overwritten by the caller to a unique value.
- **InstanceId** A unique GUID that identifies each instance of setuphost.exe
- **ReportId** With Windows Update, this is the updateID that is passed to Setup. In media setup, this is the GUID for the install.wim.
- **WuId** This is the Windows Update Client ID. With Windows Update, this is the same as the clientId.

- **TestId** A string to uniquely identify a group of events.
- **State** The exit state of a Setup360 run. Example: succeeded, failed, blocked, cancelled
- **HostOsSkuName** The OS edition which is running Setup360 instance (previous OS).
- **HostOSBuildNumber** The build number of the previous OS.
- **Setup360Scenario** The Setup360 flow type. Example: Boot, Media, Update, MCT
- **Setup360Mode** The phase of Setup360. Example: Predownload, Install, Finalize, Rollback
- **Setup360Result** The result of Setup360. This is an HRESULT error code that can be used used to diagnose errors.
- **Setup360Extended** Extension of result - more granular information about phase/action when the potential failure happened
- **SetupVersionBuildNumber** The build number of Setup360 (build number of target OS).

Windows Error Reporting events

Microsoft.Windows.WERVertical.OSCrash

This event sends binary data from the collected dump file wheneveer a bug check occurs, to help keep Windows up to date. The is the OneCore version of this event.

The following fields are available:

- **ReportId** WER Report Id associated with this bug check (used for finding the corresponding report archive in Watson).
- **BugCheckCode** Uint64 "bugcheck code" that identifies a proximate cause of the bug check.
- **BugCheckParameter1** Uint64 parameter providing additional information.
- **BootId** Uint32 identifying the boot number for this device.
- **BugCheckParameter2** Uint64 parameter providing additional information.
- **BugCheckParameter4** Uint64 parameter providing additional information.
- **BugCheckParameter3** Uint64 parameter providing additional information.
- **IsValidDumpFile** True if the dump file is valid for the debugger, false otherwise
- **DumpFileSize** Size of the dump file
- **DumpFileAttributes** Codes that identify the type of data contained in the dump file

Microsoft Store events

Microsoft.Windows.StoreAgent.Telemetry.AbortedInstallation

This event is sent when an installation or update is canceled by a user or the system and is used to help keep Windows Apps up to date and secure.

The following fields are available:

- **PFN** The product family name of the product being installed.
- **ClientAppId** The identity of the app that initiated this operation.
- **HResult** The result code of the last action performed before this operation.
- **IsUpdate** Flag indicating if this is an update.
- **AttemptNumber** Number of retry attempts before it was canceled.
- **CategoryId** The Item Category ID.
- **ProductId** The identity of the package or packages being installed.
- **IsInteractive** Was this requested by a user?
- **IsRemediation** Was this a remediation install?
- **BundleId** The Item Bundle ID.

- **IsMandatory** Was this a mandatory update?
- **SystemAttemptNumber** The total number of automatic attempts at installation before it was canceled.
- **UserAttemptNumber** The total number of user attempts at installation before it was canceled.
- **IsRestore** Is this automatically restoring a previously acquired product?
- **ParentBundleId** The product ID of the parent (if this product is part of a bundle).
- **IsBundle** Is this a bundle?
- **WUContentId** The Windows Update content ID
- **AggregatedPackageFullNames** The names of all packages to be downloaded and installed.

Microsoft.Windows.StoreAgent.Telemetry.BeginGetInstalledContentIds

This event is sent when an inventory of the apps installed is started to determine whether updates for those apps are available. It's used to help keep Windows up-to-date and secure.

Microsoft.Windows.StoreAgent.Telemetry.BeginUpdateMetadataPrepare

This event is sent when the Store Agent cache is refreshed with any available package updates. It's used to help keep Windows up-to-date and secure.

Microsoft.Windows.StoreAgent.Telemetry.CancelInstallation

This event is sent when an app update or installation is canceled while in interactive mode. This can be canceled by the user or the system. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **IsInteractive** Was this requested by a user?
- **AttemptNumber** Total number of installation attempts.
- **BundleId** The identity of the Windows Insider build that is associated with this product.
- **PreviousHResult** The previous HResult code.
- **ClientAppId** The identity of the app that initiated this operation.
- **CategoryId** The identity of the package or packages being installed.
- **PFN** The name of all packages to be downloaded and installed.
- **ProductId** The name of the package or packages requested for installation.
- **IsUpdate** Is this a product update?
- **IsRemediation** Is this repairing a previous installation?
- **RelatedCV** Correlation Vector of a previous performed action on this product.
- **PreviousInstallState** Previous installation state before it was canceled.
- **IsMandatory** Is this a mandatory update?
- **SystemAttemptNumber** Total number of automatic attempts to install before it was canceled.
- **UserAttemptNumber** Total number of user attempts to install before it was canceled.
- **IsRestore** Is this an automatic restore of a previously acquired product?
- **ParentBundleId** The product ID of the parent (if this product is part of a bundle).
- **IsBundle** Is this a bundle?
- **WUContentId** The Windows Update content ID
- **AggregatedPackageFullNames** The names of all package or packages to be downloaded and installed.

Microsoft.Windows.StoreAgent.Telemetry.CompleteInstallOperationRequest

This event is sent after the app installations or updates. It's used to help keep Windows up-to-date and secure

The following fields are available:

- **IsBundle** Is this a bundle?
- **ProductId** The Store Product ID of the product being installed.

- **SkuId** Specific edition of the item being installed.
- **CatalogId** The Store Product ID of the app being installed.
- **PackageFamilyName** The name of the package being installed.
- **HResult** HResult code of the action being performed.

Microsoft.Windows.StoreAgent.Telemetry.EndAcquireLicense

This event is sent after the license is acquired when a product is being installed. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **PFN** Product Family Name of the product being installed.
- **HResult** HResult code to show the result of the operation (success/failure).
- **ProductId** The Store Product ID for the product being installed.
- **IsInteractive** Did the user initiate the installation?
- **CategoryId** The identity of the package or packages being installed.
- **ClientAppId** The identity of the app that initiated this operation.
- **IsRemediation** Is this repairing a previous installation?
- **UpdateId** The update ID (if this is an update)
- **AttemptNumber** The total number of attempts to acquire this product.
- **IsUpdate** Is this an update?
- **IsMandatory** Is this a mandatory update?
- **SystemAttemptNumber** The number of attempts by the system to acquire this product.
- **UserAttemptNumber** The number of attempts by the user to acquire this product
- **IsRestore** Is this happening after a device restore?
- **IsBundle** Is this a bundle?
- **WUContentId** The Windows Update content ID
- **ParentBundledId** The product's parent bundle ID.
- **AggregatedPackageFullNames** Includes a set of package full names for each app that is part of an atomic set.

Microsoft.Windows.StoreAgent.Telemetry.EndDownload

This event happens during the app update or installation when content is being downloaded at the end of the process to report success or failure. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **PFN** The Product Family Name of the app being download.
- **IsRemediation** Is this repairing a previous installation?
- **DownloadSize** The total size of the download.
- **ClientAppId** The identity of the app that initiated this operation.
- **CategoryId** The identity of the package or packages being installed.
- **IsUpdate** Is this an update?
- **HResult** The result code of the last action performed.
- **IsInteractive** Is this initiated by the user?
- **AttemptNumber** Number of retry attempts before it was canceled.
- **BundleId** The identity of the Windows Insider build associated with this product.
- **ProductId** The Store Product ID for the product being installed.
- **IsMandatory** Is this a mandatory installation?
- **SystemAttemptNumber** The number of attempts by the system to download.

- **UserAttemptNumber** The number of attempts by the user to download.
- **IsRestore** Is this a restore of a previously acquired product?
- **ParentBundleId** The parent bundle ID (if it's part of a bundle).
- **IsBundle** Is this a bundle?
- **WUContentId** The Windows Update content ID.
- **ExtendedHResult** Any extended HResult error codes.
- **AggregatedPackageFullNames** The name of all packages to be downloaded and installed.

Microsoft.Windows.StoreAgent.Telemetry.EndFrameworkUpdate

This event happens when an app update requires an updated Framework package and the process starts to download it. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **HResult** The result code of the last action performed before this operation.

Microsoft.Windows.StoreAgent.Telemetry.EndGetInstalledContentIds

This event is sent after sending the inventory of the products installed to determine whether updates for those products are available. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **HResult** The result code of the last action performed before this operation.

Microsoft.Windows.StoreAgent.Telemetry.EndInstall

This event is sent after a product has been installed. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **BundleId** The identity of the build associated with this product.
- **PFN** Product Family Name of the product being installed.
- **ClientAppId** The identity of the app that initiated this operation.
- **CategoryId** The identity of the package or packages being installed.
- **ProductId** The Store Product ID for the product being installed.
- **AttemptNumber** The number of retry attempts before it was canceled.
- **HResult** The result code of the last action performed.
- **IsRemediation** Is this repairing a previous installation?
- **IsInteractive** Is this an interactive installation?
- **IsUpdate** Is this an update?
- **IsMandatory** Is this a mandatory installation?
- **SystemAttemptNumber** The total number of system attempts.
- **UserAttemptNumber** The total number of user attempts.
- **IsRestore** Is this automatically restoring a previously acquired product?
- **ParentBundleId** The product ID of the parent (if this product is part of a bundle).
- **IsBundle** Is this a bundle?
- **WUContentId** The Windows Update content ID
- **ExtendedHResult** The extended HResult error code.
- **AggregatedPackageFullNames** The names of all packages to be downloaded and installed.

Microsoft.Windows.StoreAgent.Telemetry.EndScanForUpdates

This event is sent after a scan for product updates to determine if there are packages to install. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **HResult** The result code of the last action performed.
- **IsApplicability** Is this request to only check if there are any applicable packages to install?
- **IsInteractive** Is this user requested?
- **ClientAppId** The identity of the app that initiated this operation.
- **IsOnline** Is the request doing an online check?

Microsoft.Windows.StoreAgent.Telemetry.EndSearchUpdatePackages

This event is sent after searching for update packages to install. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **IsRemediation** Is this repairing a previous installation?
- **IsUpdate** Is this an update?
- **ClientAppId** The identity of the app that initiated this operation.
- **HResult** The result code of the last action performed.
- **ProductId** The Store Product ID for the product being installed.
- **AttemptNumber** The total number of retry attempts before it was canceled.
- **IsInteractive** Is this user requested?
- **PFN** The name of the package or packages requested for install.
- **BundleId** The identity of the build associated with this product.
- **CategoryId** The identity of the package or packages being installed.
- **IsMandatory** Is this a mandatory update?
- **SystemAttemptNumber** The total number of system attempts.
- **UserAttemptNumber** The total number of user attempts.
- **IsRestore** Is this restoring previously acquired content?
- **ParentBundleId** The product ID of the parent (if this product is part of a bundle).
- **IsBundle** Is this a bundle?
- **WUContentId** The Windows Update content ID
- **AggregatedPackageFullNames** The names of all packages to be downloaded and installed.

Microsoft.Windows.StoreAgent.Telemetry.EndStageUserData

This event is sent between download and installation to see if there is app data that needs to be restored from the cloud. It's used to keep Windows up-to-date and secure.

The following fields are available:

- **IsInteractive** Is this user requested?
- **PFN** The name of the package or packages requested for install.
- **IsUpdate** Is this an update?
- **CategoryId** The identity of the package or packages being installed.
- **HResult** The result code of the last action performed.
- **AttemptNumber** The total number of retry attempts before it was canceled.
- **ProductId** The Store Product ID for the product being installed.
- **BundleId** The identity of the build associated with this product.
- **IsRemediation** Is this repairing a previous installation?
- **ClientAppId** The identity of the app that initiated this operation.
- **IsMandatory** Is this a mandatory update?
- **SystemAttemptNumber** The total number of system attempts.

- **UserAttemptNumber** The total number of system attempts.
- **IsRestore** Is this restoring previously acquired content?
- **ParentBundleId** The product ID of the parent (if this product is part of a bundle).
- **IsBundle** Is this a bundle?
- **WUContentId** The Windows Update content ID
- **AggregatedPackageFullNames** The name of all packages to be downloaded and installed.

Microsoft.Windows.StoreAgent.Telemetry.EndUpdateMetadataPrepare

This event happens after a scan for available app updates. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **HResult** The result code of the last action performed.

Microsoft.Windows.StoreAgent.Telemetry.FulfillmentComplete

This event is sent at the end of an app install or update and is used to track the very end of the install or update process.

The following fields are available:

- **ProductId** The product ID of the app that is being updated or installed.
- **PFN** The Package Family Name of the app that is being installed or updated.
- **FailedRetry** Was the installation or update retry successful?
- **HResult** The HResult code of the operation.

Microsoft.Windows.StoreAgent.Telemetry.FulfillmentInitiate

This event is sent at the beginning of an app install or update and is used to track the very beginning of the install or update process.

The following fields are available:

- **ProductId** The product ID of the app that is being updated or installed.
- **PFN** The Package Family Name of the app that is being installed or updated.

Microsoft.Windows.StoreAgent.Telemetry.InstallOperationRequest

This event happens at the beginning of the install process when an app update or new app is installed. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **CatalogId** If this product is from a private catalog, the Store Product ID for the product being installed.
- **BundleId** The identity of the build associated with this product.
- **SkuId** Specific edition ID being installed.
- **ProductId** The Store Product ID for the product being installed.
- **VolumePath** The disk path of the installation.

Microsoft.Windows.StoreAgent.Telemetry.PauseInstallation

This event is sent when a product install or update is paused either by a user or the system. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **RelatedCV** Correlation Vector of a previous performed action on this product.
- **IsRemediation** Is this repairing a previous installation?
- **PreviousHResult** The result code of the last action performed before this operation.
- **ProductId** The Store Product ID for the product being installed.

- **IsUpdate** Is this an update?
- **PreviousInstallState** Previous state before the installation or update was paused.
- **CategoryId** The identity of the package or packages being installed.
- **ClientAppId** The identity of the app that initiated this operation.
- **AttemptNumber** The total number of retry attempts before it was canceled.
- **IsInteractive** Is this user requested?
- **BundleId** The identity of the build associated with this product.
- **PFN** The Product Full Name.
- **IsMandatory** Is this a mandatory update?
- **SystemAttemptNumber** The total number of system attempts.
- **UserAttemptNumber** The total number of user attempts.
- **IsRestore** Is this restoring previously acquired content?
- **ParentBundleId** The product ID of the parent (if this product is part of a bundle).
- **IsBundle** Is this a bundle?
- **WUContentId** The Windows Update content ID
- **AggregatedPackageFullNames** The names of all packages to be downloaded and installed.

Microsoft.Windows.StoreAgent.Telemetry.ResumeInstallation

This event happens when a product install or update is resumed either by a user or the system. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **RelatedCV** Correlation Vector for the original install before it was resumed.
- **AttemptNumber** The number of retry attempts before it was canceled.
- **BundleId** The identity of the build associated with this product.
- **PreviousHResult** The previous HResult error code.
- **ClientAppId** The identity of the app that initiated this operation.
- **CategoryId** The identity of the package or packages being installed.
- **PFN** The name of the package or packages requested for install.
- **IsUpdate** Is this an update?
- **PreviousInstallState** Previous state before the installation was paused.
- **IsRemediation** Is this repairing a previous installation?
- **IsInteractive** Is this user requested?
- **ProductId** The Store Product ID for the product being installed.
- **IsMandatory** Is this a mandatory update?
- **SystemAttemptNumber** The total number of system attempts.
- **UserAttemptNumber** The total number of user attempts.
- **IsRestore** Is this restoring previously acquired content?
- **ParentBundleId** The product ID of the parent (if this product is part of a bundle).
- **IsBundle** Is this a bundle?
- **WUContentId** The Windows Update content ID
- **AggregatedPackageFullNames** The names of all packages to be downloaded and installed.
- **IsUserRetry** Did the user initiate the retry?
- **HResult** The result code of the last action performed before this operation.

Microsoft.Windows.StoreAgent.Telemetry.ResumeOperationRequest

This event happens when a product install or update is resumed by a user and on installation retries. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **ProductId** The Store Product ID for the product being installed.

Microsoft.Windows.StoreAgent.Telemetry.SearchForUpdateOperationRequest

This event is sent when searching for update packages to install. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **ProductId** The Store Product ID for the product being installed.
- **SkuId** Specfic edition of the app being updated.
- **CatalogId** The Store Product ID for the product being installed.

Microsoft.Windows.StoreAgent.Telemetry.UpdateAppOperationRequest

This event happens an app for a user needs to be updated. It's used to help keep Windows up-to-date and secure.

The following fields are available:

- **PFamN** The name of the product that is requested for update.

Windows Update Delivery Optimization events

Microsoft.OSG.DU.DeliveryOptClient.DownloadCanceled

This event describes when a download was canceled with Delivery Optimization. It's used to understand and address problems regarding downloads.

The following fields are available:

- **bytesFromIntPeers** The number of bytes received from peers not in the same LAN or in the same group.
- **fileID** The ID of the file being downloaded.
- **sessionID** The ID of the file download session.
- **scenarioID** The ID of the scenario.
- **bytesFromCDN** The number of bytes received from a CDN source.
- **updateID** The ID of the update being downloaded.
- **background** Is the download being done in the background?
- **bytesFromPeers** The number of bytes received from a peer in the same LAN.
- **clientTelId** A random number used for device sampling.
- **bytesFromGroupPeers** The number of bytes received from a peer in the same group.
- **errorCode** The error code that was returned.
- **doErrorCode** The Delivery Optimization error code that was returned.
- **cdnErrorCodes** A list of CDN connection errors since the last FailureCDNCommunication event.
- **cdnErrorCounts** The number of times each error in cdnErrorCodes was encountered.
- **experimentId** When running a test, this is used to correlate events that are part of the same test.
- **isVpn** Is the device connected to a Virtual Private Network?
- **usedMemoryStream** Did the download use memory streaming?

Microsoft.OSG.DU.DeliveryOptClient.DownloadCompleted

This event describes when a download has completed with Delivery Optimization. It's used to understand and address problems regarding downloads.

The following fields are available:

- **sessionID** The ID of the download session.

- **scenarioID** The ID of the scenario.
- **bytesFromIntPeers** The number of bytes received from peers not in the same LAN or in the same domain group.
- **updateID** The ID of the update being downloaded.
- **fileSize** The size of the file being downloaded.
- **bytesFromCDN** The number of bytes received from a CDN source.
- **fileID** The ID of the file being downloaded.
- **background** Is the download a background download?
- **bytesFromPeers** The number of bytes received from a peer in the same LAN.
- **totalTime** How long did the download take (in seconds)?
- **restrictedUpload** Is the upload restricted?
- **clientTelId** A random number used for device sampling.
- **bytesFromGroupPeers** The number of bytes received from a peer in the same domain group.
- **downloadMode** The download mode used for this file download session.
- **doErrorCode** The Delivery Optimization error code that was returned.
- **numPeers** The total number of peers used for this download.
- **cdnConnectionCount** The total number of connections made to the CDN.
- **lanConnectionCount** The total number of connections made to peers in the same LAN.
- **groupConnectionCount** The total number of connections made to peers in the same group.
- **internetConnectionCount** The total number of connections made to peers not in the same LAN or the same group.
- **cdnIp** The IP address of the source CDN.
- **downlinkBps** The maximum measured available download bandwidth (in bytes per second).
- **uplinkBps** The maximum measured available upload bandwidth (in bytes per second).
- **downlinkUsageBps** The download speed (in bytes per second).
- **uplinkUsageBps** The upload speed (in bytes per second).
- **totalTimeMs** Duration of the download (in seconds).
- **cdnErrorCodes** A list of CDN connection errors since the last FailureCDNCommunication event.
- **cdnErrorCounts** The number of times each error in cdnErrorCodes was encountered.
- **bytesRequested** The total number of bytes requested for download.
- **experimentId** When running a test, this is used to correlate with other events that are part of the same test.
- **isVpn** Is the device connected to a Virtual Private Network?
- **usedMemoryStream** Did the download use memory streaming?

Microsoft.OSG.DU.DeliveryOptClient.DownloadPaused

This event represents a temporary suspension of a download with Delivery Optimization. It's used to understand and address problems regarding downloads.

The following fields are available:

- **updateID** The ID of the update being paused.
- **errorCode** The error code that was returned.
- **scenarioID** The ID of the scenario.
- **background** Is the download a background download?
- **sessionID** The ID of the download session.
- **clientTelId** A random number used for device sampling.
- **reasonCode** The reason for pausing the download.
- **fileID** The ID of the file being paused.

- **experimentId** When running a test, this is used to correlate with other events that are part of the same test.
- **isVpn** Is the device connected to a Virtual Private Network?

Microsoft.OSG.DU.DeliveryOptClient.DownloadStarted

This event describes the start of a new download with Delivery Optimization. It's used to understand and address problems regarding downloads.

The following fields are available:

- **experimentId** When running a test, this is used to correlate with other events that are part of the same test.
- **errorCode** The error code that was returned.
- **doErrorCode** The Delivery Optimization error code that was returned.
- **peerID** The ID for this Delivery Optimization client.
- **doClientVersion** The version of the Delivery Optimization client.
- **jobID** The ID of the Windows Update job.
- **sessionID** The ID of the download session.
- **updateID** The ID of the update being downloaded.
- **scenarioID** The ID of the scenario.
- **fileID** The ID of the file being downloaded.
- **cdnUrl** The URL of the CDN.
- **filePath** The path where the file will be written.
- **groupID** ID for the group.
- **background** Is the download a background download?
- **downloadMode** The download mode used for this file download session.
- **minFileSizePolicy** The minimum content file size policy to allow the download using Peering.
- **diceRoll** The dice roll value used in sampling events.
- **deviceProfile** Identifies the usage or form factor. Example: Desktop or Xbox
- **isVpn** Is the device connected to a Virtual Private Network?
- **usedMemoryStream** Did the download use memory streaming?
- **minDiskSizePolicyEnforced** Is the minimum disk size enforced via policy?
- **minDiskSizeGB** The minimum disk size (in GB) required for Peering.
- **clientTelId** A random number used for device sampling.
- **costFlags** A set of flags representing network cost.

Microsoft.OSG.DU.DeliveryOptClient.FailureCdnCommunication

This event represents a failure to download from a CDN with Delivery Optimization. It's used to understand and address problems regarding downloads.

The following fields are available:

- **experimentId** When running a test, this is used to correlate with other events that are part of the same test.
- **fileID** The ID of the file being downloaded.
- **errorCode** The error code that was returned.
- **httpStatusCode** The HTTP status code returned by the CDN.
- **errorCount** The total number of times this error code was seen since the last FailureCdnCommunication event was encountered.
- **sessionID** The ID of the download session.
- **cdnUrl** The URL of the CDN.
- **cdnIp** The IP address of the CDN.
- **cdnHeaders** The HTTP headers returned by the CDN.

- **clientTelId** A random number used for device sampling.
- **isHeadRequest** The type of HTTP request that was sent to the CDN. Example: HEAD or GET
- **requestSize** The size of the range requested from the CDN.
- **responseSize** The size of the range response received from the CDN.

Microsoft.OSG.DU.DeliveryOptClient.JobError

This event represents a Windows Update job error. It allows for investigation of top errors.

The following fields are available:

- **jobID** The Windows Update job ID.
- **fileID** The ID of the file being downloaded.
- **errorCode** The error code returned.
- **clientTelId** A random number used for device sampling.
- **experimentId** When running a test, this is used to correlate with other events that are part of the same test.

Windows Update events

Microsoft.Windows.Update.DataMigrationFramework.DmfMigrationCompleted

This event sends data collected at the end of the Data Migration Framework (DMF) and parameters involved in its invocation, to help keep Windows up to date.

The following fields are available:

- **MigrationEndtime** A system timestamp of when the DMF migration completed.
- **UpdateIds** A collection of GUIDs for updates that are associated with the DMF session.
- **WuClientid** The GUID of the Windows Update client responsible for triggering the DMF migration.
- **MigrationDurationinmilliseconds** How long the DMF migration took (in milliseconds).
- **RevisionNumbers** A collection of revision numbers for the updates associated with the DMF session.

Microsoft.Windows.Update.DataMigrationFramework.DmfMigrationStarted

This event sends data collected at the beginning of the Data Migration Framework (DMF) and parameters involved in its invocation, to help keep Windows up to date.

The following fields are available:

- **UpdateIds** A collection of GUIDs identifying the upgrades that are running.
- **MigrationStarttime** The timestamp representing the beginning of the DMF migration.
- **MigrationOEMphases** The number of OEM-authored migrators scheduled to be ran by DMF for this upgrade.
- **WuClientid** The GUID of the Windows Update client invoking DMF.
- **MigrationMicrosoftphases** The number of Microsoft-authored migrators scheduled to be ran by DMF for this upgrade.
- **RevisionNumbers** A collection of the revision numbers associated with the UpdateIds.

Microsoft.Windows.Update.DataMigrationFramework.MigratorResult

This event sends DMF migrator data to help keep Windows up to date.

The following fields are available:

- **MigratorGuid** A GUID identifying the migrator that just completed.
- **RunDurationInSeconds** The time it took for the migrator to complete.
- **CurrentStep** This is the last step the migrator reported before returning a result. This tells us how far through the individual migrator the device was before failure.
- **MigratorName** The name of the migrator that just completed.

- **MigratorId** A GUID identifying the migrator that just completed.
- **ErrorCode** The result (as an HRESULT) of the migrator that just completed.
- **TotalSteps** Migrators report progress in number of completed steps against the total steps. This is the total number of steps.

Microsoft.Windows.Update.Orchestrator.CommitFailed

This events tracks when a device needs to restart after an update but did not.

The following fields are available:

- **wuDeviceid** The Windows Update device GUID.
- **errorCode** The error code that was returned.

Microsoft.Windows.Update.Orchestrator.Detection

This event sends launch data for a Windows Update scan to help keep Windows up to date.

The following fields are available:

- **wuDeviceid** Unique device ID used by Windows Update.
- **revisionNumber** Update revision number.
- **eventScenario** End to end update session ID, or indicates the purpose of sending this event - whether because the software distribution just started installing content, or whether it was cancelled, succeeded, or failed.
- **deferReason** Reason why the device could not check for updates.
- **detectionBlockreason** Reason for detection not completing.
- **interactive** Identifies if session is User Initiated.
- **updateId** Update ID.
- **detectionDeferreason** A log of deferral reasons for every update state.
- **flightID** A unique update ID.
- **updateScenarioType** The update session type.
- **errorCode** The returned error code.

Microsoft.Windows.Update.Orchestrator.Download

This event sends launch data for a Windows Update download to help keep Windows up to date.

The following fields are available:

- **detectionDeferreason** Reason for download not completing
- **wuDeviceid** Unique device ID used by Windows Update.
- **interactive** Identifies if session is user initiated.
- **revisionNumber** Update revision number.
- **deferReason** Reason for download not completing
- **updateId** Update ID.
- **eventScenario** End to end update session ID.
- **errorCode** An error code represented as a hexadecimal value
- **flightID** Unique update ID.
- **updateScenarioType** The update session type.

Microsoft.Windows.Update.Orchestrator.FlightInapplicable

This event sends data on whether the update was applicable to the device, to help keep Windows up to date.

The following fields are available:

- **updateId** Unique Update ID
- **revisionNumber** Revision Number of the Update

- **UpdateStatus** Integer that describes Update state
- **EventPublishedTime** time that the event was generated
- **wuDeviceid** Unique Device ID
- **flightID** Unique Update ID
- **updateScenarioType** The update session type.

Microsoft.Windows.Update.Orchestrator.InitiatingReboot

This event sends data about an Orchestrator requesting a reboot from power management to help keep Windows up to date.

The following fields are available:

- **revisionNumber** Revision number of the update.
- **EventPublishedTime** Time of the event.
- **updateId** Update ID.
- **wuDeviceid** Unique device ID used by Windows Update.
- **flightID** Unique update ID
- **interactive** Indicates the reboot initiation stage of the update process was entered as a result of user action or not.
- **rebootOutsideOfActiveHours** Indicates the timing that the reboot was to occur to ensure the correct update process and experience is provided to keep Windows up to date.
- **uxRebootstate** Indicates the exact state of the user experience at the time the required reboot was initiated to ensure the correct update process and experience is provided to keep Windows up to date.
- **updateScenarioType** The update session type.

Microsoft.Windows.Update.Orchestrator.Install

This event sends launch data for a Windows Update install to help keep Windows up to date.

The following fields are available:

- **eventScenario** End to end update session ID.
- **deferReason** Reason for install not completing.
- **interactive** Identifies if session is user initiated.
- **wuDeviceid** Unique device ID used by Windows Update.
- **batteryLevel** Current battery capacity in mWh or percentage left.
- **installCommitfailedtime** The time it took for a reboot to happen but the upgrade failed to progress.
- **errorCode** The error code reppresented by a hexadecimal value.
- **updateId** Update ID.
- **revisionNumber** Update revision number.
- **flightID** Unique update ID
- **installRebootinitiatetime** The time it took for a reboot to be attempted.
- **flightUpdate** Flight update
- **minutesToCommit** The time it took to install updates.
- **ForcedRebootReminderSet** A boolean value that indicates if a forced reboot will happen for updates.
- **rebootOutsideOfActiveHours** Indicates the timing that the reboot was to occur to ensure the correct update process and experience is provided to keep Windows up to date.
- **uxRebootstate** Indicates the exact state of the user experience at the time the required reboot was initiated to ensure the correct update process and experience is provided to keep Windows up to date.
- **updateScenarioType** The update session type.

Microsoft.Windows.Update.Orchestrator.PostInstall

This event sends data about lite stack devices (mobile, IOT, anything non-PC) immediately before data migration is launched to help keep Windows up to date.

The following fields are available:

- **wuDeviceid** Unique device ID used by Windows Update.
- **eventScenario** End to end update session ID.
- **sessionType** Interactive vs. Background.
- **bundleRevisionnumber** Bundle revision number.
- **batteryLevel** Current battery capacity in mWh or percentage left.
- **bundleId** Update grouping ID.
- **errorCode** Hex code for the error message, to allow lookup of the specific error.
- **flightID** Unique update ID.

Microsoft.Windows.Update.Orchestrator.RebootFailed

This event sends information about whether an update required a reboot and reasons for failure to help keep Windows up to date.

The following fields are available:

- **updateId** Update ID.
- **batteryLevel** Current battery capacity in mWh or percentage left.
- **RebootResults** Hex code indicating failure reason. Typically, we expect this to be a specific USO generated hex code.
- **installRebootDeferreason** Reason for reboot not occurring.
- **revisionNumber** Update revision number.
- **EventPublishedTime** The time that the reboot failure occurred.
- **deferReason** Reason for install not completing.
- **wuDeviceid** Unique device ID used by Windows Update.
- **flightID** Unique update ID.
- **rebootOutsideOfActiveHours** Indicates the timing that the reboot was to occur to ensure the correct update process and experience is provided to keep Windows up to date.
- **uxRebootstate** Indicates the exact state of the user experience at the time the required reboot was initiated to ensure the correct update process and experience is provided to keep Windows up to date.
- **updateScenarioType** The update session type.

Microsoft.Windows.Update.Orchestrator.RestoreRebootTask

This event sends data indicating that a reboot task is missing unexpectedly on a device and the task is restored because a reboot is still required, to help keep Windows up to date.

The following fields are available:

- **RebootTaskRestoredTime** Time at which this reboot task was restored.
- **wuDeviceid** Device id on which the reboot is restored
- **revisionNumber** Update revision number.
- **updateId** Update ID.

Microsoft.Windows.Update.Orchestrator.SystemNeeded

This event sends data about why a device is unable to reboot, to help keep Windows up to date.

The following fields are available:

- **eventScenario** End to end update session ID.
- **wuDeviceid** Unique device ID used by Windows Update.

- **systemNeededReason** Reason ID
- **updateId** Update ID.
- **revisionNumber** Update revision number.
- **rebootOutsideOfActiveHours** Indicates the timing that the reboot was to occur to ensure the correct update process and experience is provided to keep Windows up to date.
- **uxRebootstate** Indicates the exact state of the user experience at the time the required reboot was initiated to ensure the correct update process and experience is provided to keep Windows up to date.
- **updateScenarioType** The update session type.

Microsoft.Windows.Update.Orchestrator.UpdatePolicyCacheRefresh

This event sends data on whether Update Management Policies were enabled on a device, to help keep Windows up to date.

The following fields are available:

- **wuDeviceid** Unique device ID used by Windows Update.
- **policyCacherefreshtime** Refresh time
- **policiesNamevaluesource** Policy Name
- **updateInstalluxsetting** This shows whether a user has set policies via UX option
- **configuredPoliciescount** Policy Count

Microsoft.Windows.Update.Orchestrator.UpdateRebootRequired

This event sends data about whether an update required a reboot to help keep Windows up to date.

The following fields are available:

- **updateId** Update ID.
- **revisionNumber** Update revision number.
- **wuDeviceid** Unique device ID used by Windows Update.
- **flightID** Unique update ID.
- **interactive** Indicates the reboot initiation stage of the update process was entered as a result of user action or not.
- **uxRebootstate** Indicates the exact state of the user experience at the time the required reboot was initiated to ensure the correct update process and experience is provided to keep Windows up to date.
- **updateScenarioType** The update session type.

Microsoft.Windows.Update.UpdateStackServicing.CheckForUpdates

This event sends data about the UpdateStackServicing check for updates, to help keep Windows up to date.

The following fields are available:

- **EventScenario** The scenario of the event. Example: Started, Failed, or Succeeded
- **StatusCode** The HRESULT code of the operation.
- **CallerApplicationName** The name of the USS scheduled task. Example UssScheduled or UssBoot
- **ClientVersion** The version of the client.
- **EventInstanceID** The USS session ID.
- **WUDeviceID** The Windows Update device ID.
- **ServiceGuid** The GUID of the service.
- **BspVersion** The version of the BSP.
- **OemName** The name of the manufacturer.
- **DeviceName** The name of the device.
- **CommercializationOperator** The name of the operator.

- **DetectionVersion** The string returned from the GetDetectionVersion export of the downloaded detection DLL.

Microsoft.Windows.Update.Ux.MusNotification.RebootNoLongerNeeded

This event is sent when a security update has successfully completed.

The following fields are available:

- **UtcTime** The Coordinated Universal Time that the restart was no longer needed.

Microsoft.Windows.Update.Ux.MusNotification.RebootScheduled

This event sends data about a required reboot that is scheduled with no user interaction, to help keep Windows up to date.

The following fields are available:

- **updateId** Update ID of the update that is getting installed with this reboot.
- **ScheduledRebootTime** Time of the scheduled reboot.
- **wuDeviceid** Unique device ID used by Windows Update.
- **revisionNumber** Revision number of the update that is getting installed with this reboot.
- **forcedreboot** True, if a reboot is forced on the device. False, otherwise.
- **rebootArgument** Argument for the reboot task. It also represents specific reboot related action.
- **rebootScheduledByUser** True, if a reboot is scheduled by user. False, if a reboot is scheduled automatically.
- **activeHoursApplicable** True, If Active Hours applicable on this device. False, otherwise.
- **rebootOutsideOfActiveHours** True, if a reboot is scheduled outside of active hours. False, otherwise.
- **rebootState** The state of the reboot.

Microsoft.Windows.Update.Ux.MusNotification.ToastDisplayedToScheduleReboot

This event is sent when a toast notification is shown to the user about scheduling a device restart.

The following fields are available:

- **UtcTime** The Coordinated Universal Time when the toast notification was shown.

Microsoft.Windows.Update.Ux.MusUpdateSettings.RebootScheduled

This event sends basic information for scheduling a device restart to install security updates. It's used to help keep Windows up-to-date.

The following fields are available:

- **ScheduledRebootTime** The time that the device was restarted.
- **updateId** The Windows Update device GUID.
- **revisionNumber** The revision number of the OS being updated.
- **wuDeviceid** The Windows Update device GUID.
- **forcedreboot** Is the restart that's being scheduled a forced restart?
- **rebootArgument** The arguments that are passed to the OS for the restarted.
- **rebootScheduledByUser** Was the restart scheduled by the user? If the value is false, the restart was scheduled by the device.
- **activeHoursApplicable** Is the restart respecting Active Hours?
- **rebootOutsideOfActiveHours** Was the restart scheduled outside of Active Hours?
- **rebootState** The state of the restart.

Winlogon events

Microsoft.Windows.Security.Winlogon.SetupCompleteLogon

This event signals the completion of the setup process. It happens only once during the first logon.